Anexos

1. Cálculo de motores y caja reductora

   El robot a desplazar, tiene un peso de 40 kg. Si asumimos que en la peor de las condiciones el coeficiente de fricción es 1. Tenemos la siguiente situación:

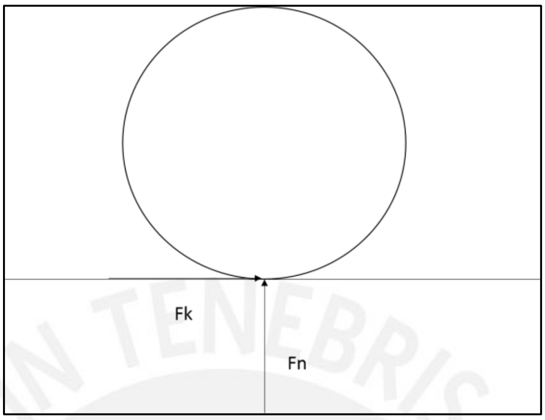


Figura 1. Esquema de fuerzas

El robot es totalmente simétrico, con lo cual se puede considerar que en cada rueda se tiene una distribución del peso equitativa, entonces, el peso por llanta sería de 10 kg, ya que se cuenta con un sistema de 4 llantas. La fuerza de fricción máxima a vencer sería:

$$Fk = Fn * uk$$

$$Fk = 10 * 9.81 * 1$$

$$Fk = 98.1\ N$$

Con una fuerza de 98.1 N el torque máximo a ejercer por el motor sería:

$$Tmax = Fk * R$$

$$Tmax = 98.1 * 55$$

$$Tmax = 5.25\ Nm$$

Entonces el eje del motor debe ser capaz de mover los 5.25 Nm, ya que sería la situación más crítica.

Primero se procederá a calcular la caja reductora, ya que de ella depende cuanto torque se le puede incrementar al motor, y seguidamente se calculará cuánto torque debe proveer el motor para que dicha caja reductora sea eficiente. Con un factor de seguridad de 1.5 el Tmax a ejercer sería de aproximadamente 7.5 Nm.

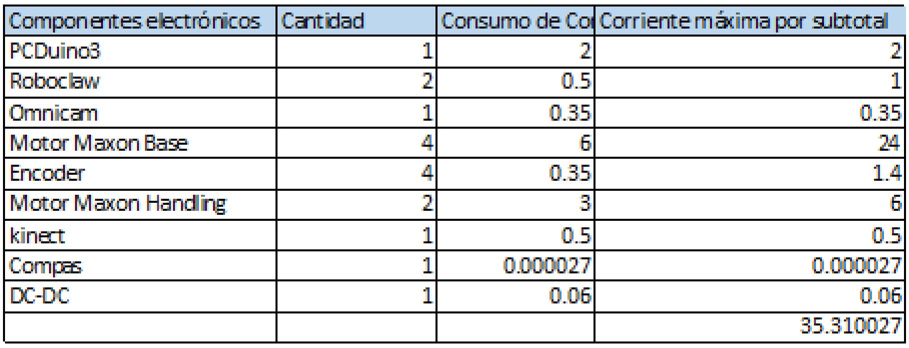
La caja reductora GP42B con una eficiencia de 72% ofrece un torque de 7.5Nm y una reducción de 43:1, con lo cual, se procederá a calcular el torque máximo del motor.

$$TmaxMotor = \frac{7.5}{0.72} * \frac{1}{43}$$

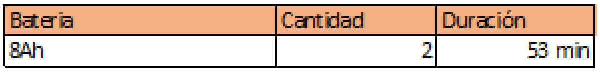
$$TmaxMotor = 169.52\ mNm$$

El motor RE40 ofrece un torque máximo de 170 mNm, con lo cual satisface la relación que se requiere para trabajar en situaciones críticas con un factor de seguridad.

2. Consumo de corriente
   Se tomó la información de los datasheets de las tarjetas y componentes para calcular la autonomía total del robot. Como se puede observar en la tabla adjunta, se tiene una autonomía de 53 min, lo cual da un factor de seguridad de 2, ya que los encuentros constan de 2 tiempos de 15 min cada uno.



Componentes electrónicos	Cantidad	Consumo de Co	Corriente máxima por subtotal
PCDuino3	1	2	2
Roboclaw	2	0.5	1
Omnicam	1	0.35	0.35
Motor Maxon Base	4	6	24
Encoder	4	0.35	1.4
Motor Maxon Handling	2	3	6
kinect	1	0.5	0.5
Compas	1	0.000027	0.000027
DC-DC	1	0.06	0.06
			35.310027



Bateria	Cantidad	Duración
8Ah	2	53 min

3. Potencia
   Para el cálculo de la potencia eléctrica se tuvo en cuenta la cantidad de corriente máxima y el voltaje de entrada del sistema, para el cual se utilizan dos baterías Lipo en seria de 24V. Entonces, la potencia eléctrica sería:

$$Pe = 24\ V * 35.31A$$

$$Pe = 847.5\ W$$

Para el cálculo de la potencia mecánica se tuvo en cuenta la velocidad de los motores, ya que esta se recude de acuerdo a la carga que mueven. (Ver *Figura 2*)

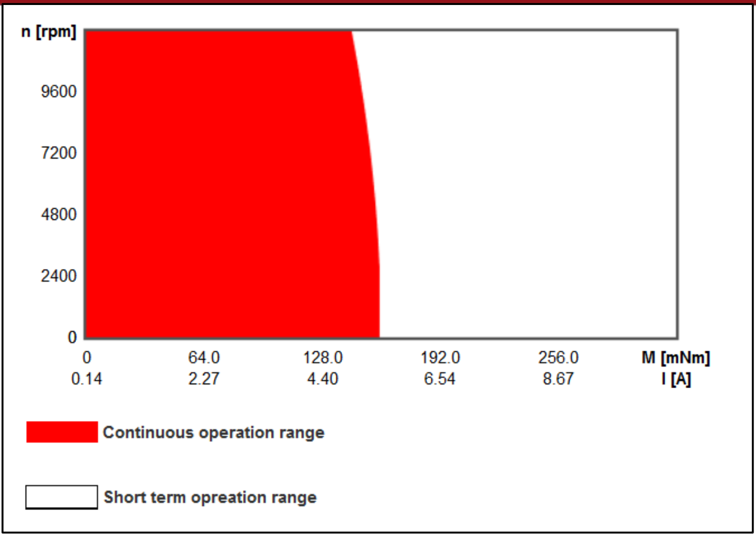


Figura 2. Gráfica del motor

Asumiendo que el robot operará constantemente a 128 mNm, el robot se puede mover a una velocidad de 9600 rpm, ya que se encuentra dentro de su zona de trabajo. En rad/s sería 9600 revoluciones en 60 segundos, por lo tanto, se tiene un valor de 160 rad/s, y el torque del motor sería de 0.128 Nm, el cálculo sería el siguiente:

$$Pmec = 0.128 * 160$$

$$Pmec \sim 21\ W$$

3. Solenoid Design

When designing a solenoid a number of variables should be taken into account. Variables that determine the behaviour of solenoids are among other things inductance, response time, resistance, force, dimensions and core-material. Moreover operating voltage should be kept in mind, because of limited power supply.

To design a good solenoid these parameters should be balanced carefully. In this chapter this parameters will be explained

3.1 *Self-Inductance*

Self-inductance of solenoids is stipulated by the number of windings, coil length and diameter. It can be calculated with Wheeler's formula (3.1) [4]. This formula holds for coils in which core isn't present inside the coil but it is a good indication for the solenoid.

Self-inductance will not remain constant in the case of a solenoid kicker due to the moving core. Self-inductance will increase when the core moves inside.

$$L[mH]=\frac{0.0315\cdot N^2\cdot\left(\frac{R_1+R_2}{2}\right)^2}{6\cdot\frac{R_1+R_2}{2}+9\cdot l_{coil}+10\cdot\left(R_2-R_1\right)} \qquad (3.1)$$

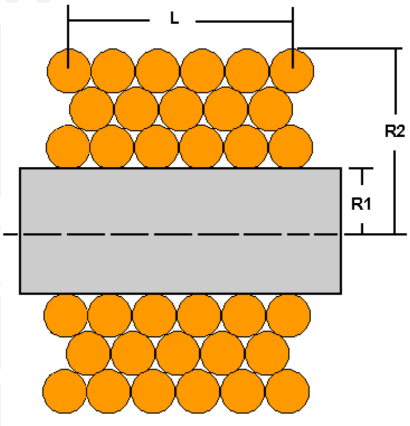


Figure 3.1: Crossection of a Solenoid

With:

N = Number of Turns [-]
R_1 = Inside radius [m]
R_2 = Outside radius [m]
l_{coil} = Length [m]
L = Self-inductance [mH]

Self-inductance is also proportional to magnetic flux, which is proportional to flux intensity.

3.2 *Resistance*

Resistance of solenoids is stipulated by the specific resistance of the wire material, diameter and wire length. Wire length can be calculated with formula (3.2).

$$l=2\pi\cdot\left(R_1+\frac{(R_2-R_1)}{2}\right)\cdot N \qquad (3.2)$$

With:

R_1 = Inside radius [m]
R_2 = Outside radious [m]
N = Number of Turns [-]
l_{wire} = Wire length [m]

Solenoids contain often copper winding wire, because it is a very good conductor, not being magnetised and good availability. Winding wire manufacturers always supply resistance by unit length. In that case resistance can be calculated by formula (3.3).

$$R = \rho \cdot l_{wire} \tag{3.3}$$

With:

R	=	Resistance	[Ω]
ρ	=	Resistance by unit length	$[\Omega\ m^{-1}]$
l_{wire}	=	Wire length	[m]

Resistance is stipulated by wire cross-sectional area. Resistance can be decrease by increasing area. A table with resistance by unit length for several winding wire diameters can be found in attachment 1.

3.3 Force

The acting force at the core can be calculated with Lorentz formula (3.4). In this formula, force is proportional to flux density.

$$F = B \cdot I \cdot l \tag{3.4}$$

With:

B	=	Flux density	[T]
I	=	Current	[A]
l	=	Length	[m]

Flux density B is proportional to current and force also is to both, thus force is proportional to I^2. For high currents saturation appears. This means that almost all small particles inside the core are in an optimum state by which flux density no longer increases. According to (3.4) force is proportional to I for high currents.

Because much energy is needed for one shooting action, the solenoid will be operating in high current domain only. This implies within the saturation area and so proportionality to I.

The Lorentz formula is only applicable for limited conditions but is a very good indication for the generated force. Lorentz holds for solenoids without shielding, because a magnetic shield adds external flux density increasing material. As told in previous chapter no interference with control hardware (like hard disks) is allowed, so magnetic shielding is crucial. In this case the magnetic field co-energy can be used for force calculation. Co-energy is the amount of energy which the magnetic field contains. This can be calculated numerically. Force can be calculated by differentiating co-energy to place. More about this in chapter 5.

Of course it isn't necessary to calculate force directly because the kinetic energy of the plunger is the most important variable. But force-stroke diagrams give a clear view about the boundary effects. A research with the influence of several solenoid parameters to force is done in Chapter 6.

3.4 Time constant

Solenoids have a time constant. This is the value of L/R. A large time constant decreases the reaction time of the solenoid. This time constant causes a delay in applied current in both on and off-switching. In general it takes 5 times the time-constant to build and break down the current. An example is shown in figure 3.2. After 5 units L/R current reaches maximum. Current is switched of and it takes again 5 units L/R to decay.

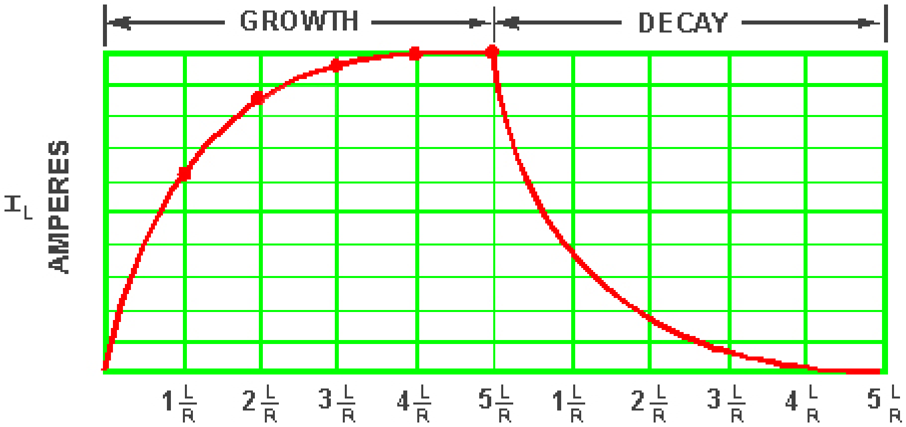


Figure 3.2: Time constant example

This current delay is caused by Faraday's law which creates an e.m.f. as current is turned on. According to Lenz law it's a back e.m.f. and opposes the change of current. This is a very unwanted effect because it takes some time until the solenoid operates at full power.

To lower the time constant a low self-inductance and / or a high resistance can be chosen. A low self-inductance can be created by decreasing the number of turns and increasing current. But increasing current causes a quadratic increase of power losses due to resistance $P = I^2R$. More heat will be dissipated, thus there is a limit.

The other option is to design with highest resistance. A high resistance is unwanted because of the higher power losses due to resistance and also more heat will be dissipated.

Thus a compromise is necessary to achieve the lowest time constant. The solenoid should be designed with the thinnest wire (means high resistance) and highest current applicable (means low self-inductance).

3.5 Dimensions

To create a concentrated magnetic field the solenoid should be designed as compact as possible in de radial direction. This means that the coil is positioned as close to the core as possible with a minimum of supporting material and space.

The length of the coil should be equal to the core length. Now the centre of the coil, which contains concentrated field lines, is completely filled with the core.

3.6 Temperature

The RoboCup solenoid has to deliver lots of energy in a very short time thus a high power is needed. Heat will be generated due to resistance. This will be a limited amount of energy, because one shooting action takes only about 10 [ms]. In that case very high currents can be applied without melting.

The robot will use the shooting mechanism incidentally in a match, so an incidental heating can be assumed. This implies that enough time to cool down between shooting actions is available. Now convection and radiation can be neglected and all energy will heat up the material. To be sure the solenoid works also on very extreme circumstances for RoboCup, the maximum temperature rise is set to 10˚C after one shooting action.

If all energy will heat up the material, following equation can be used.

$$Q = m_c \cdot c \cdot \left(T_2 - T_1\right) \tag{3.5a}$$

For Q the resistance energy loss can be taken. The mass of copper, m_c, is equal to the wire cross-sectional area times length times the mass density of copper. Now formula (3.5b) can be rewritten in formula (3.6).

$$I^2 \cdot R \cdot \Delta t = \frac{\pi}{4} d_{wire} \cdot l_{wire} \cdot \rho_{wire} \cdot c_{copper} \left(T_2 - T_1\right) \tag{3.6}$$

4. Shielding

To improve the solenoid a magnetic shield is applied. Shielding can be done just with adding a shell of steel (a pipe) on the outside and two cylindrical plates with a hole at both solenoid ends. Adding shields decreases the reluctance at outer solenoid positions.

Reluctance is like electric resistance. A high reluctance means that a high amount of magnetic energy is stored inside. To be sure most energy will be available in the solenoid core the reluctance of the air gap inside the solenoid has to be much higher than reluctance at the outer solenoid positions. For good shielding air gap reluctance has to be at least ten times larger then at outer positions

The reluctance is defined with a standard formula.

$$\Re = \frac{\text{length}}{\text{cross-section}} \cdot \frac{1}{\mu} \qquad (4.1)$$

The airgap is a cylinder with diameter $2r_{gap}$ and length l_{gap}. The relative permeability of air is equal to 1, so the absolute permeability is equal to μ_0. Applying this formula in formula (4.1) gives:

$$\Re_{gap} = \frac{l_{gap}}{\pi r_{gap}^{\ 2}} \frac{1}{\mu_0} \qquad (4.2)$$

For the cylindrical shell with inside radius r_{shell}, wall-thickness t_{shell} and length l_{shell} formula (4.1) becomes:

$$\Re_{shell} = \frac{l_{shell}}{\pi\left(r_{shell} + t_{shell}\right)^2 - \pi r_{shell}^{\ 2}} \cdot \frac{1}{\mu} \qquad (4.3)$$

For the cylindrical plates with hole radius r_{hole}, outside radious r_{plate} and thickness t_{plate} formula (4.1) has to be adapted. The cross-sectional area changes with r and thus the reluctance has to be calculated with infinitesimal steps which are add up together. This is equal to an integral which gives for reluctance:

$$\Re_{plate} = \int_{r_{hole}}^{r_{plate}} \frac{1}{2\pi r \cdot t_{plate} \cdot \mu} dr = \frac{1}{2\pi \cdot t_{plate} \cdot \mu}\left(\ln\left(\frac{r_{plate}}{r_{hole}}\right)\right) \qquad (4.4)$$

For a specified solenoid design are all radii known thus the only unknown parameters for the shield are the thicknesses of the components. To create a good shield the reluctance of both shell and washer have to be much smaller then the reluctance of the airgap. In practice a factor 10 is used. Now also maximum reluctance is known thicknesses can be calculated.

5. Model Shaping

5.1 FEMM

To analyse solenoids a finite element method is used. This method is integrated in a program called FEMM. FEMM stands for Finite Element Method Magnetics. This program is developed by Dr. David Meeker of the University of Virginia.
FEMM is developed to solve 2-dimensional time-independent magnetic problems. The Maxwell equations are used to solve these problems. Axi-symmetric problems can also be analysed, which is very handy for cylindrical solenoids.
FEMM is Freeware and can be downloaded at http://femm.foster-miller.net/cgi-bin/efileman/efileman.cgi

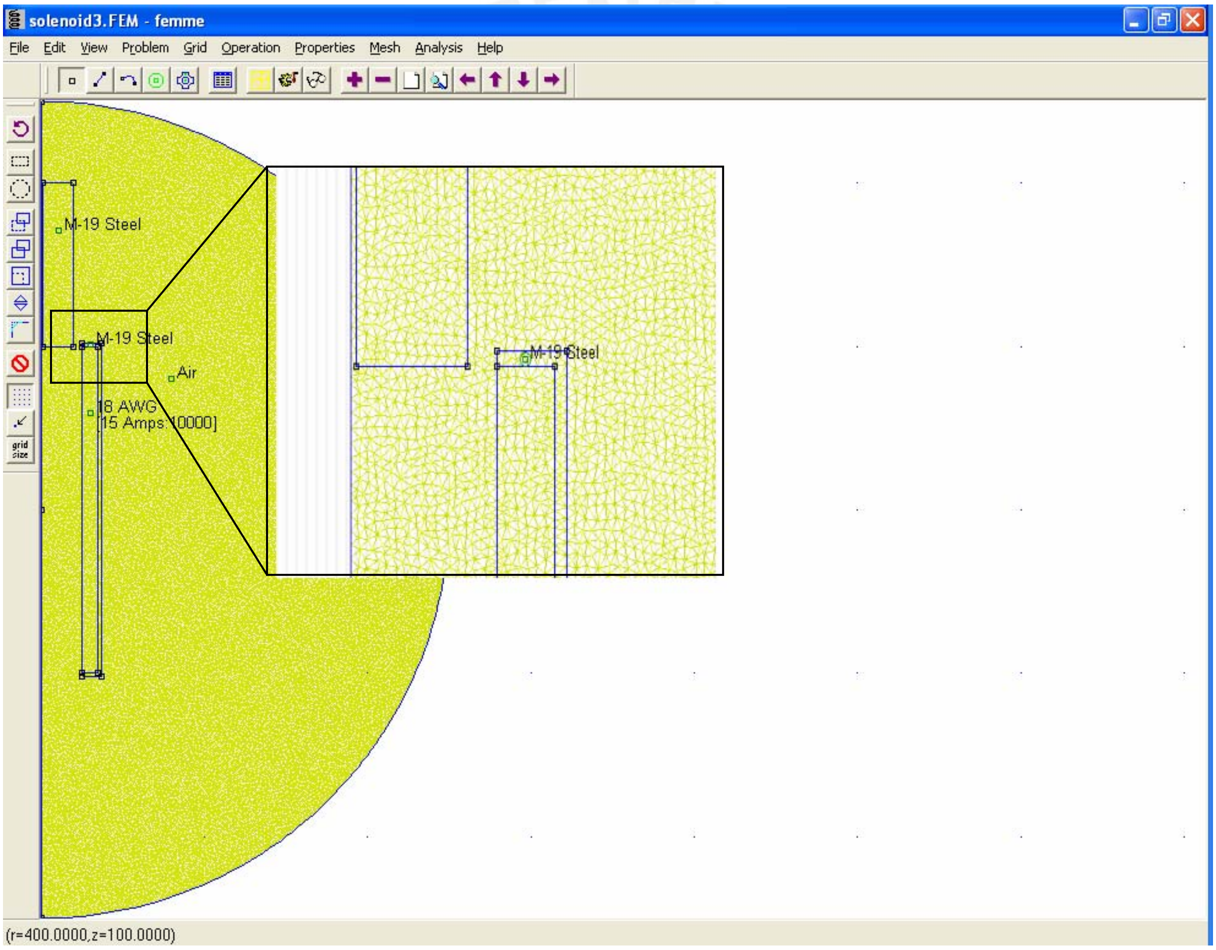
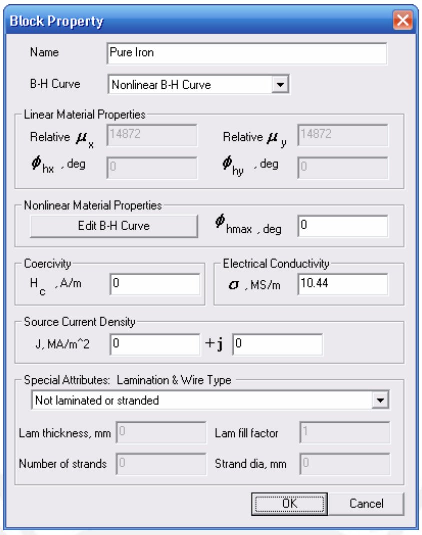


Figure 5.1: Screenshot of FEMM with a solenoid design

Figure 4.1 contains a screenshot of the interface of FEMM. The placed in square is a magnification of the area around the core to show the mesh. By placing nodes en connecting them together lines can be drawn. Almost every form can be created. These forms have to contain closed surfaces. Different properties can be attached to surfaces, like material properties, circuit properties, geometric properties, boundary conditions and mesh conditions.

The program also contains a library with common used materials. This library contains winding-wire, different core materials and air. All relevant material parameters are included, both linear and non-linear.



Block Property

Name: Pure Iron

B-H Curve: Nonlinear B-H Curve

Linear Material Properties

Relative μ_x: 14872

Relative μ_y: 14872

ϕ_{hx}, deg: 0

ϕ_{hy}, deg: 0

Nonlinear Material Properties

Edit B-H Curve

ϕ_{hmax}, deg: 0

Coercivity

H_c, A/m: 0

Electrical Conductivity

σ, MS/m: 10.44

Source Current Density

J, MA/m^2: 0 +j 0

Special Attributes: Lamination & Wire Type

Not laminated or stranded

Lam thickness, mm: 0

Lam fill factor: 1

Number of strands: 0

Strand dia, mm: 0

Figure 5.2: Screenshot of material properties

The linear properties contain permeability. The non-linear section contains a table with saturation data, also known as the B-H curve. In figure 4.3 displays the B-H curve of iron. The effects of saturation are clearly visible. Iron begins to saturate at 1.5 Tesla. The user can decide either to solve linear or to solve non-linear in the materials property menu.

In the boundary conditions menu six different boundary conditions can be chosen. Only a mixed boundary condition is used in the solenoid case to approximate an unbounded open space. This is a approximation to simulate the solenoid in free air circumstances in where it looks that there is no boundary. For more information about the other boundary conditions I refer to the FEMM manual [9].

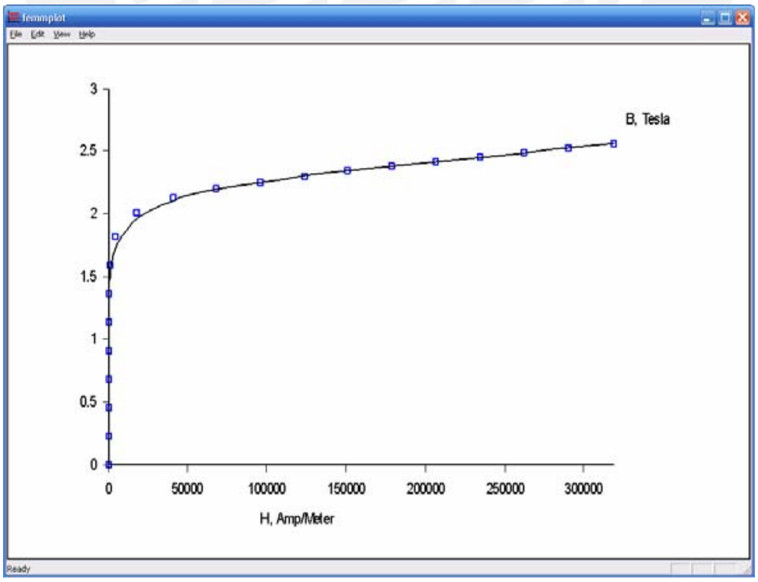


Figure 5.3: B-H curve of iron

When all properties are applied a mesh can be created and the problem can be solved. When the solver is ready, results can be viewed in a new window. Several parameters can be calculated. In the solenoid case only the Lorentz force and magnetic field co-energy are used.

FEMM also contains a post- and pre-processor. It is possible to create scriptfiles and change your model with the pre-processor and to calculate automatically with the postprocessor. With these scriptfiles a loop of calculations can be made with changed conditions. With a pre-processor scriptfile the plunger can be moved automatically with small steps to calculate energy-stroke data for example. The programming language which is used in the scriptfiles is lua. More information and a detailed documentation about this language can be found at www.lua.org.

5.2 Solving method

A modelled RoboCup solenoid is a time independent magnetostatic problem. In This situation following equation holds for flux density B.

$$\nabla \bullet B = 0 \tag{5.1}$$

And for field intensity (H):

$$\nabla \times H = J \tag{5.2}$$

When using lineair media, flux density B must satisfy:

$$B = \mu H \tag{5.3}$$

And in non-linear (saturated) media permeability satisfies:

$$\mu = \frac{B}{H(B)} \tag{5.4}$$

FEMM goes about finding a field that satisfies equation (5.1) until (5.3) via a magnetic vector potential (A) approach.

Flux density is related to this vector with:

$$B = \nabla \times A \tag{5.5}$$

Equation (5.5) always satisfies equation (5.1) and now equation (5.2) can be rewritten.

$$\nabla \times \left(\frac{1}{\mu(B)} \nabla \times A \right) = J \tag{5.6}$$

With this equation the vector a can be calculated. Vector A contains 3 values in a 3D problem, but in a 2D axi-symmetric problem 2 out of 3 are equal to zero. After vector A is known flux density and field intensity can be calculated by differentiating and substituting in the upper equations.

5.3 Magnetic Field Energy Calculations:

When the values for B and H are know many properties of the solenoid can be calculated. In our case only the energy is relevant. For energy is known that:

$$W_c = \int \left(\int_0^H B(H')dH' \right) dV \tag{5.7}$$

After calculating the co-energy for a number of positions within the solenoid force estimation can be done with formula (5.8).

$$F = \frac{W_c(x+\delta) - W_c(x)}{\delta} \tag{5.8}$$

When δ -> 0 this is mathematical equal to differentiating F with respect to place

6. Solenoid Tests

Because the magnetic field force determines the amount of kinetic energy, it is important to know how force changes by changing several parameters. There are two parameters which can be adapted easily, current, amount of core material and material used. This should be proportional for a infinitely long solenoid according to chapter 3.

6.1 Current

At first a free chosen solenoid is modelled in FEMM. This solenoid has 10000 turns of 18 AWG wire, see also appendix 1. The core contains an bar of M19 steel with radius 20 [mm]. Between core and coil is a gap of 1[mm] and they both have a length of respectively 100[mm] and 200[mm]. The solenoid is shielded with a shell of 2 mm thickness.

For the boundary an asymptotic boundary conditions is chosen to simulate an unbounded open space to model the solenoid in a limited area with open space behaviour.

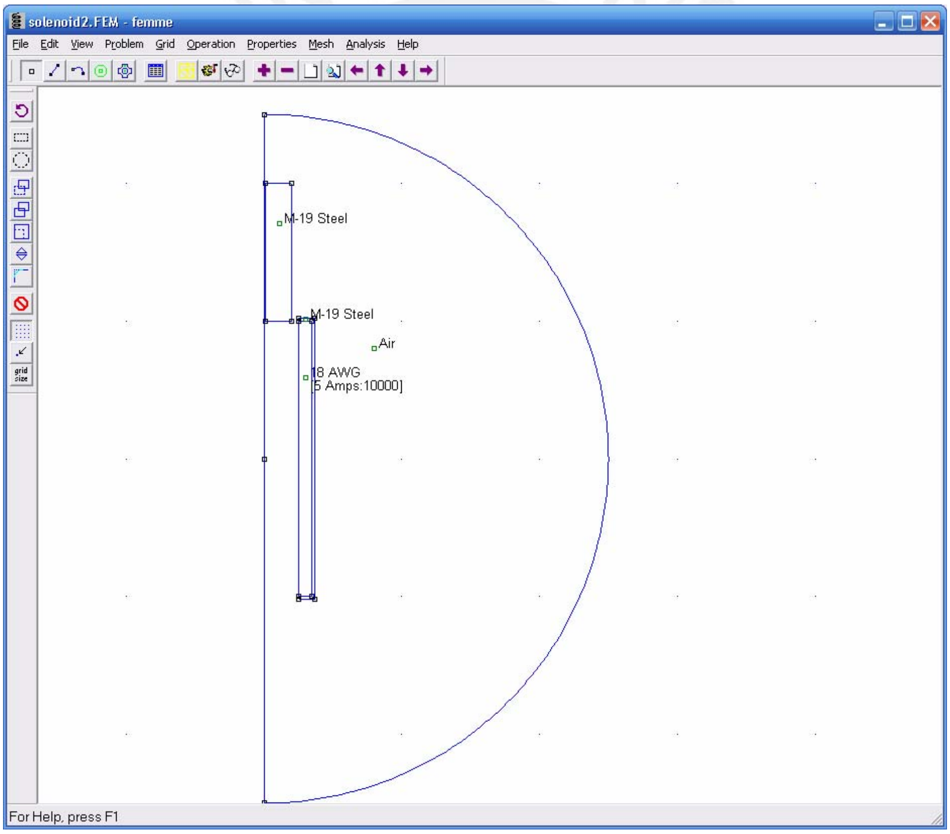


Figure 6.1: The modelled solenoid

The model is simulated with a current of 5, 10, 15 [A]. The results are plot in figure 6.2. The dotted pink line represents the 10 [A] current force/stroke line divided by 2, which should be equal to the 5 [A] current when force is proportional to current. This holds when the largest part of the core is inside the coil ($-100 \le stroke \le 100$), but not for larger strokes due to side effects.

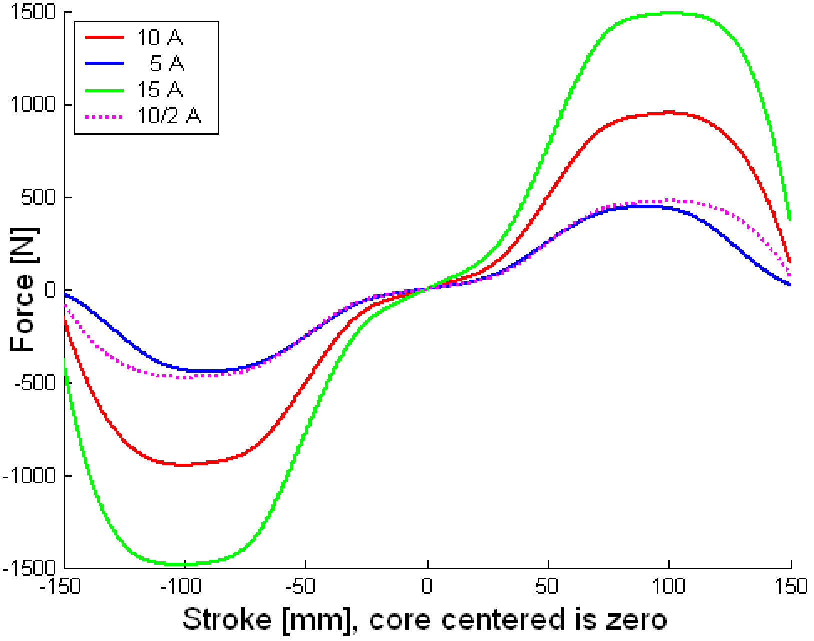


Figure 6.2: Force, stroke for different currents

6.2 Amount of Material

A similar test is done for the amount of material. It is the same solenoid as in the previous test, but now with a fixed current and with variable core dimensions. Current is set to 10 [A] and results are calculated for a full and a half size core. The results are plot in figure 6.3 with the results of the previous test. The outcome is clear, the force in the inside region ($-100 \leq stroke \leq 100$) with a half core is the same as the force with a full core but half current thus force is also proportional to amount of material.

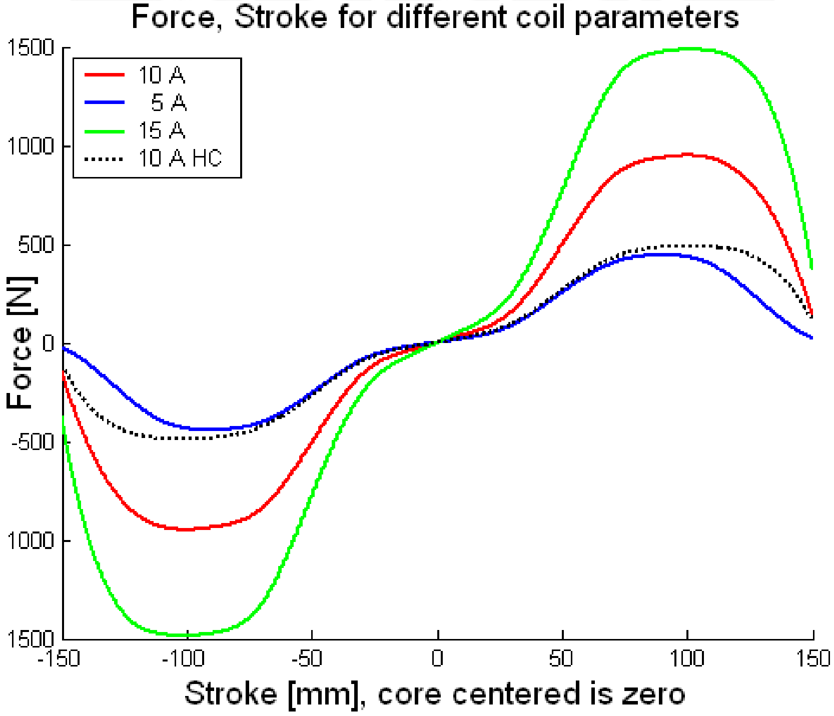


Figure 6.3: Force, stroke for different cores

7. Final design

As seen in previous chapters there are some parameters that should be taken into account. The most important parameters are temperature, the time constant and availability of components for the solenoid circuit board. These are explained In chapter 3.

The time constant is the major parameter for the RoboCup solenoid. To keep this constant low, inductance must be kept low and a minimal wire thickness should be used without melting. To create enough force high current is needed through these thin wires. This results in a very high voltage because of Ohm's law.

A large capacitor is needed as an energy buffer. These capacitors become more expensive as rated voltage increases. Above 450V are suitable capacitors not anymore available. So the maximum voltage is set to 450V for our solenoid.

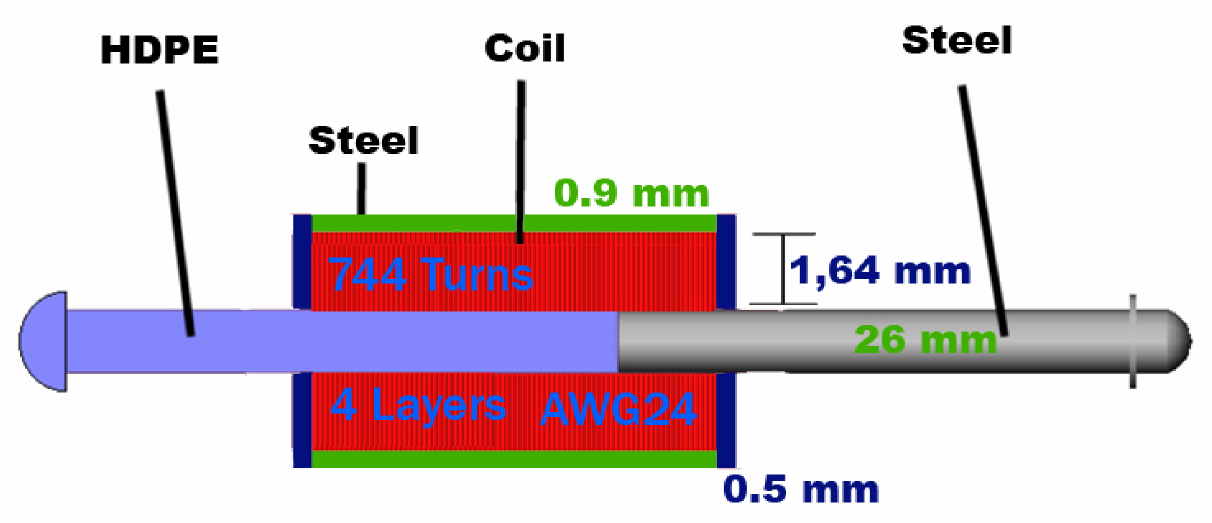


Figure 7.1 Solenoid

To create at least 42.5 [J] at full power the number of turns is set to 744 in 4 layers with current of 68 [A]. This is the result of optimisation in Femm. The maximum voltage may not raise the 450 [V] barrier, because there are no electric components available above that voltage. Resistance is kept high, by using less windings with a high current, to create a low time constant. But heat generation is also taken in account.

Each layer contains 186 turns with a winding factor of 0.95. AWG 24 wire is used thus the coil-resistance becomes 5.56 [Ohm] (See appendix [1]). Self-Inductance can be calculated with formula (3.1). This gives 3790 [uH]. The time constant L/R is thus 0.68 [ms]. The solenoid length is 10 [cm], due to the maximum stroke set by the Robocup Rules. All other dimensions are shown in figure 7.1.

A bar of 1018 steel with a HDPE bar attached is used with a diameter of 26 [mm] as a core. 1018 steel is chosen because it has the highest magnetic performance in FEMM. The B-H curve can be found in figure 7.3. HDPE is chosen because it has the right density to give the complete bar an equal mass with respect to the ball. Between the coil and core is an little gap of 1mm to reserve space for a pipe or something similar to wind the coil on to.

The shield thickness is calculated with the reluctance formula's in chapter 4. A higher thickness then needed is used for the (blue) endplates, because otherwise they were very thin and fragile. The (green) outer shell reluctance is equivalent to the end plates when a thickness of 0.9 [mm] is used.

A FEMM model is made with this design. The final solenoid graph is shown in figure 7.1. The average plunger velocity is 9.09 [m/s] thus the core travels from end to centre in 11 milliseconds with constant current thus one shooting action uses 336 [J]. The maximum plunger speed becomes 14.12 [m/s] and the ball reaches 10.93 [m/s] according to the energy calculation in chapter 2.1

A remark has to be made, because all simulations are done with constant current. It is the average current supplied by the circuit. But In practice current varies in time because of the time constant so the final plunger speed is in practice a little bit lower.

This solenoid is not able to reverse it's movement, because the plunger is in a dead point after a shot. Thus another measure is required to bring the solenoid after actuation back. The easiest and cheapest way is to connect a little spring at the steel bar's end. The spring should have a spring constant which is just high enough to overcome friction.

An overview with all solenoid properties can be found in table 7.1.

Solenoid Properties	
Energy consumption per shot [J]	336
Inductance [mH]	37.9
Capacitor [mF]	4.7
Transistor Switch	International Rectifier IRG4PC50FD
Windings per layer [-]	186
Layers [-]	4
Winding factor [-]	0.95
Wire Size [AWG]	24
Solenoid Length [mm]	100
Coil Resistance [Ohm]	5.56
Core Diameter [mm]	26
Core Material [-]	1018 Steel and HDPE
Shielding Shell Thickness [mm]	0.9
Shielding End Plates Thickness [mm]	0.5
Theoretical Ball Speed [m/s]	10.9
Voltage [V]	450

Table 7.: Solenoid Properties

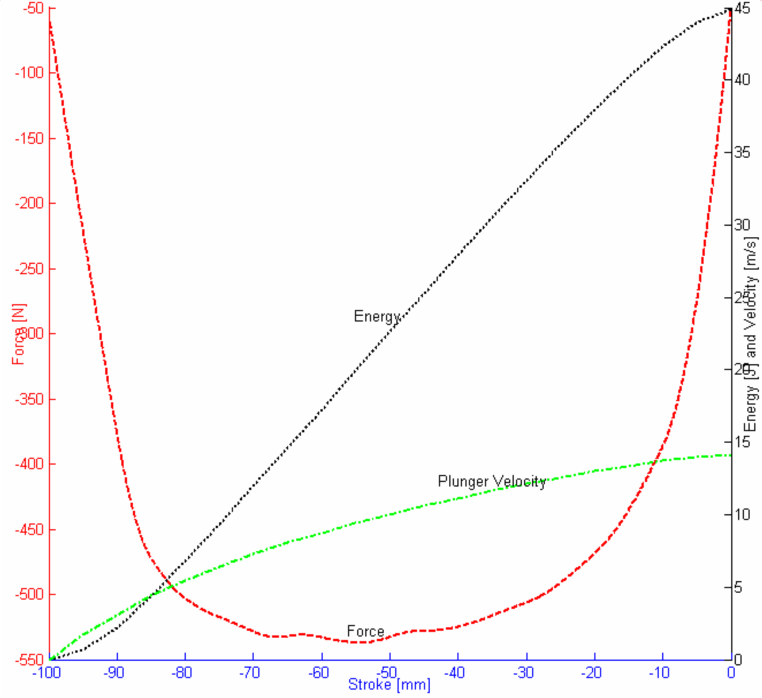


Figure 7.2: Final Solenoid diagram

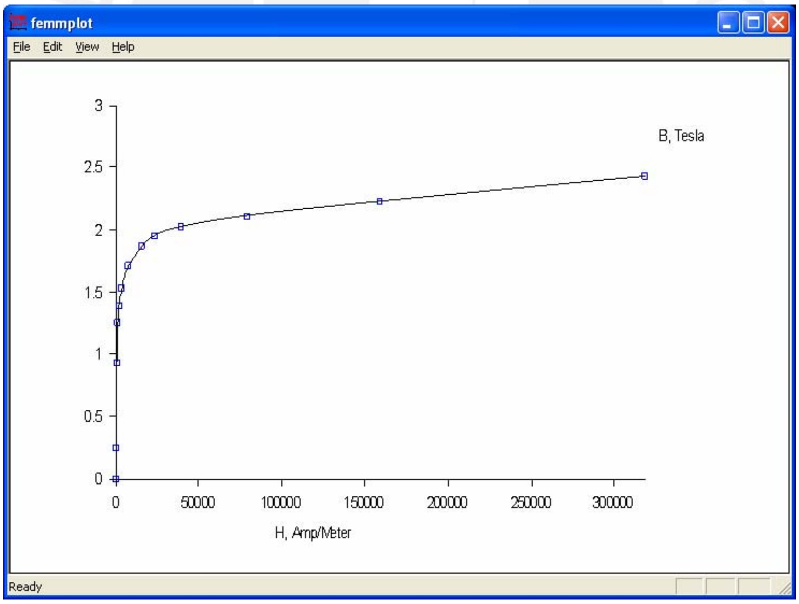


Figure 0.3: B-H Curve of 1018 steel

8. Circuit Design

The in the previous chapter designed solenoid needs a very fast switching circuit otherwise the solenoid will melt due to generated heat. Active cooling is not an option because the solenoid has a Power of 30 kW. The circuit must also contain a capacitor to slowly store energy in idle state and release it fast over the solenoid. An adapted RLC circuit (Figure 8.1) is used and analyzed in Pspice.

The first component is a dc/dc converter to convert the available battery power to 450 volt.

The Second component is a resistance which determines the load speed of the capacitor. The capacitor has a capacity of 4.7 [mF] and is rated for 450V. This is enough to shoot once at full power and once at half power without reloading. It takes 10 seconds to reload after one full power shot.

Parallel to the capacitor is the solenoid, modeled with a coil and a resistance with the values of the RoboCup solenoid.

The solenoid is connected to a transistor which is controlled by a pulse source. The transistor can handle high currents and is manufactured by the International Rectifier Group, item nr IRG4PC50FD.When the source-signal is high the transistor is closed (solenoid is activated), when low it is open (solenoid is idle). The transistor opens in 380 [ns] and closes in 70 [ns]. Specification sheet is available in attachment 2.

Parallel to the solenoid is a diode with a resistance to “catch” the back-current generated when turning of the solenoid (see also paragraph 3.5).

Analysis is done in PSpice. At t=0 is the capacitor empty and is switch open. At t=14.1 is the capacitor full and the switch closes. In figure 8.2 is the current applied on the solenoid shown. Figure 8.3 contains the load diagram of the capacitor.

In figure 8.3 can be seen that the capacitor is filled in 14 seconds, because the current over the capacitor is almost zero. Then a shot is simulated. The applied current on the solenoid is shown in figure 8.2. The start- and endeffects of the time constant are present after applying current and after removing the current. When the shot starts, the capacitor starts immediately to reload after 7 seconds is it filled again. But when there is not enough time to reload there is enough energy available to shoot one more time at half power.

The solenoid shooting force can also be modulated with this circuit when Pulse Width Modulation is applied to the pulse source. With PWM current can be turned off at every moment and the solenoid stops with generating energy. So the plunger will not accelerate anymore.

The circuit with capacitor should be placed in a “black box” to guarantee safety for all. It is very dangerous because of high voltage and high current. The capacitor contains also a huge amount of electric energy. People can get injured or be killed when touching the wrong parts, so this safety measure is required.

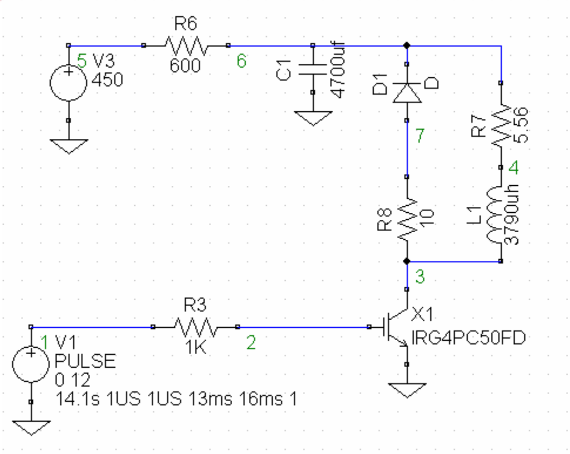


Figure 8.1: Solenoid circuit design

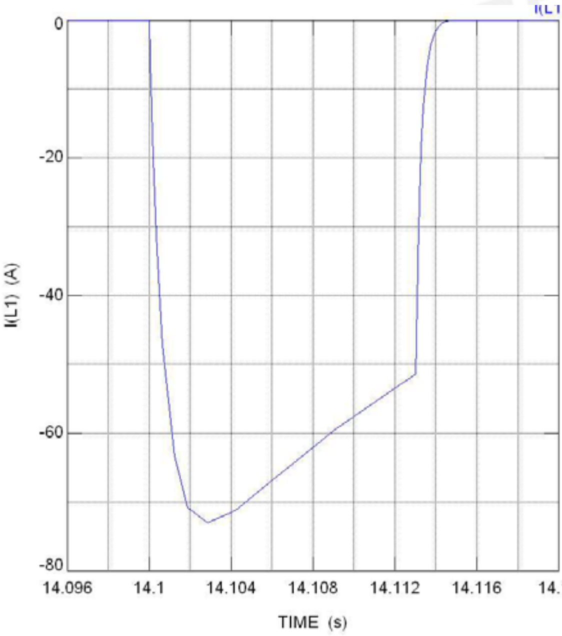


Figure 8.2: Current Applied on Solenoid

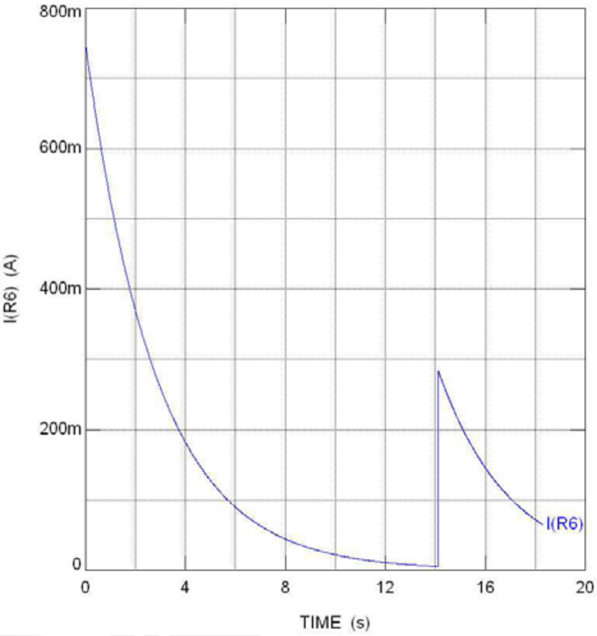


Figure 8.3: Capacitor Load

Conclusion

The solenoid is the best option for a RoboCup shooting device. It is powerful, not very expensive, robust, lightweight and small. It is also able to modulate shooting power by applying pulse width modulation on the pulse source in the control circuit.

It is the best option but also the most dangerous one because of the use of high power electric components. This should also be taken into account, because people can get seriously injured. But when good safety measures are applied, by placing the dangerous components in a shielded box, nobody would get harmed.

The shooting device is also ready for the near future in the RoboCup competition. It's expected (and one of our goals) that in the near future robots can pass to other robots. Power modulation and thus different ball speeds are required for good passing. Power modulation is possible with the solenoid, so this should not be a problem.

The design made in chapter 7 is able to meets also Tech United's requirement to shoot with 10 m//s and is, together with the electric circuit, ready to be build.

Further research recommendations:

- To build this prototype
- Research controllability

Bibliography

1. The solenoid, a module on electricity and magnetism, Carl R. Stannard et al.

2. Segway CMBalance Robot Soccer Player, Jeremy Searock et al., May 2004

3. RoboCup website
http://www.robocup.org

4. The Magnetic Gun Club Website:
http://mgc314.home.comcast.net

5. Coilgun Systems Website:
http://www.coilgun.eclipse.co.uk

6. Weico Wire Winding Data
http://www.weicowire.com/windingdata.htm

7. Philips CFT Team Website
http://www.apptech.philips.com/robocup/

8. Team Minho Website
http://www.robotica.dei.uminho.pt/robocup/

9. FEMM manual v3.4, Dr. David Meeker, December 2004

10. FEMM Tutorial, Dr. David Meeker, December 2004

List of Symbols

$\ddot{x}$	Acceleration [m/s^2]
x	Distance [m]
K	Spring Constant [N/m]
m	Mass [Kg]
t	Thickness [m]
$m_{plunger}$	Plunger Mass [Kg]
$v_{plunger}$	Plunger Velocity [m/s]
m_{ball}	Ball Mass [Kg]
v_{ball}	Ball Velocity [m/s]
J_{ball}	Moment of inertia of Ball [Kg m^2]
ω	Rotational Speed [rad/s]
r_{ball}	Ball radius [m]
$E_{solenoid}$	Solenoid Energy [J]
L	Self-Inductance [mH]
N	Number of Turns [-]
R_1	Inner Coil radius [m]
R_2	Outer Coil radius [m]
l_{coil}	Coil length [l]
l_{wire}	Wire length [l]
R	Resistance [Ohm]
ρ	Specific Resistance [Ohm/m]
F	Force [N]
B	Flux Density [T]
I	Current [A]
l	Length [l]
P	Power [W]
Q	Generated Heat [J]
m_c	Copper Mass [Kg]
c	Specified heat coefficient [J/Kg/K]
T_2	Begin Temperature [K]
T_1	End Temperature [K]
d_{wire}	Wire Diameter [m]
ρ_{wire}	Density Wire [Kg/m^3]
c_{copper}	Specified heat coefficient copper [J/Kg/K]
$\Re$	Reluctance [AN]
$\Re_{gap}$	Airgap Reluctance [AN[
$\Re_{shell}$	Shell Reluctance [AN]
$\Re_{plate}$	Plate Reluctance [AN]
μ	Permeability [Hm^{-1}]
μ_0	Permeability of vacuum [Hm^{-1}]
μ_r	Relative permeability [Hm^{-1}]
l_{gap}	Airgap length [m]
r_{gap}	Airgap radius [m]
l_{shell}	Shell length [m]
r_{shell}	Shell radius [m]

t_{shell}	Shell Thickness [m]
t_{plate}	Endplate thickness [m]
r_{hole}	Hole radius [m]
r_{plate}	Endplate radius [m]
H	Field Strength [A/m]
J	Current Density [A/m^2]
A	Area [m^2]
W_c	Magnetic Field Co-Energy

Middle Size Robot League Rules and Regulations for 2014

Version - 17.1 20140123

MSL Technical Committee 1997–2014

Minoru Asada
Tucker Balch
Andrea Bonarini
Ansgar Bredenfeld
Steffen Gutmann
Gerhard Kraetzschmar
Pedro Lima
Emanuele Menegatti
Pieter Jonker
Alireza Fadaei Tehrani
Takayuki Nakamura
Gerald Steinbauer
Martin Lauer
Yasunori Takemura
Huimin Lu
Enrico Pagello
Fernando Ribeiro
Thorsten Schmitt
Wei-Min Shen
Hans Sprong
Shoji Suzuki
Yasutake Takahashi
Paul G. Ploeger
Frank Schreiber
Jürge van Eijck
Akihiro Matsumoto
Saeed Shiry Ghidary
Roel Merry
Bernardo Cunha
Darwin Lau
Saeed Ebrahimijam
Oliver Zweigle
António J. R. Neves
José Miguel Almeida
Hamed Rasam Farad
Robin Soetens
Zhao Yong
Shota Chikushi
Wu Jia Hao

January 23, 2014

Disclaimer, Acknowledgements, and Copyright

This documents reproduces and makes reference to the official Laws of the Game as set forth by FIFA and the International Football Association Board. For clauses reproduced from the official Laws of the Game, the copyright and all rights are reserved by FIFA and the International Football Association Board. For all other parts, in particular the RoboCup Changes and Comments to FIFA Laws, the copyright and all rights are reserved by the RoboCup Federation.

Document Status

This is the draft version (ver. 16.1) of the rules that will be used for the 18th RoboCup Soccer World Championships, to be held in Joao Pessoa, Brazil and for MSL competitions in 2014.

Middle Size Robot League – Rules and Regulations –

Preamble

Rules Philosophy:

1. RoboCup rules should not in any way describe the behaviour of how the game is played. Rules should only ensure that a fair competition takes place, and encourage both technical and creative development.
2. RoboCup rules should avoid to constrain the design of robots, including their mechanical construction, their use of sensory systems, communication equipment, etc., **unless** the constraints seem necessary to foster scientific progress or to ensure a fair competition.

   Example constraint: Global vision systems are not permitted in the Middle Size League.
3. Teams should avoid to search for gaps or inconsistencies in the rules to achieve advantages in specific game situations. If a team finds such gaps or inconsistencies, they are explicitly requested to report those to the technical committee.

Design Philosophy:

1. Each team should design their robots without making interpretations or placing expectations on how the environment around the field will look like, about spectators, what other teams will do, what robots should look like, or how they will behave.
2. Each team is under no obligation to accommodate modifications to their own robots to suit other teams. Any such modification is by mutual consent only.

Organization of Rules:

Rules and regulations for the RoboCup Middle Size robot League are given in two major sections:

1. Official FIFA Laws.
   They are reproduced in this document. FIFA Laws are annotated with RoboCup Changes and Comments as appropriate.
2. Competition Rules, which define issues like team qualification, etc., for a specific tournament like the annual RoboCup Robot Soccer World Championships.

Conflict Resolution Rules:

In case of any conflict between Laws and Rules:

1. Regulations specified as RoboCup Changes and Comments in the FIFA Laws section override FIFA Laws.
2. Regulations specified in Competition Laws override any FIFA Laws, including RoboCup Changes and Comments.

Download Sites

The current version of the rules can be downloaded as PDF document at

`http://wiki.robocup.org/wiki/Middle_Size_League/`.

The original FIFA Laws can be found at

`http://wiki.msl.robocup-federation.org/`.

Rule Change Proposals and Corrections

If you have found any contradictions or inconsistencies please contact the RoboCup Middle Size League mailing list

`robocup-mid@cc.gatech.edu`,

or the MSL Technical Committee

`rc-msl-tc@lists.robocup.org`.

For subscription to the RoboCup Middle Size League mailing list please send an email to

`https://lists.cc.gatech.edu/mailman/listinfo/robocup-mid`.

For unsubscription to the RoboCup Middle Size League mailing list please send an email to

`https://lists.cc.gatech.edu/mailman/listinfo/robocup-mid`.

Contents

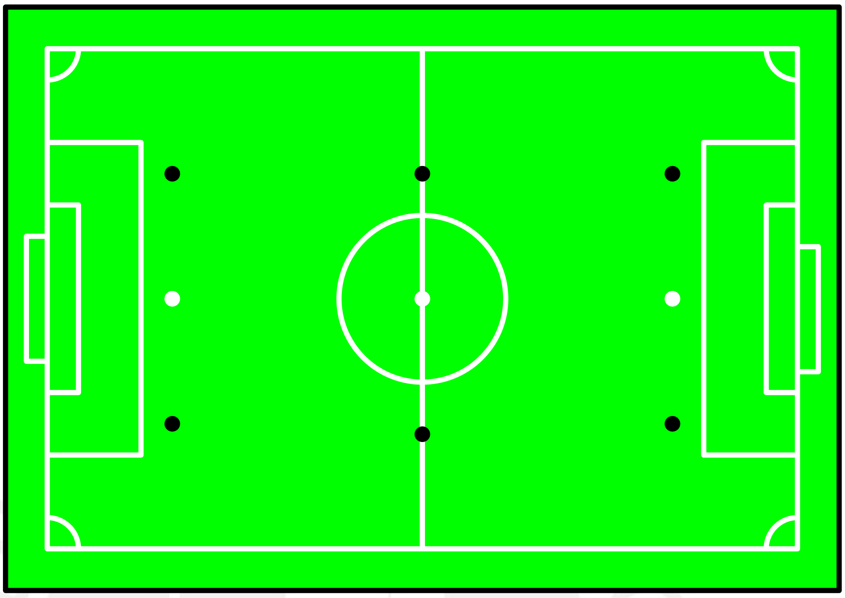
Chapter

FIFA Laws and RoboCup Modifications

Official FIFA Laws include:

- FIFA Law 1 - The Field of Play
- FIFA Law 2 - The Ball
- FIFA Law 3 - The Number of Players
- FIFA Law 4 - The Players' Equipment
- FIFA Law 5 - The Referee
- FIFA Law 6 - The Assistant Referees
- FIFA Law 7 - The Duration of the Match
- FIFA Law 8 - The Start and Restart of Play
- FIFA Law 9 - The Ball In and Out of Play
- FIFA Law 10 - The Method of Scoring
- FIFA Law 11 - Offside
- FIFA Law 12 - Fouls and Misconduct
- FIFA Law 13 - Free Kicks
- FIFA Law 14 - The Penalty Kick
- FIFA Law 15 - The Throw-In
- FIFA Law 16 - The Goal Kick
- FIFA Law 17 - The Corner Kick

FIFA LAW 1 – The Field of Play

	RoboCup Changes and Comments
RC-1.0: **The Field**	 Attention: The above image doesn't show correct proportions!
RC-1.0.1: **Surface**	Choosing a material for the floor is the responsibility of the local organizers of an event. The color of the field surface must be green.
RC-1.0.2: **Lighting**	Artificial light sources may be hung over the field.

FL 1.1 Dimensions

The field of play must be rectangular. The length of the touch line must be greater than the length of the goal line.

National Matches

Length: minimum 90m (100yds), maximum 120m (130yds)
Width: minimum 45m (50yds), maximum 90m (100yds)

International Matches

Length: minimum 100m (110yds), maximum 110m (120yds)
Width: minimum 64m (70yds), maximum 75m (80yds)

	RoboCup Changes and Comments
RC-1.1: **Dimensions**	**RoboCup Matches** Length: minimum 8m, maximum 18m. Width: minimum 6m, maximum 12m. The official field size for this year is 18m × 12m.

FL 1.2 Field Markings

The field of play is marked with lines. These lines belong to the areas of which they are boundaries. The two longer boundary lines are called touch lines. The two shorter lines are called goal lines. All lines are not more than 12.5cm (5ins) wide. The field of play is divided into two halves by a halfway line. The center mark is indicated at the midpoint of the halfway line. A circle with a radius of 9.15m (10yds) is marked around it.

RoboCup	**RoboCup Changes and Comments**
RC-1.2: **Field Markings**	The width of touch and goal lines is 12.5cm. The radius of the center circle is 2m. For RoboCup, the width of all internal lines, like center circle, goal area, and penalty area, is also 12.5cm. Lines are part of the areas they delimit. Therefore measurements are to be taken from the outside of the lines regarding each of these areas.
RC-1.2.1: **Safety Boundary**	The field is surrounded by a black safety boundary, the height of which is between 8cm and 15cm above he field. It is placed 1.5m outside of each field border. The purpose for this boundary is to ensure safety and preventing robots from running into the audience. Further implementation details are defined in the MSL competition construction book. All teams are expected to make no assumptions about this boundary, namely about its mechanical resistance. Thus the above height constraints might be changed by the organizer within their limits without prior notice.
RC-1.2.2: **Ad Panels**	The organizers may place ad panels. The length of a panel may not exceed 150 cm and the height may not exceed 50 cm. Panels must be placed outside of the safety boundary. The ad panels are not intended for localization.
RC-1.2.3: **Restart Spots**	The RoboCup field defines 9 places used for game restart. In addition to the white center marking and the white penalty marks, there are 6 extra virtual spots on the field which will be used for this purpose. The positions of these extra virtual spots (see black dots in the field drawing above) are specified as follows: • 2 spots on the center line, one each halfway between touch line and center mark. • 2 spots each on a line parallel to the goal line through each penalty mark, one each halfway between touch line and the penalty mark. The penalty marks are not used for game restarts, unless a penalty shoot out is called by the referee. The diameter of the center mark is 15cm, while the diameter of the white penalty marks is 10cm.

FL 1.3 The Goal Area

A goal area is defined at each end of the field as follows:

Two lines are drawn at right angles to the goal line, 5.5m (6yds) from the inside of each goalpost. These lines extend into the field of play for a distance of 5.5m (6yds) and are joined by a line drawn parallel with the goal line. The area bounded by these lines and the goal line is the goal area.

RoboCup	**RoboCup Changes and Comments**
RC-1.3: **Goal Area**	For RoboCup, replace "*5.5m*" by "**0.75m**".

FL 1.4 The Penalty Area

A penalty area is defined at each end of the field as follows:

1. Two lines are drawn at right angles to the goal line, 16.5m (18yds) from the inside of each goalpost. These lines extend into the field of play for a distance of 16.5m (18yds) and are joined by a line drawn parallel with the goal line. The area bounded by these lines and the goal line is the penalty area.
2. Within each penalty area a penalty mark is made 11m (12yds) from the midpoint between the goalposts and equidistant to them. An arc of a circle with a radius of 9.15m (10yds) from each penalty mark is drawn outside the penalty area.

	RoboCup Changes and Comments
RC-1.4: **Penalty Area**	For RoboCup, replace "*16.5m*" by "**2.25m**", and "*11m*" by "**3m**". In RoboCup, the penalty spot is outside the penalty area. No circle arc is used in RoboCup.

FL 1.5 Flag-posts

A flag-post, not less than 1.5m (5ft) high, with a non-pointed top and a flag is placed at each corner. flag posts may also be placed at each end of the halfway line, not less than 1m (1yd) outside the touch line.

	RoboCup Changes and Comments
RC-1.5: **Flag-posts**	No flag-posts are currently used in RoboCup.

FL 1.6 The Corner Arc

A quarter circle line with a radius of 1m (1yd) from each corner flag-post is drawn inside the field of play.

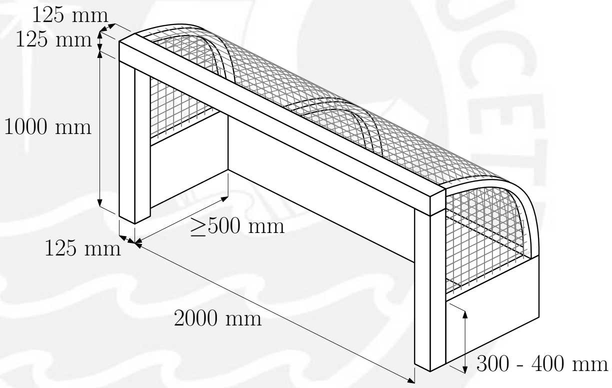
	RoboCup Changes and Comments
RC-1.6: **Corner Arc**	For RoboCup, replace "*1m*" with "**0.75m**".

FL 1.7 Goals

Goals must be placed on the center of each goal line. They consist of two upright posts equidistant from the corner flag-posts and joined at the top by a horizontal crossbar. The distance between the posts is 7.32m (8yds) and the distance from the lower edge of the crossbar to the ground is 2.44m (8ft).

Both goalposts and the crossbar have the same width and depth which do not exceed 12.5cm (5ins). The goal lines are the same width as that of the goalposts and the crossbar. Nets may be attached to the goals and the ground behind the goal, provided that they are properly supported and do not interfere with the goalkeeper.

The goalposts and crossbars must be white.

	RoboCup Changes and Comments
RC-1.7: **Goals**	For RoboCup, replace "*7.32m*" with "**2m**" and "*2.44m*" with "**1m**". A net is spawn between the crossbar and goal posts to the safety boundary of the field. To avoid direct contact of the net with parts of the robots (wheels, kicking device, etc.), the lower part of the net must be covered over a height between 30 and 40cm as a safety zone. This safety zone has to be done by the local organizer and may vary in each tournament. The inside depth of the goal is at least 0.5m (see image bellow). The goalposts the crossbar and the lower inside covered part are painted white. See also COMPETITION RULE 4 for color samples. 

FL 1.8 Safety

Goals must be anchored securely to the ground. Portable goals may only be used if they satisfy this requirement.

Decisions of the International F.A. Board

Decision 1: If the crossbar becomes displaced or broken, play is stopped until it has been repaired or replaced in position. If a repair is not possible, the match is abandoned. The use of a rope to replace the crossbar is not permitted. If the crossbar can be repaired, the match is restarted with a dropped ball at the place where the ball was located when play was stopped (see FIFA LAW 8).

Decision 2: Goalposts and crossbars must be made of wood, metal or other approved material. Their shape may be square, rectangular, round or elliptical and they must not be dangerous to players.

Decision 3: No kind of commercial advertising, whether real or virtual, is permitted on the field of play and field equipment (including the goal nets and the areas they enclose) from the time the teams enter the field of play until they have left it at half-time and from the time the teams re-enter the field of play until the end of the match. In particular, no advertising material of any kind may be displayed on goals, nets, flag-posts or their flags. No extraneous equipment (cameras, microphones, etc.) may be attached to these items.

Decision 4: The reproduction, whether real or virtual, of representative logos or emblems of FIFA, confederations, national associations, leagues, clubs or other bodies, is forbidden on the field of play and field equipment (including the goal nets and the areas they enclose) during playing time, as described in Decision 3.

Decision 5: A mark may be made off the field of play, 9.15 meters (10 yds) from the corner arc and at right angles to the goal lines to ensure that this distance is observed when a corner kick is being taken.

	RoboCup Changes and Comments
RC-Decision 5:	FIFA Decision 5 applies with 2m and 3m distances in RoboCup.

FIFA LAW 2 – The Ball

FL 2.1 Qualities and Measurements

The ball is

- spherical,
- made of leather or other suitable material,
- of a circumference of not more than 70cm (28ins) and not less than 68cm (27ins),
- not more than 450g (16oz) in weight and not less than 410g (14oz) at the start of the match,
- of a pressure equal to 0.6–1.1 atmosphere (600 - 1100g/cm^2) at sea level (8.5lbs/sqin – 15.6 lbs/sqin).

RoboCup	**RoboCup Changes and Comments**
RC-2.1: Qualities and Measurements	Each team has to be able to play with a defined FIFA standard size 5 football at any time. The official tournament ball will be announced at least one month before the tournament and won't mainly consist of the colors black, white or green (except in the technical challenges). The vision system of the robots has to be robust enough to handle any above mentioned ball without recalibration. The ball should not be worn down too much and both team leaders have to find an agreement before a match about which of the official balls they want to use. Else the referee decides about the ball.

FL 2.2 Replacement of a Defective Ball

If the ball bursts or becomes defective during the course of a match:

- The match is stopped.
- The match is restarted by dropping the replacement ball at the place where the first ball became defective (see FIFA LAW 8).

If the ball bursts or becomes defective whilst not in play at a kick-off, goal kick, corner kick, free kick, penalty kick or throw-in:

- The match is restarted accordingly.

The ball may not be changed during the match without the authority of the referee.

Decisions of the International F.A. Board

Decision 1: In competition matches, only footballs which meet the minimum technical requirements stipulated in FIFA LAW 2) are permitted for use. In FIFA competition matches, and in competition matches organized under the auspices of the confederations, acceptance of a football for use is conditional upon the football bearing one of the following three designations: The official "FIFA APPROVED" logo, or the official "FIFA INSPECTED" logo, or the reference "INTERNATIONAL MATCHBALL STANDARD". Such a designation on a football indicates that it has been tested officially and found to be in compliance with specific technical requirements, different for each category and additional to the minimum specifications stipulated in FIFA LAW 2). The list of the additional requirements specific to each of the respective categories must be approved by the International F.A. Board. The institutes conducting the tests are subject to the approval of FIFA. National association competitions may require the use of balls bearing any one of these three designations. In all other matches the ball used must satisfy the requirements of FIFA LAW 2.

Decision 2: In FIFA competition matches and in competition matches organized under the auspices of the confederations and national associations, no kind of commercial advertising on the ball is permitted, except for the emblem of the competition, the competition organizer and the authorized trademark of the manufacturer. The competition regulations may restrict the size and number of such markings.

RoboCup	**RoboCup Changes and Comments**
RC-Decision 2:	The organizing committee of a tournament is responsible for approving the balls to be used, including any kind of advertisement, logo, or emblem on the ball.

FIFA LAW 3 – The Number of Players

FL 3.1 Players

A match is played by two teams, each consisting of not more than eleven players, one of whom is the goalkeeper. A match may not start if either team consists of fewer than seven players.

RoboCup	RoboCup Changes and Comments
RC-3.1: Players	A match is played by two teams, each consisting of not more than five players, one of whom is the goalkeeper. A match may not start if either team consists of fewer than two players. If the number of players of a team falls under 2 during a match, then the match will still continue. However, if the number of players of a team falls to zero, the match will be ended and competition rule 3.7 will be applied.
RC-3.1.1: Incapable Players	Players not capable of play, e.g. players not able to move, or players with defective or malfunctioning sensing and/or actuating systems, are not permitted to participate in the game. It is up to the referee to judge whether a player is capable of play. The referee may ask the team leader of a player suspected to be incapable of play to demonstrate playing ability at any time, in particular before and after a game, during half-time, and during any stoppage of the game.

FL 3.2 Official Competitions

Up to a maximum of three substitutes may be used in any match played in an official competition organized under the auspices of FIFA, the confederations or the national associations. The rules of the competition must state how many substitutes may be nominated, from three up to a maximum of seven.

RoboCup	RoboCup Changes and Comments
RC-3.2: Official Competitions	In RoboCup currently no substitutions are used.

FL 3.3 Other Matches

In other matches, up to five substitutes may be used, provided that:

- The teams concerned reach agreement on a maximum number.
- The referee is informed before the match.

If the referee is not informed, or if no agreement is reached before the start of the match, no more than three substitutes are allowed.

FL 3.4 All Matches

In all matches the names of the substitutes must be given to the referee prior to the start of the match. Substitutes not so named may not take part in the match.

FL 3.5 Substitution Procedure

To replace a player by a substitute, the following conditions must be observed:

- The referee is informed before any proposed substitution is made.

- A substitute only enters the field of play after the player being replaced has left and after receiving a signal from the referee.
- A substitute only enters the field of play at the halfway line and during a stoppage in the match.
- A substitution is completed when a substitute enters the field of play.
- From that moment, the substitute becomes a player and the player he has replaced ceases to be a player.
- A player who has been replaced takes no further part in the match.
- All substitutes are subject to the authority and jurisdiction of the referee, whether called upon to play or not.

FL 3.6 Changing the Goalkeeper

Any of the other players may change places with the goalkeeper, provided that:

- The referee is informed before the change is made.
- The change is made during a stoppage in the match.

RoboCup	**RoboCup Changes and Comments**
RC-3.5: Goalie Substitution Procedure	In RoboCup it is allowed to exchange the goalie with a field player. This has to be done during a break of the game, usually when the goalie is taken from the field for repair.

FL 3.7 Infringements/Sanctions

If a substitute enters the field of play without the referee's permission:

- Play is stopped.
- The substitute is cautioned, shown the yellow card and required to leave the field of play.
- Play is restarted with a dropped ball at the place it was located when play was stopped (see FIFA LAW 8).

If a player changes places with the goalkeeper without the referee's permission before the change is made:

- Play continues.
- The players concerned are cautioned and shown the yellow card when the ball is next out of play.

For any other infringements of this Law:

- The players concerned are cautioned and shown the yellow card.

FL 3.8 Restart of Play

If play is stopped by the referee to administer a caution:

- The match is restarted by an indirect free kick, to be taken by a player of the opposing team from the place where the ball was located when play was stopped (see FIFA LAW 8).

FL 3.9 Players and Substitutes Sent Off

A player who has been sent off before the kick-off may be replaced only by one of the named substitutes. A named substitute who has been sent off, either before the kick-off or after play has started, may not be replaced.

FL 3.10 Decisions of the International F.A. Board

Decision 1: Subject to the overriding conditions of FIFA LAW 3, the minimum number of players in a team is left to the discretion of national associations. The Board is of the opinion, however, that a match should not continue if there are fewer than seven players in either team.

Decision 2: The coach may convey tactical instructions to the players during the match. He and the other officials must remain within the confines of the technical area, where such an area is provided, and they must behave in a responsible manner.

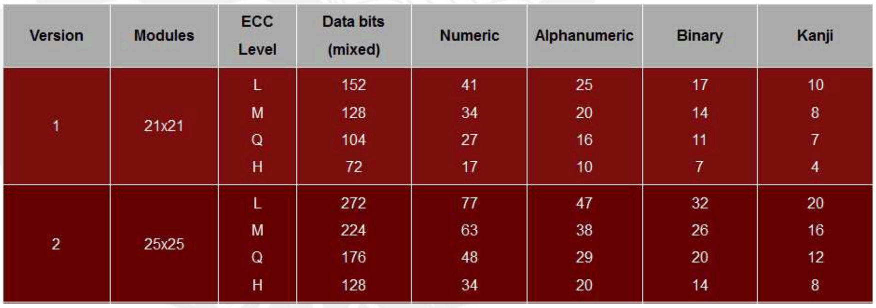
	RoboCup Changes and Comments
RC-Decision 2:	RoboCup players must play autonomously. Coaching and any kind of human interference with robots, with or without technical means, is not allowed, except when according to the ruling RC-Decision 2.1 or where otherwise specified in the Laws. Human interference is only allowed for substitutes and robots outside of the playground, and only if the robots are inactive, in particular, if they do not send any kind of signals, including wireless communications.

RoboCup Changes and Comments

RC-Decision 2.1: High level human coaching is allowed. For that purpose each team may use a set of paper boards with QR Codes to be directly interpreted by the robots. High level coaching has to comply the following rules:

- boards can not be larger than 30 x 30 cm;
- only robots that are in the field can be coached;
- each team will designate a member to perform coaching;
- coaching is only allowed from the team leader position, in front of the teams' base station pc;
- the robot(s) that is/are being coached should remain within the field of play, and cannot be touched by human team members;
- the human that is coaching should stay always outside of the field of play;
- no electronic device, other than electronic devices that are mounted on the robot itself, can be used to transfer coaching instructions to the robot;
- coaching can take place only during 'dead time' (i.e., the 10 seconds between a stop and a start by the assistant referee).

QR Codes can only be of the version 1 or 2. Any level of Error Correction can be used and any type of data can be encoded. In the figure below a summary of version 1 and 2 of QR Codes can be seen. Use of other higher versions of the QR Codes as well as Micro QR Code, iQr Code, SQR Code or LogoQ is strictly forbidden.



Version	Modules	ECC Level	Data bits (mixed)	Numeric	Alphanumeric	Binary	Kanji
1	21x21	L	152	41	25	17	10
		M	128	34	20	14	8
		Q	104	27	16	11	7
		H	72	17	10	7	4
2	25x25	L	272	77	47	32	20
		M	224	63	38	26	16
		Q	176	48	29	20	12
		H	128	34	20	14	8

NOTE: Complementary information regarding high level coaching may be released until February 28, 2014.

FIFA LAW 4 – The Players' Equipment

RoboCup	**RoboCup Changes and Comments**
RC-4.0.1: **Design Guideline**	Robots for playing soccer must be designed such that they are **both robust and safe**. Both terms are subsequently explained. To support the TC during the verification of the robot size and of the ball manipulation devices, the adequate measurements (e.g. construction plans, etc.) of the robots have to be published before the tournaments. This is regulated in more detail during the qualification process of a tournament.

FL 4.1 Safety

A player must not use equipment or wear anything which is dangerous to himself or another player (including any kind of jewellery).

RoboCup	**RoboCup Changes and Comments**
RC-4.1: **Safety**	Robot soccer players must be built such that they are **safe**. Safe means that robots do not damage other robots or any objects of the playground, or pose a threat to the audience, or to the referees, or to human team members. In particular, **the design of the robots should ensure that "Fouls and Misconduct" (FIFA LAW 12)** are avoided. Each robot must have a a **vertically continuous** safety border, at least 1 cm thick and 6 cm high, made out of soft material, which is added around the bottom of the robot. This soft material layer should be supported on the back over its complete height. Borders made of independent non connected pieces, or that only partially covers the outer limits of the robot (with the exception of the natural openings such as the ball engaging area), are not allowed and will be checked upon during technical verifications or at any time during a game by the referee. This border must be well attached to the robot and may not fall down, partially or totally, during the game. If such a thing happens, the robot must be removed from the field for repair. Anyone is allowed to take whatever action that seems necessary to prevent a robot from causing urgent danger. This includes lifting the robot and/or switching it off. Teams must provide an emergency stop button on their robot that interrupts all actuation.
RC-4.1.1: **Jamming**	Robots must be designed and programmed such that they try to avoid interference concerning the operation of sensor systems and/or communication devices. The use of particular equipment which may cause interference of communication and/or sensors must be reported to the league organizing committee of a tournament and eventually negotiated between two teams before a match. If a team uses communications and sensors other than those previously declared to the tournament committee and/or the opponent, the game may be forfeited, and the league organizing committee may exclude the robots from further participation.

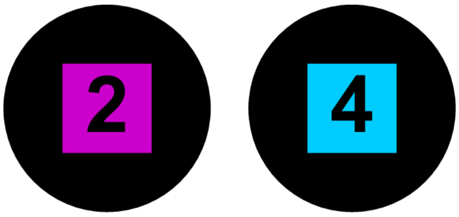
	RoboCup Changes and Comments
RC-4.1.2: External Boundary Area	All robots must be able to detect those situations where they are currently placed outside the external line of the field of play (inside the external 1.5m wide boundary area). Although robots are allowed to use this area at its own discretion they may not, either intentionally or not, crash against the field Safety Boundary. A robot is considered to have crashed against the field Safety Boundary if its speed is high enough to potentially damage this boundary. In any case, if it is clear that the robot is not making an attempt to stop, and hits the Safety Boundary even at low speed, this is considered to be a crash. It is up to the referee to judge those situations and call a free kick against the offending team whenever he considers a crash has occurred. It is advisable that whenever a robot, or set of robots, are chasing the ball, they stop their movement as soon as they have detected that the ball is outside the field of game. NOTE: If a robot is pushed against the Safety Boundary by an opponent robot, a pushing fault will be called instead.
RC-4.1.3: Exclusion	Robots that violate the above conditions, in particular if they threaten to seriously damage opponents or pose a threat to the audience and/or referees and/or human team members, may be excluded from play in a tournament by the league organizing committee.

FL 4.2 Basic Equipment

The basic compulsory equipment of a player is:

- a jersey or shirt,
- shorts - if thermal undershorts are worn, they are of the same main color as the shorts,
- stockings,
- shin-guards,
- footwear.

	RoboCup Changes and Comments
RC-4.2.0: **Robot Size**	The size of each robot player must obey the following constraints: 1. Each robot must possess a configuration of itself and its actuators, where the projection of the robot's shape onto the floor fits into a square of size at least 30cm × 30cm and at most 52cm × 52cm. 2. The usual field player has to keep at any time the size limit of 52cm × 52cm. 3. The goalie is allowed to increase his size instantaneously (at most 1 second) up to 60cm × 60cm width or 90cm height if the goal is endangered by an approaching ball. The goalie resolves on this situation by itself, but he is only allowed to increase his size again after a pause of 4 seconds and having completely reduced his size to the normal state before the increase. Additionally, it is only allowed to increase its size instantaneously in one direction (left, right or upwards). 4. The robot's height must be at least 40 cm and at most 80 cm (exception goalie: 90 cm during extended phase). 5. The field players may never exceed the 80 cm height limit. 6. The size of the robots will be checked before the tournament by the TC. If a robot doesn't match the described limits it will be excluded from the competition.
RC-4.2.1: **Robot Shape**	Any shape is allowed as long as the size restrictions are not violated. Robots may exhibit concavities in their shape or may dynamically change shape, provided that the Laws concerning "Fouls and Misconduct" (FIFA LAW 12) are not violated.
RC-4.2.2: **Robot Weight**	The maximum weight of a robot is 40 kg.
RC-4.2.3: **Robot Colours**	The base color of a robot's body must be black. The paint or used material must be **matte** in order to minimize reflectivity. **Note 1:** This law does not mean that your robot must be completely black; for example, one cannot paint the lens of a camera. However, every team is expected to try hard to hide non-black parts of the robot as much as possible, especially parts that have colors used for the ball or the field of play. **Note 2:** Teams should avoid using any kind of shiny material for robot surfaces. The league committee may exclude robots that do not conform with colouring laws.
RC-4.2.4: **Robot Markers**	A robot must have markings in order to be recognized by other robots and to be distinguished by the referee. Each robot must carry color markers, number markers, and top markers. Robots not carrying all markers **are not eligible to play**.

	RoboCup Changes and Comments
RC-4.2.4.1: **Colour Markers**	Colour markers should be designed as follows: • Above 30cm and below 60cm, color markers must be present and visible from all sides. • A color marker can be any shape. • The height and width of a color marker must be greater than or equal to 10 cm in any direction. • The marker color must be one of the two predetermined official colors. The official colors are magenta and cyan. Please use the colors defined by the RAL codes in the competition rule 4. It is the responsibility of team leaders to obtain samples of marker colors and to ensure that their team color markers are as close as possible to the official colors. The Middle Size League Technical Committee may request to build new color markers on site, if the team's color markers do not sufficiently comply with the official colors. • Every team must provide the markers in each team color and attach one of them prior to each game. • See also COMPETITION RULE 4 for colour examples.
RC-4.2.4.2: **Number Markers**	A number marking should be designed as follows: • Each robot must carry a number (consisting of two digits at most), in black digits of height no smaller than 8 cm. • The number will be fixed on the color markers and must at least be visible from all four major sides (front, back, left and right) of the robot. • The number 1 is reserved for goal keepers. • The number marking must be easily visible for the referee, other humans and robots from all sides. • The number markers of the robots will be checked before the tournament by the TC. If the markers do not comply with the above rules, the team will be requested to build new number markers on site.
RC-4.2.4.3: **Top Markers**	Each robot must carry a top marker as follows (see figure below): • The top marker is a black circle with a diameter of 20cm. • In the center of the circle, a square in the respective team color and of at least 8cm side length must be present. • On top the coloured square a robot's number marker must be present. 

	RoboCup Changes and Comments
RC-4.2.5: Communications	Communication between the robots of a team using wireless links is allowed according to the following rules. Communication between the robots and one remote computer system (herein after referred as Base Station) is also allowed, provided that human interference is excluded. Robots may receive data or commands from this remote computer, as long as these does not include any further information gained by non-robot sensors (e.g. position of the robot itself, or teammates, or opponents, or the ball on the field). It is especially allowed to fuse data on the external computer, if that data is gained only by robots. Wireless communication equipment satisfying IEEE 802.11a and/or IEEE 802.11b specifications is allowed. Use of any other kind of wireless communications using these or other frequencies is explicitly forbidden. All communications between robots, as well as between robots and the Base Station, must be established through one of the two Access Points available at the field of game (either in 'a' or in 'b' mode), and provided by the organization. Use of ad hoc wireless networking is strictly forbidden. Teams may use either unicast or multicast wireless communication modes. Use of broadcast is strictly forbidden. Unicast and IPv4 multicast IP addresses are defined for each team in these rules (see bellow). Teams may not use any IP addresses other than those assigned to them. Although robots may send arbitrary kinds and amounts of data between each other and the Base Station, bandwidth restrictions will apply in order to assure a fair game. All competing teams should have the same network limits, whichever IEEE 802.11 mode is being used. This way, the slower mode (IEEE 802.11b specification) is the one that actually limits the amount of data that can be transmitted. Each team is then allowed to use, at most, 20% of the bandwidth provided by the IEEE 802.11b Access Point. Therefore, 2.2 Megabits/second is the actual maximum bit rate that can be used by any team during the tournament. Apart from the communication equipment placed in the robots, no other team computers or equipment may use any form of wireless communication. In particular, it is mandatory that the team's Base Station Computer has its wireless support turned off. No Access Points, other than those provided by the organization, and the ones used by the robots that are currently playing may be turned on during the tournament games. Each team has to provide, together with other team qualification materials, a list of all MAC addresses they intend to use during the tournament, with explicit indication of those that will be used for wireless communication.

RoboCup Changes and Comments

RC-4.2.5: Communications

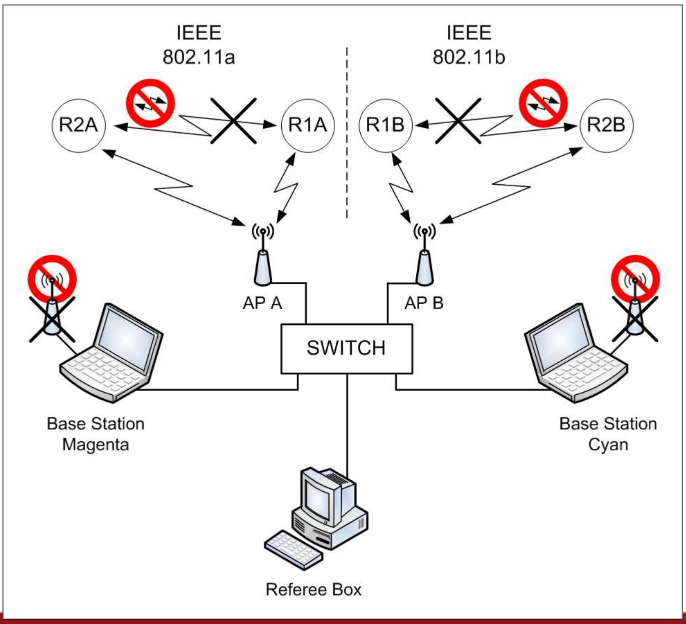
Competition setup

Each field of the competition will be equipped with the following base elements provided by the organization:

- Two Access Points. One working in IEEE 802.11a and other working in IEEE 802.11b. These Access Points may or may not be included in a single piece of equipment.
- One computer for running the Referee Box software
- Two LCD screens for Base Station visualization. As defined elsewhere, Base Station laptops must have their covers closed during the entire game.

Both Access Points will connect to a wired network by means of one or two switches.
Both the Referee Box and the team Base Stations must also be connected by cable to the wired network.
Each team has to design their software in such a way that it is possible for them to use only one Base Station to manage a game.
Commands from the Referee Box will be sent to the team's Base Station using the wired connection. It is the team responsibility to re-send these commands to their robots on the field.
Network settings, during the competition, will be as follows:

- Password for connecting to the APs may be turned on. If that's the case it will be disclosed to the teams at the beginning of the tournament.
- WEP encryption is turned off.
- Broadcast of SSID is turned on.
- Subnet mask normal PC: 255.255.255.0.
- Subnet mask of a PC connected to the RefBox: 255.255.0.0.
- Access Point Beacon Interval should be set to 20-30.
- Access Point DTIM Interval should be set to 2-3.
- Access Point power save mode is disabled.



RoboCup	**RoboCup Changes and Comments**
RC-4.2.5: **Communications**	**Technical verifications and sanctions** During technical verifications teams must be prepared to demonstrate and explain their communication setup and software to the MSL Technical Committee. This will include network configuration and bandwidth usage. Furthermore, team robots must be placed in the field and respond to basic Referee Box commands. The Network Monitor software will be used to verify that the team communication setup is in accordance to the rules. Teams that fail to comply with the current communications rules will be asked by the Technical Committee to re-adjust their software and setup in order to correct the detected incompatibilities. If the team fails a second technical verification, it may be excluded from playing in the tournament by the league organizing committee. Power emitted by any of the team's robots wireless equipment must be limited in order to ensure that all teams have the same conditions for wireless communications. To ensure that, during technical verifications, a Fluke Wi-Fi AirCheck(*Trade Mark*) with and external directional antenna will be used to measure the RSSI (expressed in dBm) in the following conditions: • team robots will be placed along the mid line and connected to the field router; • the measurement equipment will be placed over the goal line with the antenna pointed towards the robots (9 meter distance); • the maximum received power may not exceed -40dBm. Teams failing to comply with this limit will be requested to re-adjust the power of their WiFi equipment. Only after that they will be considered able to enter the competition. If, during or after a match, the Network Monitor shows a clear violation of the rules either by the playing teams, or by any other MSL teams in the neighbourhood, the reported offending team will be awarded a warning by the technical committee. A second warning, issued to the same team during the tournament, may lead to the team exclusion, by decision of the league organizing committee, under recommendation of the technical committee.
RC-4.2.6: **Sensing Systems**	Any sensing system is allowed as long as the following constraints are met: 1. All parts of the sensing system (i.e .the actual sensing device and, if applicable, a signal emitting device) must be on the robots. 2. There may be no manipulation of the environment, such as placing specific markers as landmarks.
RC-4.2.7: **Ball Handling Mechanisms**	Robots may have special devices for ball handling. Ball handling devices must be designed such that they are safe. The robot's use of ball handling devices must comply with the Laws set forth in "Fouls and Misconduct" (FIFA LAW 12).

FL 4.3 Shin-guards

- are covered entirely by the stockings.
- are made of a suitable material (rubber, plastic, or similar substances).

- provide a reasonable degree of protection.

	RoboCup Changes and Comments
RC-4.3.1: Robustness	Robot soccer players must be built such that they are **robust**. Robust means that the physical integrity of the robot is not endangered by incidental, accidental, or intentional collisions with the ball or objects of the field or other robots. The robot's sensing systems and software should be able to handle potentially significant levels of noise caused by other sources, such as other robots, game officials, team members, spectators, or the media. Robots are allowed to kick a ball upward with no penalty. This means that the robots should be build strong enough to tolerate it.

FL 4.4 Goalkeepers

- wear colors which distinguish them from the other players, the referee and the assistant referees

	RoboCup Changes and Comments
RC-4.4: Goalkeepers	Goalkeepers must obey the same colouring and marker constraints as the other robots in their team. The number on the top marker must indicate that a player is a goal keeper. If a normal player replaces the goal keeper at the beginning of the game, this player must exhibit the number 1 in order to clarify its function in the game. If the replacement occurs during the game the replacing goal keeper does not have to wear the side color, number marker or top marker of the regular goal keeper. If a goal keeper is removed from the field during a game stoppage, then, just after the ending of the stoppage, it can be automatically replaced by one of the field players. The robot that enters the goal area first, automatically become the new goalie, and stays as that until number 1 player re-enters the field or until it is removed from the field itself. If the number 1 player later re-enters the game, the replacing goalie must resume its role as field player.

FL 4.5 Infringements/Sanctions

For any infringement of this Law:

- Play need not be stopped.
- The player at fault is instructed by the referee to leave the field of play to correct his equipment.
- The player leaves the field of play when the ball next ceases to be in play, unless he has already corrected his equipment.
- Any player required to leave the field of play to correct his equipment does not re-enter without the referee's permission.
- The referee checks that the player's equipment is correct before allowing him to re-enter the field of play.
- The player is only allowed to re-enter the field of play when the ball is out of play.

A player who has been required to leave the field of play because of an infringement of this Law and who enters (or re-enters) the field of play without the referee's permission is cautioned and shown the yellow card.

RoboCup	**RoboCup Changes and Comments**
RC-4.5.1: **Repair of Robots**	Team leaders may ask the referee for permission to remove a player from the field, if there is a problem with the players hardware and/or software. If the referee gives permission to remove a player, one human team member, who must be properly dressed and who has been identified to the referee before the start of the game, may enter the field and remove a robot only during a game stoppage. Robots must be repaired outside of the safety boundary around the field. Checking that the player's equipment is correct may also be done by assistant referees. After a player has been repaired or a players equipment has been corrected the player is allowed to enter the field only during a game stoppage and after having received a corresponding signal from the RefBox over the network socket. The referee will signal the person at the RefBox when a player is removed from the field. The person at the RefBox will press a button for the team which indicates that a player is removed. After 30s the RefBox will send automatically a signal over the network socket that a robot can enter the field again during the next game stoppage. The RefBox will show when the 30s period is over. If a robot re-enters the field before the 30s are passed or when the game is not stopped, the opponent team will be awarded a free kick executed on the kick-off point. The robot that illegally entered the field has to be removed again and the 30s period will be restarted. This rule doesn't apply to a single particular robot. If a robot is removed from the field, then no other robot from the same team can re-enter the field before the 30s are passed, the RefBox signal is received and the game is stopped. If another robot from the same team is removed during the on going 30s period in the next game stoppage, this time is restarted and, for another 30s, no robot of the same team can re-enter the field.

FL 4.6 Restart of Play

If play is stopped by the referee to administer a caution:

- the match is restarted by an indirect free kick taken by a player of the opposing side, from the place where the ball was located when the referee stopped the match.

FIFA LAW 5 – The Referee

FL 5.1 The Authority of the Referee

Each match is controlled by a referee who has full authority to enforce the Laws of the Game in connection with the match to which he has been appointed.

FL 5.2 Powers and Duties

The Referee:

- enforces the Laws of the Game.
- controls the match in co-operation with the assistant referees and, where applicable, with the fourth official.
- ensures that the ball meets the requirements of FIFA LAW 2.
- ensures that the players' equipment meets the requirements of FIFA LAW 4.
- acts as timekeeper and keeps a record of the match.
- stops, suspends or terminates the match, at his discretion, for any infringements of the Laws.
- stops, suspends or terminates the match because of outside interference of any kind.
- stops the match if, in his opinion, a player is seriously injured and ensures that he is removed from the field of play.
- allows play to continue until the ball is out of play if a player is, in his opinion, only slightly injured.
- ensures that any player bleeding from a wound leaves the field of play. The player may only return on receiving a signal from the referee, who must be satisfied that the bleeding has stopped.
- allows play to continue when the team against which an offence has been committed will benefit from such an advantage and penalizes the original offence if the anticipated advantage does not ensue at that time.
- punishes the more serious offence when a player commits more than one offence at the same time.
- takes disciplinary action against players guilty of cautionable and sending-off offences. He is not obliged to take this action immediately but must do so when the ball next goes out of play.
- takes action against team officials who fail to conduct themselves in a responsible manner and may at his discretion, expel them from the field of play and its immediate surrounds.
- acts on the advice of assistant referees regarding incidents which he has not seen.
- ensures that no unauthorized persons enter the field of play.
- restarts the match after it has been stopped.
- provides the appropriate authorities with a match report which includes information on any disciplinary action taken against players, and/or team officials and any other incidents which occurred before, during or after the match.

RoboCup	**RoboCup Changes and Comments**
RC-5.3: Powers and Duties	In RoboCup, some referee duties like time keeping and keeping a record of the match may be delegated to one of the assistant referees.

RoboCup	**RoboCup Changes and Comments**
RC-5.3.1: **Referee Box**	In RoboCup, assisting technology is used to support the referee, in particular for conveying referee decisions to robot players and for maintaining a record of the game. Such assisting technology includes a referee box and possibly other sorts of technology. In particular, whenever the Laws of the Game specify that the referee is giving a signal, the referee box protocol specifies the communication of one or more messages to the team remote computer system. Operation of the referee box is delegated to an assistant referee.
RC-5.3.2: **Permission to stop the robots**	Whenever a robot shows a behaviour which is clearly dangerous either to the opponent robots or to spectators, a single human team-member is allowed to enter the field without permission of the referee to stop the robot by means of its mandatory emergency stop button. If a robot is stopped in this way, the game is also stopped by the referee and resumed with a free-kick for the other team. This free-kick will be taken from the position where the ball was when the robot was stopped, or from one of the closest restart points.

FL 5.3 Decisions of the Referee

The decisions of the referee regarding facts connected with play are final. The referee may only change a decision on realizing that it is incorrect or, at his discretion, on the advice of an assistant referee, provided that he has not restarted play.

Decisions of the International F.A. Board

Decision 1: A referee (or where applicable, an assistant referee or fourth official) is not held liable for:

- any kind of injury suffered by a player, official or spectator,
- any damage to property of any kind,
- any other loss suffered by any individual, club, company, association or other body, which is due or which may be due to any decision which he may take under the terms of the Laws of the Game or in respect of the normal procedures required to hold, play and control a match.

This may include:

- a decision that the condition of the field of play or its surrounds or that the weather conditions are such as to allow or not to allow a match to take place,
- a decision to abandon a match for whatever reason,
- a decision as to the condition of the fixtures or equipment used during a match including the goalposts, crossbar, flag posts and the ball,
- a decision to stop or not to stop a match due to spectator interference or any problem in the spectator area,
- a decision to stop or not to stop play to allow an injured player to be removed from the field of play for treatment,
- a decision to request or insist that an injured player be removed from the field of play for treatment,
- a decision to allow or not to allow a player to wear certain apparel or equipment,
- a decision (in so far as this may be his responsibility) to allow or not to allow any persons (including team or stadium officials, security officers, photographers or other media representatives) to be present in the vicinity of the field of play,
- any other decision which he may take in accordance with the Laws of the Game or in conformity with his duties under the terms of FIFA, confederation, national association or league rules or regulations under which the match is played.

Decision 2: In tournaments or competitions where a fourth official is appointed, his role and duties must be in accordance with the guidelines approved by the International F.A. Board.

Decision 3: Facts connected with play shall include whether a goal is scored or not and the result of the match.

FIFA LAW 6 – The Assistant Referees

FL 6.1 Duties

Two assistant referees are appointed whose duties, subject to the decision of the referee, are to indicate

- when the whole of the ball has passed out of the field of play,
- which side is entitled to a corner kick, goal kick or throw-in,
- when a player may be penalized for being in an offside position,
- when a substitution is requested,
- when misconduct or any other incident has occurred out of the view of the referee.

RoboCup	**RoboCup Changes and Comments**
RC-6.1: **Duties**	In RoboCup, one or more assistants may be appointed for a match. The recommendation is to have three assistant referees. One of the assistant referees should be responsible for time keeping and keeping a game record. The referee may assign assistant referees additional duties aside of those specified in the FIFA Law.

FL 6.2 Assistance

The assistant referees also assist the referee to control the match in accordance with the Laws of the Game.

In the event of undue interference or improper conduct, the referee will relieve an assistant referee of his duties and make a report to the appropriate authorities.

FIFA LAW 7 – The Duration of the Match

FL 7.1 Periods of Play

The match lasts two equal periods of 45 minutes, unless otherwise mutually agreed between the referee and the two participating teams.
Any agreement to alter the periods of play (for example to reduce each half to 40 minutes because of insufficient light) must be made before the start of play and must comply with competition rules.

RoboCup	**RoboCup Changes and Comments**
RC-7.1: Periods of Play	In RoboCup, a match lasts two equal periods of 15 minutes (clock-time). For friendly games, referee and both teams may decide on different periods of play. In official tournaments, modifications of periods of play may be specified by the organizing committee.

FL 7.2 Half-Time Interval

Players are entitled to an interval at half-time. The half-time interval must not exceed 15 minutes. Competition rules must state the duration of the half-time interval. The duration of the half-time interval may be altered only with the consent of the referee.

RoboCup	**RoboCup Changes and Comments**
RC-7.2: Half-Time Interval	In RoboCup, the half-time interval must not exceed 10 minutes. The referee may limit the game to the first half if this time is exceeded by clear responsibility of one of the teams. In this case competition rule 3.7 will apply. Upon agreement with both team leaders, the referee may alternatively reduce the second half overall time.

FL 7.3 Allowance for Time Lost

Allowance is made in either period for all time lost through

- substitution(s),
- assessment of injury to players,
- removal of injured players from the field of play for treatment,
- wasting time,
- any other cause.

The allowance for time lost is at the discretion of the referee.

FL 7.4 Penalty Kick

If a penalty kick has to be taken or retaken, the duration of either half is extended until the penalty kick is completed.

FL 7.5 Extra Time

Competition rules may provide for two further equal periods to be played. The conditions of FIFA LAW 8 will apply.

FL 7.6 Abandoned Match

An abandoned match is replayed unless the competition rules provide otherwise.

FIFA LAW 8 – The Start and Restart of Play

FL 8.1 Preliminaries

A coin is tossed and the team which wins the toss decides which goal it will attack in the first half of the match.
The other team takes the kick-off to start the match.
The team which wins the toss takes the kick-off to start the second half of the match.
In the second half of the match the teams change ends and attack the opposite goals.

RoboCup	**RoboCup Changes and Comments**
RC-8.1.1: **Start Delay**	For RoboCup, a match must start at the scheduled time. In exceptional situations only, the referee may re-adjust the time for starting the game in accordance with both team leaders.
RC-8.1.2: **Remote Start**	All robots of a team are started (and stopped) by receiving a signal through wireless communication from outside the field.

FL 8.2 Kick-off

A kick-off is a way of starting or restarting play

- at the start of the match,
- after a goal has been scored,
- at the start of the second half of the match,
- at the start of each period of extra time, where applicable.

A goal may be scored directly from the kick-off.

FL 8.3 Procedure (for kick-off)

- All players are in their own half of the field.
- The opponents of the team taking the kick-off are at least 9.15m (10yds) from the ball until it is in play.
- The ball is stationary on the center mark.
- The referee gives a signal.
- The ball has to be kicked forward into the side of the defending team.
- The ball is in play when it is kicked and moves forward.
- Unless the ball is in play, all players stay in their own half of the field.
- The kicker does not touch the ball a second time until it has touched another player.

After a team scores a goal, the kick-off is taken by the other team.

	RoboCup Changes and Comments
RC-8.3: Procedure (for Kick-Off)	For RoboCup, the following procedure is followed for kick-off: • All players are in their own half of the field, with the exception of the robot taking the kick, which may be partially inside the opponent half of the field • The opponents of the team taking the kick-off must remain at least 3m away from the ball until the ball is in play. • The robot of the attacking team that is taking the kick is positioned at the ball. • The players of the team taking the kick-off other than the kicking robot must remain at least 2m from the ball until the ball is in play. • No robot, except the kicking robot, is allowed to touch the ball until the ball is in play. • The ball is stationary on the center mark. • The referee gives a signal. • A player of the team who was awarded the kick-off kicks the ball. Kicking the ball into its own half of the field is also allowed. • The robot taking the kick should either use its kicker or one of its sides to instantaneously kick (i.e., without dribbling or dragging) the ball such that it travels freely over a distance of at least 0.5m. • The ball is in play immediately after being kicked. • After the kick, the attacking team is only allowed to touch the ball a second time after it moved over a distance of at least 0.5m. • A goal may be scored only when the ball was touched by another player of the same team. • When 10 seconds have passed and the ball wasn't kicked by the attacking team, the defending team can approach the ball and score a goal directly, even without any contact between the ball and any other player. However, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players. • If a robot of the attacking team except the kicking robot approaches the ball before the ball is in play, the kick-off will be awarded to the other team. The above mentioned 2m and 3m refers to the radius of a circle centered on the ball. The robots must be completely out of each circle respectively, depending on its status (attacking or defending). The referee must restart the game within 10 seconds after game stops.
RC-8.3.1: Positioning of Robots	During kick off, robots must autonomously reposition themselves in any position on the field that is consistent with "RC-8.3" FIFA LAW28. NOTE: In national or regional competitions only, the local organizing committee may decide if, during kick-off, teams are allowed to manually re-position the robots at the game restart points.

FL 8.4 Infringements/Sanctions (for kick-off)

If the kicker touches the ball a second time before it has touched another player:

- An indirect free kick is awarded to the opposing team to be taken from the place where the infringement occurred.

For any other infringement of the kick-off procedure:

- The kick-off is retaken.

RoboCup	**RoboCup Changes and Comments**
RC-8.4.1: Kicking Directly to the goal	In RoboCup, if the ball is kicked by the team that has kick-off and enters the goal without being touched by a second player of the same team before crossing the goal line, the goal is not scored and the kick-off is awarded to the opposing team. When 10 seconds have passed since the signal and the attacking team didn't touch the ball, a goal may be scored directly by the defending team. **NOTE:** Rules regarding the validation of scored goals, namely those established in RC-10 and RC-12, overrules or complements all others, including the above one and those defined for every other game restart situations.

FL 8.5 Dropped Ball

A dropped ball is a way of restarting the match after a temporary stoppage which becomes necessary, while the ball is in play, for any reason not mentioned elsewhere in the Laws of the Game.

RoboCup	**RoboCup Changes and Comments**
RC-8.5.1: Game Stuck	In RoboCup, the referee may call a game stuck situation if there is no progress of the game. The game is continued using the dropped ball procedure. The ball is placed at the point it was when the dropped ball was called.

FL 8.6 Procedure (for dropped ball)

The referee drops the ball at the place where it was located when play was stopped. Play restarts when the ball touches the ground.

	RoboCup Changes and Comments
RC-8.6: **Procedure** **(for Dropped Ball)**	In RoboCup, the following procedure is followed for dropped ball: • The referee gives a “stop” signal. • All players have to stop their movement. • The ball is stationary positioned in the place where it was located when the game was stopped. • The referee gives a “dropped ball” signal. • All players remain 1m away from the ball. One robot may stay anywhere inside the penalty area (except goal area) of its own team, even if the distance to the ball is shorter than 1m. • The referee gives a “start” signal. • The ball is in play immediately after the referee gives the signal. • In RoboCup a goal may not be scored directly from a dropped ball. To score a goal, the ball has to be touched by at least two robots (not necessarily of the same team). • See also ”RC-12.3.8 - Delay of game”. It is forbidden to reposition robots by hand or by any other means with the only exception of the use of high level coaching of the robots (see FL 3.10, RC-Decision 2.1). The referee may show a yellow card to the robot that doesn’t stay at least 1m from the ball, following the referees instructions more than twice consecutively. After that, if the robot doesn’t follow the position restrictions of the procedure, the referee may instruct the team to remove the robot from the field. The above mentioned 1m refers to the radius of a circle centered on the ball. The robots must be completely out of that circle. The referee must restart the game within 10 seconds after game stops.

FL 8.7 Infringements/Sanctions (for dropped ball)

The ball is dropped again:

- If it is touched by a player before it makes contact with the ground.
- If the ball leaves the field of play after it makes contact with the ground, without a player touching it.

RoboCup	**RoboCup Changes and Comments**
RC-8.7: Infringements/ Sanctions	If a player moves within 1m from the ball before the referee gives the signal, an indirect free kick is awarded to the opponent.

FL 8.8 Special Circumstances

A free kick awarded to the defending team inside its own goal area is taken from any point within the goal area.
An indirect free kick awarded to the attacking team in its opponents' goal area is taken from the goal area line parallel to the goal line at the point nearest to where the infringement occurred.
A dropped ball to restart the match after play has been temporarily stopped inside the goal area takes place on the goal area line parallel to the goal line at the point nearest to where the ball was located when play was stopped.

FIFA LAW 9 – The Ball In and Out of Play

FL 9.1 Ball Out of Play

The ball is out of play when:

- It has wholly crossed the goal line or touch line whether on the ground or in the air.
- Play has been stopped by the referee.

RoboCup	**RoboCup Changes and Comments**
RC-9.1.1: Dead Call	In RoboCup, a special “*dead call*” signal may be given by the referee, upon which all robots immediately have to cease operating any kind of actuator. The referee may signal a dead call at any time upon his discretion. In particular, the referee may signal a dead call whenever he considers it necessary to maintain and ensure safety and security of players, team members, referees, and spectators.
RC-9.1.2: Continuation after Dead Call	After a dead call, the game continues with a dropped ball at the position nearest to the ball location when the game was interrupted, except when the referee issued a different call prior to the dead call.

FL 9.2 Ball In Play

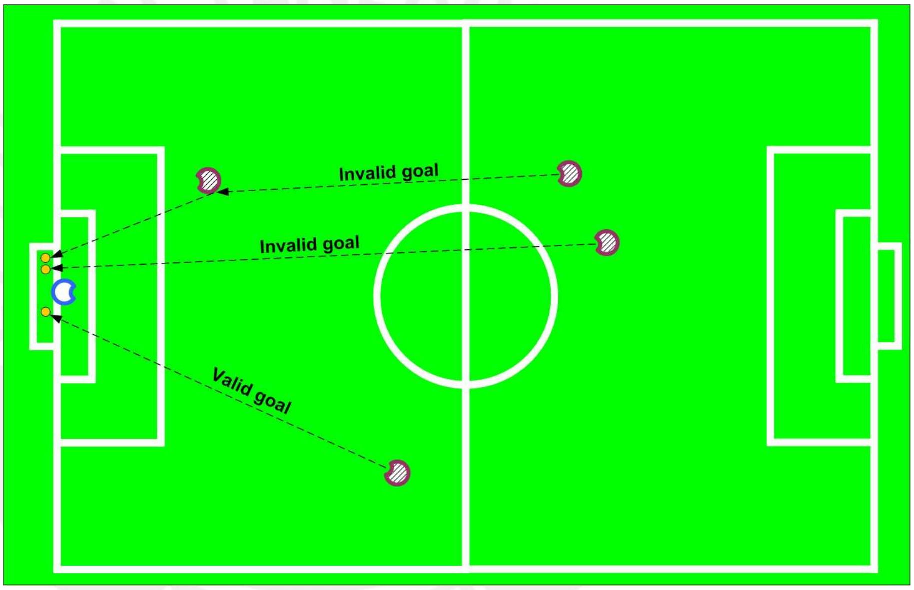
The ball is in play at all other times, including when:

- It rebounds from a goalpost, crossbar or corner flag-post and remains in the field of play.
- It rebounds from either the referee or an assistant referee when they are on the field of play.

FIFA LAW 10 – The Method of Scoring

FL 10.1 Goal Scored

A goal is scored when the whole of the ball passes over the goal line, between the goalposts and under the crossbar, provided that no infringement of the Laws of the Game has been committed previously by the team scoring the goal.

RoboCup	**RoboCup Changes and Comments**
RC-10: **Valid methods of scoring**	Any goal scored by a robot in the opponent goal will be only valid if the robot taking the kick is inside the opponent side of the field (example below). This does not apply if the attacking robot kicks into the goal of its own team. Regardless of this rule, all other rules related to indirect fouls procedures still apply. If a kick is taken from the robot own half of the field and touches another robot of the same team before entering the goal, this goal is still not valid even if the touched robot is in the opponent half of the field.  Furthermore, whenever the ball is regained by one of the teams within its own half of the field (either in game restart situations or during play on) a valid goal can only be scored after the ball has been received or touched by a team mate within the opponent side of the field and assuming that the ball has rolled freely for at least one meter before being received or touched. This rule does not apply if the ball is regained in the opponent side of the field during play on. In this particular case the robot regaining the ball may score directly as long as the kick is taken from within the opponent side. Regaining the ball implies that the ball was previously in possession of the opponent team (that is, the ball was previously being played by the opponent or the opponent had the faculty to play in a restart situation).

FL 10.2 Winning Team

The team scoring the greater number of goals during a match is the winner. If both teams score an equal number of goals, or if no goals are scored, the match is drawn.

FL 10.3 Competition Rules

For matches ending in a draw, Competition Rules may state provisions involving extra time, or other procedures approved by the International F.A. Board to determine the winner of a match.

FIFA LAW 11 – Offside

RoboCup	**RoboCup Changes and Comments**
RC-11: Offside	This law does currently not apply in RoboCup matches.

FL 11.1 Offside Position

It is not an offence in itself to be in an offside position.
A player is in an offside position if:

- He is nearer to his opponents' goal line than both the ball and the second last opponent.

A player is not in an offside position if:

- he is in his own half of the field of play,
- or he is level with the second last opponent,
- or he is level with the last two opponents.

FL 11.2 Offence

A player in an offside position is only penalized if, at the moment the ball touches or is played by one of his team, he is, in the opinion of the referee, involved in active play by:

- interfering with play,
- or interfering with an opponent,
- or gaining an advantage by being in that position.

FL 11.3 No Offence

There is no offside offence if a player receives the ball directly from:

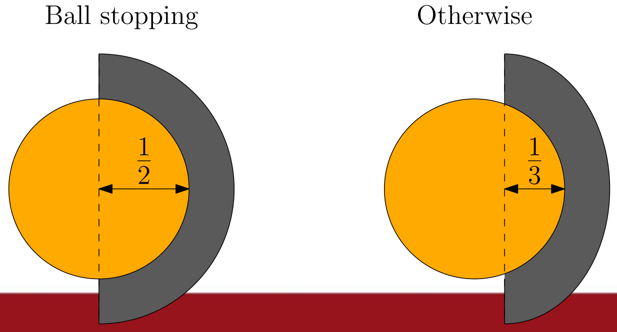
- a goal kick,
- or a throw-in,
- or a corner kick.

FL 11.4 Infringements/Sanctions

For any offside offence, the referee awards an indirect free kick to the opposing team to be taken from the place where the infringement occurred.

FIFA LAW 12 – Fouls and Misconduct

RoboCup Changes and Comments
Definition of Permissible Actions for Robot Soccer Players

	RoboCup Changes and Comments
RC-12.0.1: Ball Manipulation	• During a game the ball must not enter the convex hull of a robot by more than a third of its diameter except when the robot is stopping the ball. The ball must not enter the convex hull of a robot by more than half of its diameter if the robot is stopping the ball. This case only applies to instantaneous contact between robot and ball lasting no longer than one second. In any case it must be possible for another robot to take possession of the ball. • The robot may exert a force onto the ball only by direct physical contact between robot and ball. Forces exerted onto the ball that hinder the ball from rotating in its natural direction of rotation are allowed for no more than one second and a maximum distance of movement of fifty centimetres. Exerting this kind of forces repeatedly is allowed only either after a waiting time of at least four seconds or if the robot has previously completely released the ball. Natural direction of rotation means that the ball is rotating in the direction of its movement. • Ball rotation also implies that the ball is rotating continuously, even if slightly slower than its natural rotation speed. Movements of the ball such as ”roll-stop-roll-stop” are not considered a valid ball rotation and will be considered ball holding. • For any kind of ball dribbling, direct contact between the robot and the ball can only be maintained within a circle with a radius of three meters, centered on the point where the robot last caught the ball. To move past that circle, the robot has to completely release the ball so that this ball release can be directly observable by any of the referees. After that, the robot can capture the ball again and the center of the circle moves to the new catch position. It is up to the referees to determine if the ball has actually been completely released from the robot. Dribbling with direct contact between the robot and the ball outside this circle will be considered ball holding. It is up to the referee to decide if the robot dribbling the ball has complied with the above rule, namely in what concerns the three meters radius. The referee decision on this is final and non disputable. • Dribbling the ball backwards, that is, dribbling while the robot is moving towards the opposite direction of its relative position to the ball is allowed for a maximum distance of 2 meters. During the backward dribble the ball must also be rolling in its natural direction. Once any particular robot has dribbled the ball backwards for more than 1 meter, it can not repeat the same backward dribbling again before the ball has been completely released by that robot or until the robot has engaged a new ball struggle against an opponent robot (i.e. the ball is actively disputed between the two opponent robots for more than 2 seconds). • Violating any of the above rules is considered ball holding. 

Fouls and misconduct are penalized as follows:

FL 12.1 Direct Free Kick

A direct free kick is awarded to the opposing team if a player commits any of the following six offences in a manner considered by the referee to be careless, reckless or using excessive force:

- kicks or attempts to kick an opponent
- trips or attempts to trip an opponent
- jumps at an opponent
- charges an opponent
- strikes or attempts to strike an opponent
- pushes an opponent

A direct free kick is also awarded to the opposing team if a player commits any of the following four offences:

- tackles an opponent to gain possession of the ball, making contact with the opponent before touching the ball
- holds an opponent
- spits at an opponent
- handles the ball deliberately (except for the goalkeeper within his own penalty area)

A direct free kick is taken from where the offence occurred.

RoboCup	**RoboCup Changes and Comments**
RC-12.1: **Direct Free Kick**	Direct free kicks are currently awarded as indirect free kicks.

FL 12.2 Penalty Kick

A penalty kick is awarded if any of the above ten offences is committed by a player inside his own penalty area, irrespective of the position of the ball, provided it is in play.

RoboCup	**RoboCup Changes and Comments**
RC-12.2: **Penalty Kick**	In RoboCup, penalty kicks are currently not awarded during the two periods of play in a match. The competition rules may specify the execution of penalty kicks to decide the winner of a game which ends in a draw after two periods of play.

FL 12.3 Indirect Free Kick

An indirect free kick is awarded to the opposing team if a goalkeeper, inside his own penalty area, commits any of the following five offences:

- takes more than four steps while controlling the ball with his hands, before releasing it from his possession

- touches the ball again with his hands after it has been released from his possession and has not touched any other player
- touches the ball with his hands after it has been deliberately kicked to him by a team-mate
- touches the ball with his hands after he has received it directly from a throw-in taken by a team-mate
- wastes time

An indirect free kick is also awarded to the opposing team if a player, in the opinion of the referee:

- plays in a dangerous manner
- impedes the progress of an opponent
- prevents the goalkeeper from releasing the ball from his hands
- commits any other offence, not previously mentioned in Law 12, for which play is stopped to caution or dismiss a player

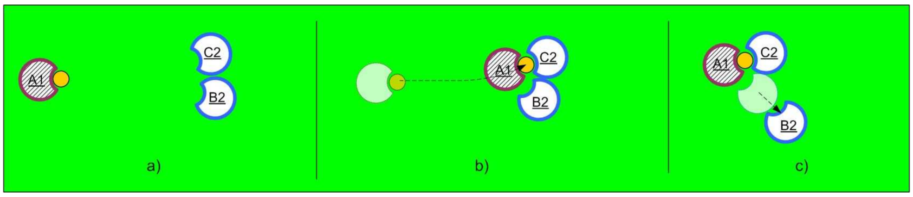
The indirect free kick is taken from where the offence occurred.

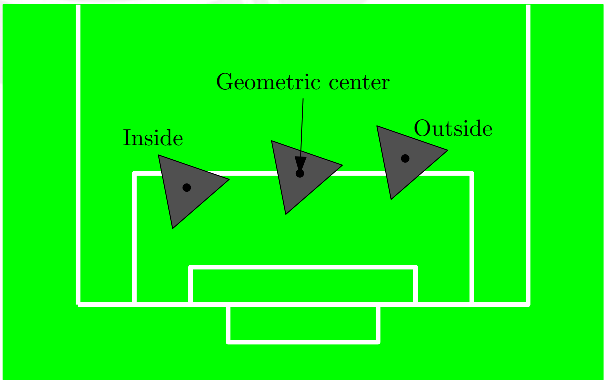
RoboCup	**RoboCup Changes and Comments**
RC-12.3: Indirect Free Kick	An indirect free kick is awarded to the opposing team, if a player, in the opinion of the referee, commits any of the following offences: • holding the ball • pushing an opponent • manual interference (see below) • kicking an opponent • performing illegal defense • performing illegal attack • delay of game **Manual interference:** An indirect free kick is awarded to the opposing team, if a human member of a team, in the opinion of the referee, commits any of the following offences: • entering the field during the game and breaks without permission of the referee • touching the robot during the game and breaks • interfering with the game on the field, e.g. through touching the ball while removing a robot • interfering with the game remotely through wireless communication, e.g. by remotely joysticking a robot, or send commands to robots from a machine not on the robots that convey information about the position of objects on the field or activate particular patterns of actions on the robots • behaving otherwise in an unsportsmanlike manner The referee may stop the game and give a yellow card to the player or the human member of the team that has committed one of those offences. The indirect free kick will be started from where the ball was when the offences occurred if the ball was not inside a penalty area, and from the closest restart point if the ball was inside of a penalty area. The offences are described and clarified subsequently.
RC-12.3.1: Ball Holding	If a player commits any violation of the clauses on stopping, dribbling, or kicking the ball, a **ball holding foul** will be called. Ball holding or hindering the ball from rolling in its natural direction is only allowed for at most one second and at most one meter of movement. This kind of action can only be repeated after a waiting time of, at least, four seconds.

RoboCup Changes and Comments

RC-12.3.2: Pushing

- Robots must play such that they try to avoid physical contact. However, physical contact *per se* does not represent an offence.
- All robots must be equipped to detect situations of physical contact with other robots (direct pushing situations). The obligation to detect pushing situations includes also indirect contact with another robot through the ball (i.e. the ball is between the player and an opponent).
- If physical contact with other robots cannot be avoided, it must be **soft**, i.e. at slow speed and with as small physical impact as possible, in order to avoid damage to itself and other robots. Robots moving at high speed must significantly decelerate before a collision with another robot.
- Whenever a robot produces direct or indirect physical contact with another robot while moving, it must stop movement immediately in that direction and choose a new direction for movement.
- If pushing occurs between a moving and a standing robot, the moving robot causes the pushing situation and is responsible for resolving it.
- If pushing occurs between two moving robots, both robots are responsible for resolving the pushing situation. If one robot continues pushing by moving in its initial direction, while the other robot is recognizably reacting and trying to take another direction, the foul will be called on the pushing robot.
- If an indirect pushing situation occurs between two robots, and neither of the robots can come out of it or shows significant attempts to get out of it within 10 seconds, a dropped ball will be called.
- If two robots encounter physical contact and cannot resolve the situation because they get entangled, the referee may issue a Dead Call (see RC-9.1.1) and order, afterwards, robot attendants to enter the field and slightly separate the entangled robots.
- If, in the opinion of the referee, physical contact between two robots is not soft, or if one or both of the robots do not change direction after encountering physical contact, a **pushing foul** will be called.
- While two robots from opponent teams are actively disputing the ball, that is, while the ball is being simultaneously touched by two robots of opponent teams, no other robot from either team can produce a direct contact with the two robots disputing the ball. If more than one robot is actively trying to intercept an opponent robot which is currently dribbling the ball (example bellow) then, as soon as two opponent robots are able to touch the ball, all other robots should move away, and can no longer be in contact with either the ball or any of the two robots disputing the ball. Violation of this rule will result in a pushing foul awarded to the offended team.



RoboCup	**RoboCup Changes and Comments**
RC-12.3.3: **Kicking**	The pushing clauses above apply in a similar manner to kicking motions: • Robots must avoid kicking each other. However, physical contact through an actuated kicking device does not *per se* represent an offence. • All robots must be equipped to decide whether they can execute a kick without threatening or damaging other robots. • If physical contact with other robots cannot be avoided, it must be **soft**, i.e. with small force. The stronger a kicking device is designed, the more fine-grained control of its activation is required. • If, in the opinion of the referee, physical contact through a kicking device is not soft, a **kicking foul** will be called.
RC-12.3.4: **Illegal Defense**	• Only the goal keeper may stay permanently in the own penalty area of a team. • A time limit of 10 seconds is allowed for a defending robot to leave its own penalty area. The referee may extend the time limit at his own discretion if the robot is actively making progress to leave the penalty area, or if it is prevented from leaving the penalty area by other robots. • If, in the opinion of the referee, a defending robot is not taking appropriate action to leave its own penalty area, or if a second defending robot is in the penalty area, an **illegal defense foul** will be called. • This rule overrides all other rules. • A robot is considered to be inside the penalty area if the projection of the robot's geometric center on the field overlaps or goes beyond the penalty area line. • The line of the penalty area is a part of the penalty area itself. 

	RoboCup Changes and Comments
RC-12.3.5: **Illegal Attack**	• A time limit of 10 seconds is allowed for an attacking robot to (possibly execute a kick motion and) leave the opponent's penalty area. The referee may extend the time limit at his own discretion, if the robot is actively making progress to leave the opponent's penalty area, or if it is prevented from leaving the penalty area by other robots. • If, in the opinion of the referee, the attacking robot is not taking appropriate action to leave the opponent's penalty area, or if a second attacking robot is in the opponent's penalty area, an **illegal attack foul** will be called. • This rule overrides all other rules. • The line of the penalty area is a part of the penalty area itself. • Touching the goalie in his goal area is an illegal attack foul; independently of the question whether the goalie moved to the attacker or vice versa. The goalie is considered to be inside the goal area as long as the projection of its geometric center on the field is inside or over the goal area limit.
RC-12.3.6: **Manual Interference**	• Robot attendants must, at all times, avoid any interference with the game process. • If, in the opinion of the referee, a human team member is interfering with the game process on the field, a **manual interference foul** will be called (e.g. touching the robots during game or breaks without the intention to take them out for repair or when a team member instead of the team leader is entering the field of play during the game or breaks). • A team member asking the spectators to move or hide because they wear clothes with colors used in RoboCup, interferes also with the game process. A referee can treat this also as a **manual interference foul**.
RC-12.3.7: **Remote Interference**	• No remote human interference of any kind with the game process is allowed. • In particular, the laptops used for coaching the robots (base station) must remain closed for the full duration of the game. • If, in the opinion of the referee, a human team member is remotely interfering with the game process, a **remote interference foul** will be called. • The only exception to the above rules is high level coaching of the robots (see FL 3.10, RC-Decision 2.1).
RC-12.3.8: **Delay Of Game**	• If a player removes (for the second time) the ball from its position during a game-stoppage, a **delay of game foul** can be called.

	RoboCup Changes and Comments
RC-12.3.9: Unsportsmanlike Behaviour	• Human team members must at all times during the match behave in an appropriate manner. • At least the following behaviours are considered unsportsmanlike: – not following instructions of the referee and the assistant referees – yelling at or insulting the referees, or the opponent, or the audience • If, in the opinion of the referee, a human team member is behaving in an inappropriate manner, an **unsportsmanlike behaviour foul** will be called.
RC-12.3.10: Goalie Protection	• Only the goalie is allowed to be in the goal area. • If an attacking robot enters the defender's goal area (i.e. when the projection of the robot's geometric center on the field overlaps or goes beyond the goal area line), then it will cause a foul. • If a defending robot other than the goalie enters the defender's goal area (i.e. when the projection of the robot's geometric center on the field overlaps or goes beyond the goal area line), then it will cause a foul. • The line of the goal area is a part of the goal area itself.
RC-12.3.11: Manual Positioning	• It is not allowed to re-position robots manually during a game break. If a robot needs to be re-positioned, it has to be taken out for repair. See also NOTE from RC-8.3.1.

FL 12.4 Disciplinary Sanctions

Only a player or substitute or substituted player may be shown the red or yellow card.

FL 12.5 Cautionable Offences

A player is cautioned and shown the yellow card if he commits any of the following seven offences:

- is guilty of unsporting behaviour
- shows dissent by word or action
- persistently infringes the Laws of the Game
- delays the restart of play
- fails to respect the required distance when play is restarted with a corner kick or free kick
- enters or re-enters the field of play without the referee's permission
- deliberately leaves the field of play without the referee's permission

RoboCup	**RoboCup Changes and Comments**
RC-12.5: Cautionable Offences	In RoboCup, a robot player or human team member is cautioned and shown the yellow card if he commits any of the following offences: • is guilty of unsporting behaviour • persistently infringes the Laws of the Game • delays the restart of play • fails to respect the required distance when play is restarted with a corner kick, goal kick, throw in or free kick • enters the field of play without the referee's permission. Exceptions are defined in RC-4.1 and RC-5.3.2 • crash at high speed with the field safety boundary, Goal, or any other robot
RC-12.5.1: Yellow Cards	The assistant referees will maintain a count of yellow cards given to each player (e.g. by the RefBox).

FL 12.6 Sending-Off Offences

A player is sent off and shown the red card if he commits any of the following seven offences:

- is guilty of serious foul play
- is guilty of violent conduct
- spits at an opponent or any other person
- denies the opposing team a goal or an obvious goal-scoring opportunity by deliberately handling the ball (this does not apply to a goalkeeper within his own penalty area)
- denies an obvious goal-scoring opportunity to an opponent moving towards the player's goal by an offence punishable by a free kick or a penalty kick
- uses offensive, insulting or abusive language
- receives a second caution in the same match

RoboCup	**RoboCup Changes and Comments**
RC-12.6: Temporary Sent-Off	A player is temporarily sent off the field after receiving its second yellow card. The player may return to the game on the next game interrupt, but no less than 2 minutes after being temporarily sent off.

RoboCup	RoboCup Changes and Comments
RC-12.6: Sending-Off Offences	A robot player or human team member is shown the red card and sent off for the remainder of the game if it commits any of the following offences: • is guilty of serious foul play, especially when exhibiting behaviour considered to be reckless and rude, or presenting a threat to other robots, human team members, the referees, and the audience • spits at an opponent or any other person • denies the opposing team a goal or an obvious goal-scoring opportunity by deliberately handling the ball (this applies only to human team members) • is guilty of violent conduct • uses offensive, insulting or abusive language • crashes at high speed with the field safety boundary or Goal causing substantial damages on it • crashes at high speed with any other robot causing visible and substantial damage on it • receives the third yellow card in the same match

Decisions of the International F.A. Board

Decision 1: A penalty kick is awarded if, while the ball is in play, the goalkeeper, inside his own penalty area, strikes or attempts to strike an opponent by throwing the ball at him.

Decision 2: A player who commits a cautionable or sending-off offence, either on or off the field of play, whether directed towards an opponent, a team-mate, the referee, an assistant referee or any other person, is disciplined according to the nature of the offence committed.

Decision 3: The goalkeeper is considered to be in control of the ball by touching it with any part of his hand or arms. Possession of the ball includes the goalkeeper deliberately parrying the ball, but does not include the circumstances where, in the opinion of the referee, the ball rebounds accidentally from the goalkeeper, for example after he has made a save. The goalkeeper is considered to be guilty of time-wasting if he holds the ball in his hands or arms for more than 5-6 seconds.

Decision 4: Subject to the terms of Law 12, a player may pass the ball to his own goalkeeper using his head or chest or knee, etc. If, however, in the opinion of the referee, a player uses a deliberate trick while the ball is in play in order to circumvent the Law, the player is guilty of unsporting behaviour. He is cautioned, shown the yellow card and an indirect free kick is awarded to the opposing team from the place where the infringement occurred.

A player using a deliberate trick to circumvent the Law while he is taking a free kick, is cautioned for unsporting behaviour and shown the yellow card. The free kick is retaken.

In such circumstances, it is irrelevant whether the goalkeeper subsequently touches the ball with his hands or not. The offence is committed by the player in attempting to circumvent both the letter and the spirit of Law 12.

Decision 5: A tackle from behind, which endangers the safety of an opponent, must be sanctioned as serious foul play.

RoboCup	**RoboCup Changes and Comments**
RC-Decision 1-4:	The Intl. F.A. Board Decisions 1 to 4 do currently not apply to RoboCup.

FIFA LAW 13 – Free Kicks

FL 13.1 Types of Free Kicks

Free kicks are either direct or indirect.
For both direct and indirect free kicks, the ball must be stationary when the kick is taken and the kicker does not touch the ball a second time until it has touched another player.

	RoboCup Changes and Comments
RC-13.1.1: Free kicks	In RoboCup, an indirect free kick is awarded in all situations where the Laws of the Game specify a direct free kick.
RC-13.1.2: Touching	In RoboCup, the kicker may touch the ball more than one time as long as the ball has not moved over a distance of more than 0.20m after an indirect free kick. After that, the ball must be touched by another player before the kicking robot can touch the ball again. A goal may be scored only after the ball has been touched by another player of the same team.

FL 13.2 The Direct Free Kick

- If a direct free kick is kicked directly into the opponents' goal, a goal is awarded.
- If a direct free kick is kicked directly into the team's own goal, a corner kick is awarded to the opposing team.

FL 13.3 The Indirect Free Kick

Signal The referee indicates an indirect free kick by raising his arm above his head. He maintains his arm in that position until the kick has been taken and the ball has touched another player or goes out of play.

Ball Enters the Goal A goal can be scored only if the ball subsequently touches another player before it enters the goal.

- If an indirect free kick is kicked directly into the opponents' goal, a goal kick is awarded.
- If an indirect free kick is kicked directly into the team's own goal, a corner kick is awarded to the opposing team.

FL 13.4 Position of Free Kick

Free Kick Inside the Penalty Area

Direct or indirect free kick to the defending team:

- all opponents are at least 9.15m (10yds) from the ball.
- all opponents remain outside the penalty area until the ball is in play.
- the ball is in play when it is kicked directly beyond the penalty area.
- a free kick awarded in the goal area is taken from any point inside that area.

Indirect free kick to the attacking team:

- All opponents are at least 9.15m (10yds) from the ball until it is in play, unless they are on their own goal line between the goalposts.
- the ball is in play when it is kicked and moves
- an indirect free kick awarded inside the goal area is taken from that part of the goal area line which runs parallel to the goal line, at the point nearest to where the infringement occurred.

Free Kick Outside the Penalty Area

- All opponents are at least 9.15m (10yds) from the ball until it is in play.
- The ball is in play when it is kicked and moves.
- The free kick is taken from the place where the infringement occurred.

	RoboCup Changes and Comments
RC-13.4: Position of the Free Kick	For RoboCup, replace "*9.15m*" with "**3m or anywhere (except goal area) within their own penalty area**". This means that one robot may be placed anywhere inside the own penalty area (except goal area), even if the distance to the ball is shorter than 3m.

	RoboCup Changes and Comments
RC-13.4.1: Procedure	In RoboCup, the following procedure is used for free-kick: • The referee gives a "stop" signal. • All players have to stop their movement. • The indirect free kick will be started from where the ball was when the offences occurred, if the ball was not inside a penalty area and from the closest restart point if the ball was inside of a penalty area. • The referee gives a "free-kick" signal. • The robot of the attacking team that is taking the kick is positioned at the ball. • All other players of the free-kick awarded team can stay anywhere on the field except in a circle with a radius of 2m around the ball until the ball is in play. • All players of the defending team can stay anywhere on the field except in a circle with a radius of 3m around the ball until the ball is in play. One robot may stay anywhere inside the penalty area (except goal area) of its own team, even if the distance to the ball is shorter than 3m. • The referee gives a "start" signal. • A player of the team who was awarded the free-kick kicks the ball. • The robot taking the kick should either use its kicker or one of its sides to instantaneously kick (i.e., without dribbling or dragging) the ball such that it travels freely over a distance of at least 0.5m. • The ball is in play immediately after being kicked. • After the kick, the attacking team is only allowed to touch the ball a second time after it moved over a distance of at least 0.5m. • A goal may be scored only when the ball was touched by another player of the same team. • When 10 seconds have passed after the signal and the ball wasn't kicked by the attacking team, the defending team can approach the ball and score a goal directly (if the ball is inside the opponent half field), even without any contact between the ball and any other player. However, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players. • If a robot of the attacking team except the kicking robot approaches the ball before the ball is in play, the free-kick will be awarded to the other team. It is forbidden to reposition robots by hand or by any other means with the only exception of the use of high level coaching of the robots (see FL 3.10, RC-Decision 2.1). The referee may show a yellow card to the robot that doesn't stay at least 2m (for the attacking team) or 3m (for the defending team) from the ball, following the referee's instructions more than twice consecutively. After that, if the robot doesn't follow the position restrictions of the procedure, the referee will instruct the team to remove the robot from the field. The above mentioned 2m and 3m refers to the radius of a circle centered on the ball. The robots must be completely out of each circle respectively, depending on its status (attacking or defending). The referee must restart the game within 10 seconds after game stops.

FL 13.5 Infringements/Sanctions

If, when a free kick is taken, an opponent is closer to the ball than the required distance:

- the kick is retaken.

If, when a free kick is taken by the defending team from inside its own penalty area, the ball is not kicked directly into play:

- the kick is retaken.

Free kick taken by a player other than the goalkeeper

If, after the ball is in play, the kicker touches the ball a second time (except with his hands) before it has touched another player:

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred .

If, after the ball is in play, the kicker deliberately handles the ball before it has touched another player:

- a direct free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.
- a penalty kick is awarded if the infringement occurred inside the kicker's penalty area.

Free kick taken by the goalkeeper

If, after the ball is in play, the goalkeeper touches the ball a second time (except with his hands), before it has touched another player:

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the goalkeeper deliberately handles the ball before it has touched another player:

- a direct free kick is awarded to the opposing team if the infringement occurred outside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.
- an indirect free kick is awarded to the opposing team if the infringement occurred inside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.

RoboCup	**RoboCup Changes and Comments**
RC-13.5: Infringements/ Sanctions	In RoboCup, all clauses referring to situations, where the player taking the free kick is touching the ball a second time, do currently not apply, except for the limitations defined in RC-13.1.2.

FIFA LAW 14 – The Penalty Kick

A penalty kick is awarded against a team which commits one of the ten offences for which a direct free kick is awarded, inside its own penalty area and while the ball is in play.
A goal may be scored directly from a penalty kick.
Additional time is allowed for a penalty kick to be taken at the end of each half or at the end of periods of extra time.

FL 14.1 Position of the Ball and the Players

The ball:

- is placed on the penalty mark.

The player taking the penalty kick:

- is properly identified.

The defending goalkeeper:

- remains on his goal line, facing the kicker, between the goalposts until the ball has been kicked.

The players other than the kicker are located:

- inside the field of play.
- outside the penalty area.
- behind the penalty mark.
- at least 9.15m (10yds) from the penalty mark.

RoboCup	**RoboCup Changes and Comments**
RC-14.1: Position of Ball and Players	In RoboCup, the following modification apply: • The defending goal keeper stays within his own goal area until the ball is even slightly moved. • The goalie can move at any time as long as it does no leave its goal area. • The kicker is located inside the center circle. • The players other than the kicker are located inside the field of play, outside the center circle, and behind the center line on the opposite side of the defending goal keeper.

FL 14.2 The Referee

- does not signal for a penalty kick to be taken until the players have taken up position in accordance with the Law.
- decides when a penalty kick has been completed.

RoboCup	**RoboCup Changes and Comments**
RC-14.2: The Referee	The ”Penalty Procedure” FIFA LAW48 specifies additional criteria for deciding when a penalty kick has been completed.

FL 14.3 Procedure

- The player taking the penalty kicks the ball forward.
- He does not play the ball a second time until it has touched another player.
- The ball is in play when it is kicked and moves forward.

When a penalty kick is taken during the normal course of play, or time has been extended at half-time or full time to allow a penalty kick to be taken or retaken, a goal is awarded if, before passing between the goalposts and under the crossbar:

- the ball touches either or both of the goalposts and/or the crossbar, and/or the goalkeeper.

RoboCup	**RoboCup Changes and Comments**
RC-14.3: Procedure	In RoboCup, the following procedure is followed for a penalty kick: • The penalty starts 5 min. after the end of the game. • Each team is awarded five penalty shots. • All players take their positions. The ball is placed on the penalty mark by the referee or one of the assistant referees. • The referee gives a ”penalty” signal followed by a ”start” signal. • The field robot that is taking the penalty must stay inside the center circle until the ”start” signal is issued. • The ball is in play when it was even slightly moved by the field robot. • After the field robot catches the ball, the ball may only move in a direction towards the goal. • The robot must kick the ball before the ball enters the penalty area, otherwise no goal is awarded. • The goalie is allowed to move at any time within the goal area only. • A goal is awarded if the ball passes the goal line between the goal posts and under the crossbar within 30 seconds after the ”start” signal of the referee. • No field player can touch the ball again and exert force on it after it has been touched by the goal keeper. • If the goalie leaves the goal area this shot will be repeated. If the same happens again the goal will be awarded. • It is only allowed to kick the ball once. • If the kicker touches the ball a second time, a possible goal will not be awarded. • All 5 penalties of one team are taken sequentially, followed directly by the 5 penalties of the other team. • If after the series, there is a tie, the penalty shoot-out will be repeated. If after 5 more penalties of each team there is still no winner, the game will be decided according to competition rule 3.5.

FL 14.4 Infringements/Sanctions

If the referee gives the signal for a penalty kick to be taken and, before the ball is in play, one of the following situations occurs:

The player taking the penalty kick infringes the Laws of the Game:

- The referee allows the kick to proceed.
- If the ball enters the goal, the kick is retaken.
- If the ball does not enter the goal, the kick is not retaken.

The goalkeeper infringes the Laws of the Game:

- The referee allows the kick to proceed.
- If the ball enters the goal, a goal is awarded.
- If the ball does not enter the goal, the kick is retaken.

A team-mate of the player taking the kick enters the penalty area or moves in front of or within 9.15 m (10 yds) of the penalty mark:

- The referee allows the kick to proceed.
- If the ball enters the goal, the kick is retaken.
- If the ball does not enter the goal, the kick is not retaken.
- If the ball rebounds from the goalkeeper, the crossbar or the goal post and is touched by this player, the referee stops play and restarts the match with an indirect free kick to the defending team.

A team-mate of the goalkeeper enters the penalty area or moves in front of or within 9.15 m (10 yds) of the penalty mark:

- The referee allows the kick to proceed.
- If the ball enters the goal, a goal is awarded.
- If the ball does not enter the goal, the kick is retaken.

A player of both the defending team and the attacking team infringe the Laws of the Game:

- The kick is retaken.

If, after the penalty kick has been taken:

The kicker touches the ball a second time (except with his hands) before it has touched another player:

- An indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

The kicker deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

The ball is touched by an outside agent as it moves forward:

- the kick is retaken.

The ball rebounds into the field of play from the goalkeeper, the crossbar or the goalposts, and is then touched by an outside agent:

- The referee stops play.
- Play is restarted with a dropped ball at the place where it touched the outside agent.

RoboCup	**RoboCup Changes and Comments**
RC-14.4: Infringements/ Sanctions	For RoboCup, replace "*enters the penalty area or moves in front of or within 9.15m (10yds) of the penalty mark*" with "**enters the side of the field where the penalty kick takes place**".

FIFA LAW 15 – The Throw-In

A throw-in is a method of restarting play.
A goal cannot be scored directly from a throw-in.
A throw-in is awarded

- when the whole of the ball passes over the touch line, either on the ground or in the air,
- from the point where it crossed the touch line,
- to the opponents of the player who last touched the ball.

FL 15.1 Procedure (The Throw-In)

At the moment of delivering the ball, the thrower

- faces the field of play,
- has part of each foot either on the touch line or on the ground outside the touch line,
- uses both hands,
- delivers the ball from behind and over his head.

The thrower may not touch the ball again until it has touched another player.
The ball is in play immediately after it enters the field of play.

	RoboCup Changes and Comments
RC-15.1: Procedure	In RoboCup, the following procedure is used for throw-in: • The referee gives a “stop” signal. • All players have to stop their movement. • The ball is placed on the touch line by the referee or one of the assistant referees. • The referee gives a “throw-in” signal. • The robot of the attacking team that is taking the kick is positioned at the ball. • All other players of the throw-in awarded team can stay anywhere on the field except in a circle with a radius of 2m around the ball until the ball is in play. • All players of the defending team can stay anywhere on the field except in a circle with a radius of 3m around the ball until the ball is in play. One robot may stay anywhere inside the penalty area (except goal area) of its own team, even if the distance to the ball is shorter than 3m. • The referee gives a “start” signal. • A player of the team who was awarded the throw-in kicks the ball. • The robot taking the kick should either use its kicker or one of its sides to instantaneously kick (i.e., without dribbling or dragging) the ball such that it travels freely over a distance of at least 0.5m. • The ball is in play immediately after being kicked. • After the kick, the attacking team is only allowed to touch the ball a second time after it moved over a distance of at least 0.5m. • A goal may be scored only when the ball was touched by another player of the same team. • When 10 seconds have passed after the signal and the ball wasn’t kicked by the attacking team, the defending team can approach the ball and score a goal directly (if the ball is inside the opponent half field), even without any contact between the ball and any other player. However, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players. • If a robot of the attacking team except the kicking robot approaches the ball before the ball is in play, a free-kick will be awarded to the other team. It is forbidden to reposition robots by hand or by any other means with the only exception of the use of high level coaching of the robots (see FL 3.10, RC-Decision 2.1). The referee may show a yellow card to the robot that doesn’t stay at least 2m (for the attacking team) or 3m (for the defending team) from the ball, following the referee’s instructions more than twice consecutively. After that, if the robot doesn’t follow the position restrictions of the procedure, the referee will instruct the team to remove the robot from the field.

	RoboCup Changes and Comments
RC-15.1: Procedure	The above mentioned 2m and 3m refers to the radius of a circle centered on the ball. The robots must be completely out of each circle respectively, depending on its status (attacking or defending). The referee must restart the game within 10 seconds after game stops.

FL 15.2 Infringements/Sanctions

Throw-in taken by a player other than the goalkeeper

If, after the ball is in play, the thrower touches the ball a second time (except with his hands) before it has touched another player

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the thrower deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.
- A penalty kick is awarded if the infringement occurred inside the thrower's penalty area.

Throw-in taken by the goalkeeper

If, after the ball is in play, the goalkeeper touches the ball a second time (except with his hands), before it has touched another player

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the goalkeeper deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team if the infringement occurred outside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.
- An indirect free kick is awarded to the opposing team if the infringement occurred inside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.

If an opponent unfairly distracts or impedes the thrower

- he is cautioned for unsporting behaviour and shown the yellow card.

For any other infringement of this Law

- the throw-in is taken by a player of the opposing team.

	RoboCup Changes and Comments
RC-15.2: Infringements/ Sanctions	In RoboCup, all clauses referring to situations, where the player taking the throw-in is touching the ball a second time, do currently not apply, except for the limitations defined in RC-13.1.2.

FIFA LAW 16 – The Goal Kick

A goal kick is a method of restarting play.
A goal may be scored directly from a goal kick, but only against the opposing team.
A goal kick is awarded when

- the whole of the ball, having last touched a player of the attacking team, passes over the goal line, either on the ground or in the air, and a goal is not scored in accordance with Law 10

FL 16.1 Procedure (for Goal Kick)

- The ball is kicked from any point within the goal area by a player of the defending team.
- Opponents remain outside the penalty area until the ball is in play.
- The kicker does not play the ball a second time until it has touched another player.
- The ball is in play when it is kicked directly beyond the penalty area.

	RoboCup Changes and Comments
RC-16.1: Procedure	In RoboCup, the following procedure is used for a goal kick: • The referee gives a “stop” signal. • All players have to stop their movement. • The ball is placed at the nearest restart marker to the position where the ball passed the goal line by the referee or one of the assistant referees. • The referee gives a “goal kick” signal. • The robot of the attacking team that is taking the kick is positioned at the ball. • All other players of the goal kick awarded team can stay anywhere on the field except in a circle with a radius of 2m around the ball until the ball is in play. • All players of the opponent team can stay anywhere on the field except in a circle with a radius of 3m around the ball until the ball is in play. • The referee gives a “start” signal. • A player of the team who was awarded the goal kick kicks the ball. • The robot taking the kick should either use its kicker or one of its sides to instantaneously kick (i.e., without dribbling or dragging) the ball such that it travels freely over a distance of at least 0.5m. • The ball is in play immediately after being kicked. • After the kick, the attacking team is only allowed to touch the ball a second time after it moved over a distance of at least 0.5m. • A goal may be scored only when the ball was touched by another player of the same team. • When 10 seconds have passed after the signal and the ball wasn’t kicked by the attacking team, the defending team can approach the ball and score a goal directly (if the ball is inside the opponent half field), even without any contact between the ball and any other player. However, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players. • If a robot of the attacking team except the kicking robot approaches the ball before the ball is in play, a free-kick will be awarded to the other team. It is forbidden to reposition robots by hand or by any other means with the only exception of the use of high level coaching of the robots (see FL 3.10, RC-Decision 2.1). The referee may show a yellow card to the robot that doesn’t stay at least 2m (for the attacking team) or 3m (for the defending team) from the ball, following the referee’s instructions more than twice consecutively. After that, if the robot doesn’t follow the position restrictions of the procedure, the referee will instruct the team to remove the robot from the field. The above mentioned 2m and 3m refers to the radius of a circle centered on the ball. The robots must be completely out of each circle respectively, depending on its status (attacking or defending). The referee must restart the game within 10 seconds after game stops.

FL 16.2 Infringements/Sanctions

If the ball is not kicked directly into play beyond the penalty area

- the kick is retaken.

Goal kick taken by a player other than the goalkeeper

If, after the ball is in play, the kicker touches the ball a second time (except with his hands) before it has touched another player

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the kicker deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.
- A penalty kick is awarded if the infringement occurred inside the kicker's penalty area

Goal kick taken by the goalkeeper

If, after the ball is in play, the goalkeeper touches the ball a second time (except with his hands) before it has touched another player

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the goalkeeper deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team if the infringement occurred outside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.
- An indirect free kick is awarded to the opposing team if the infringement occurred inside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.

For any other infringement of this Law:

- The kick is retaken

RoboCup	**RoboCup Changes and Comments**
RC-16.2: Infringements/ Sanctions	In RoboCup, all clauses referring to situations, where the player taking the goal kick is touching the ball a second time, do currently not apply, except for the limitations defined in RC-13.1.2.

FIFA LAW 17 - The Corner Kick

A corner kick is a method of restarting play.
A goal may be scored directly from a corner kick, but only against the opposing team.
A corner kick is awarded when

- the whole of the ball, having last touched a player of the defending team, passes over the goal line, either on the ground or in the air, and a goal is not scored in accordance with Law 10.

FL 17.1 Procedure (for Corner Kick)

- The ball is placed inside the corner arc at the nearest corner flag-post.
- The corner flag-post is not moved.
- Opponents remain at least 9.15m (10yds) from the ball until it is in play.
- The ball is kicked by a player of the attacking team.
- The ball is in play when it is kicked and moves.
- The kicker does not play the ball a second time until it has touched another player.

	RoboCup Changes and Comments
RC-17.1: Procedure	In RoboCup, the following procedure is used for a corner kick: - The referee gives a “stop” signal. - All players have to stop their movement. - The ball is placed inside the corner arc at the nearest corner to the position where the ball passed the goal line by the referee or one of the assistant referees. - The referee gives a “corner kick” signal. - The robot of the attacking team that is taking the kick is positioned at the ball. - All other players of the corner kick awarded team can stay anywhere on the field except in a circle with a radius of 2m around the ball until the ball is in play. - All players of the opponent team can stay anywhere on the field except in a circle with a radius of 3m around the ball until the ball is in play. One robot may stay anywhere inside the penalty area (except goal area) of its own team, even if the distance to the ball is shorter than 3m. - The referee gives a “start” signal. - A player of the team who was awarded the corner kick kicks the ball. - The robot taking the kick should either use its kicker or one of its sides to instantaneously kick (i.e., without dribbling or dragging) the ball such that it travels freely over a distance of at least 0.5m. - The ball is in play immediately after being kicked. - After the kick, the attacking team is only allowed to touch the ball a second time after it moved over a distance of at least 0.5m. - A goal may be scored only when the ball was touched by another player of the same team. - When 10 seconds have passed after the signal and the ball wasn’t kicked by the attacking team, the defending team can approach the ball and score a goal directly (if the ball is inside the opponent half field), even without any contact between the ball and any other player. However, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players. - If a robot of the attacking team except the kicking robot approaches the ball before the ball is in play, a free-kick will be awarded to the other team. It is forbidden to reposition robots by hand or by any other means with the only exception of the use of high level coaching of the robots (see FL 3.10, RC-Decision 2.1). The referee may show a yellow card to the robot that doesn’t stay at least 2m (for the attacking team) or 3m (for the defending team) from the ball, following the referee’s instructions more than twice consecutively. After that, if the robot doesn’t follow the position restrictions of the procedure, the referee will instruct the team to remove the robot from the field.

RoboCup	**RoboCup Changes and Comments**
RC-17.1: Procedure	The above mentioned 2m and 3m refers to the radius of a circle centered on the ball. The robots must be completely out of each circle respectively, depending on its status (attacking or defending). The referee must restart the game within 10 seconds after game stops.

FL 17.2 Infringements/Sanctions

Corner kick taken by a player other than the goalkeeper

If, after the ball is in play, the kicker touches the ball a second time (except with his hands) before it has touched another player

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the kicker deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.
- A penalty kick is awarded if the infringement occurred inside the kicker's penalty area.

Corner kick taken by the goalkeeper

If, after the ball is in play, the goalkeeper touches the ball a second time (except with his hands) before it has touched another player

- an indirect free kick is awarded to the opposing team, the kick to be taken from the place where the infringement occurred.

If, after the ball is in play, the goalkeeper deliberately handles the ball before it has touched another player:

- A direct free kick is awarded to the opposing team if the infringement occurred outside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.
- An indirect free kick is awarded to the opposing team if the infringement occurred inside the goalkeeper's penalty area, the kick to be taken from the place where the infringement occurred.

For any other infringement:

- The kick is retaken

RoboCup	**RoboCup Changes and Comments**
RC-17.2: Infringements/ Sanctions	In RoboCup, all clauses referring to situations, where the player taking the corner kick is touching the ball a second time, do currently not apply, except for the limitations defined in RC-13.1.2.

Chapter

Competition Rules

Competition Rules include:

- Competition Rule 1 - Team Qualification
- Competition Rule 2 - Referees
- Competition Rule 3 - Tournament Regulations
- Competition Rule 4 - Summary of Object Colouring
- Competition Rule 5 - Referee Box

COMPETITION RULE 1 - Team Qualification

CR 1.0 Admissibility of Team Qualification Procedures

The organizing committee of a tournament may limit the number of teams, that are allowed to participate in the tournament, for any of the following reasons:

- Scientific reasons, for example, when allowing more teams is likely to hurt scientific exchange and discussion or the overall scientific standard of the tournament.
- Space limitations are imposed by the site of the tournament.
- Time limitations are imposed by the overall tournament schedule.
- Any other kind of organizational constraint limits the number of teams that can be accommodated for.

In order to limit the number of participating teams, the organizing committee of a tournament may request teams to successfully complete a qualification procedure. The team qualification procedure shall be such that scientific progress and exchange is fostered.

CR 1.1 Team Qualification for RoboCup-2014

For RoboCup-2014, the team qualification procedure requires teams to submit the following material:

- A list of 5 scientific papers published during the last 5 years of the team which are related to RoboCup. Abstracts of these papers must also be submitted.
- Team Description Paper/Innovations of the team.
- A list of results and awards obtained by the team in the last 3 years.
- A video showing the capabilities of the team's robots.
- A list of contributions of the team to the RoboCup MSL community.
- Declaration if the team will be part of a mixed team.
- Each team must provide a mechanical and electrical description of their robots (e.g. by providing drawings) as well as a flow chart of the software structure.

The material have to be submitted to the TC no later than that date defined by the OC in the "Call for Participation". Material which arrived after this date will be not considered for the qualification process.

CR 1.2 Evaluation of the Qualification Material

The submitted qualification material will be evaluated by the MSL EXEC and TC and partially by the Team Leaders involved in the qualification process. For each part of the submitted qualification material a number of points are awarded. These points are summed up per team. The teams are ranked as result of the collected points starting with the highest score. For the ranking of mixed teams the result of the best evaluated sub-team is used. Teams also may send their qualification material only once as one mixed team.

Scientific results

Because RoboCup is primarily a scientific (not an entertainment) event, teams are strongly encouraged to submit technical papers to journals, major conferences and workshops. In particular, if a RoboCup tournament is associated or held in conjunction with a conference or a workshop, teams are strongly encouraged to submit papers to the RoboCup-2014 symposium.

In order to decide if the 5 listed publications of the team are relevant for the MSL or RoboCup in general, the papers are reviewed by the team leaders of the teams which have submitted qualification material. The TC assigns a number of papers (abstracts only) to the team leaders. The team leader has to give a feedback if he considers the papers as relevant for the MSL or RoboCup in general. Every paper has to be rated by at least two other team leaders. The participation of the team leader in the review process is mandatory for a successful qualification of the team and has to be done until a date defined by the TC. Papers have to be written in English in order to be evaluated.

Per accepted paper the following points are awarded:

- 6 points for a publication in an international journal or a book chapter
- 4 points for a publication at an international conference (peer-reviewed)
- 2 points for a publication at a national conference (peer-reviewed)
- 1 point for other publications (e.g., not peer-reviewed or PhD-thesis)

A maximum number of 30 points are awarded.

Performance in Past Events

A maximum of 20 points are awarded to a team for the performance in the last 3 years. If a team proceeds to the last 8 teams in a RoboCup world championship in the last 3 years 20 points are awarded to the team. If a team is ranked among the best 3 in the Technical Challenge in a RoboCup world championship in the last 3 years 15 points are awarded to the team. If a team proceeds to the last 8 teams in a regional RoboCup championship (e.g., German Open, Dutch Open, Japan Open, US Open, Iran Open, China Open, Portuguese Open) in the last 3 years 10 points are awarded to the team.

Team Description Paper/Innovations

Teams have to describe their most innovative contributions or scientific results in a paper with up to 8 pages in the Springer LNCS style. The content of the paper may comprise all topics related to RoboCup MSL or RoboCup in general (e.g. AI Planning, Vision, reinforcement learning, adaptive neural control, development of specialized hardware like sensors or processors for RoboCup, construction of innovative mechanical bases, self-localization, robot cooperation, team coordination, etc.). The submission of a team description paper/innovations is mandatory for the qualification process. The paper will be reviewed by the members of the TC. Each member can award up to 20 points. Finally, the average over all evaluation will be awarded to the team.

Qualification Video

Teams have to submit a 60 seconds long qualification video. If the submitted video is longer than 60 seconds, then only the first 60 seconds will be considered for evaluation. The video should show that the robots of the team are able to perform at least the basic actions necessary for the RoboCup MSL. The requested actions are: dribbling the ball, avoiding obstacles, kick towards the goal, self repositioning for a kick off, making a pass and a defending action of the goal keeper. For each of these actions which are shown on the video 3 points are awarded. For exceptional abilities, apart from those described above, each member of the TC can award up to 8 points. The average over all these evaluation plus the points of the necessary actions will be awarded to the team.

Contribution to the RoboCup MSL community

Contributions or service of the teams to the RoboCup MSL community are very important for the success of our league. Therefore, teams which actively serve for the community has to be honoured. Each member of the TC can award up to 10 points for the submitted list of contributions to the community. As contributions or service to the community count serving in league committees (EXEC, TC, OC), providing code for general use, e.g., the referee box, maintaining the league's homepage. The average over all evaluation will be awarded to the team.

Mechanical and electrical description of the robot and software flow chart

A reasonable mechanical and electrical description of the robot and a flow chart of the software architecture must be provided. If the material is provided and is not apparently wrong or changed on purpose 10 points will be awarded to the team.

CR 1.3 Due Date, Submission, and Review of Team Qualification Material

All team qualification material must be submitted to the chair of the organizing committee for the Middle Size Robot League no latter than the date previously announced by the MSL OC.

The technical committee will review the material submitted as part of the qualification procedure and select teams for the tournament.

CR 1.4 Agreement on Open Source Development

For the benefit of scientific exchange, teams should make available technology and software developed for RoboCup as much as possible after a tournament has been played.

COMPETITION RULE 2 - Referees

CR 2.1 Selection of Referees

Every team participating in a tournament must name at least two team members who serve as referees for matches. The named persons must have good knowledge of the rules as applied in the tournament and have to be able to lead a game in English. The persons should be selected among the more senior members of a team, and preferably have prior experience with games in the RoboCup MSL.

CR 2.2 Referee Assignment

The assignment of referees and assistants to matches is the task of the league organizing committee. A first selection of possible referees from all participating teams should be done during the registering phase of a tournament. Usually the OC ask all teams to send lists of referee volunteers.

One referee and at least one assistant will be assigned for judgement of a match. The league organizing committee may choose to assign more assistants. The recommended number is one main referee, one assistant referee and two goal assistants.

Assistants can be assigned specific tasks, like handling the stop watch, ensuring the absence of manual interference by team members, and such. It is recommended that the assistant referee takes care of timing, taking notes on cards shown, and filling out the referee game sheet. The duties of the goal assistants is to check the occurrence of goals and survey the timing rules regarding robots in the goal area.

If either a referee or an assistant assigned to a match cannot fulfil his duty for some reason, he has to inform the organizing committee as soon as possible, give the reason for his inability to fulfil his duty, and request a replacement to be named.

CR 2.3 Referees during Match

The referee and assistant referees should wear black clothing/shoes and avoid reserved colors for the field, and player markings in their clothing.

The referee and his assistants will be close to but off the field during play. The referee should take a position at some distance to team areas. The referee may order team members to maintain positions at an appropriate distance.

The referee and the assistants may enter the field in particular situations, e.g. to re-position the ball when the game gets stuck.

The referee and his assistants should avoid to interfere with robots as much as possible, unless a robot is threatening to cause serious damage to people, other robots, or other equipment.

The referee may order team members onto the field in order to remove a robot. Orders by the referee have to be executed promptly.

The referee may allow members of a team to enter the field, in particular during game breaks. No team members are allowed to enter the field or to interfere otherwise with the game process unless permitted or ordered by the referee. Exceptions to this Competition Rule are established in RC-5.3.2.

CR 2.4 Infringements/Sanctions

A team failing to meet its refereeing duties, either by not naming appropriate persons to the organizing committee or by the assigned referee not fulfilling his duties, is subject to penalties decided upon by the organizing committee of a tournament.

Penalties may include fines, to be paid immediately before the team's next match, or exclusion from the ongoing or future tournaments.

COMPETITION RULE 3 - Tournament Regulations

CR 3.0 Preliminary Remarks

In order to provide a good opportunity to gain match experience, the tournament plan shall be designed such that all teams can play as many games as possible.

CR 3.1 Parts of the Competition

The RoboCup-2014 competition consist of the following parts:

- Team Registration, Setup, and Technical Inspection
- Technical Evaluation Rounds to assign Scientific and Engineering Awards and define groups for tournament
- Preliminary Rounds (Round-Robin)
- Playoffs

Every team that is admitted to the tournament must participate in all parts of the competition. Teams that, for any reason, may no longer be actively participating in the games, will still have to serve as referees and for various other duties. Teams failing to serve the duties they have been assigned to may be excluded from future tournaments.

CR 3.2 Team Registration, Setup, and Technical Inspection

Every team has to set up and register on site at least 24 hours before the first game of the tournament is scheduled.

The Middle Size League Committee **strongly recommends** arriving no later than 48 hours before scheduled games start.

It is the sole responsibility of the teams to plan transport of equipment and travel of team members such that both arrive on time. Teams should carefully take into account any potential visa and/or customs problems that may arise.

Teams which did not personally register on site at least 24 hours prior to the first scheduled game may be excluded from the tournament.

Teams excluded from the tournament because of showing up late are not eligible to a refund of registration fees or any other kind of expenses. Neither the RoboCup Federation, nor the local organizers, nor anyone else involved in organizing a RoboCup tournament can be held liable for any cost, or damage suffered, by teams excluded from the tournament.

Team shall set up their robots and equipment and make any adjustments to local conditions well before the first game starts.

During setup, teams should use the field only when necessary, and only for shortest possible times. The League Organizing Committee may impose restrictions on the use of fields during setup and install special procedures for obtaining access to the field.

For the 2014 edition the Local Organizing Committee will provide, within the MSL area, but outside the fields, a 9m x 6m artificial grass field with adequate white lines (available throughout the competition). This field will be used for the Technical Challenge and will be made available for teams that want to use it for experiments. The length of the grass leaves may be between 8 and 10 mm.

The Middle Size League Technical Committee will organise a technical inspection of robots during the setup phase. Currently, the following procedure will be followed:

1. Size and shape constraints will not be checked during technical inspection, unless some team leader explicitly requests a check.
2. All robots will be photographed or filmed during technical inspection.
3. Once technical inspection is over, size constraints are checked only if someone assumes some physical change to a robot. Objections may be raised only up to 10 minutes before the game starts and only by a team leader.
4. Compliance with the connection to the referee box and field AP will be tested. Also maximum emitted power by robots WiFi equipment will be tested according to RC-4.2.5.
5. The opponent team must permit inspection of robots for 10 minutes, from 20 minutes to 10 minutes before game start.
6. No objections will be taken later on.

Note: Details of technical inspection are subject to changes by the League Technical Committee at any time, depending on the situation or requests by a team leader meeting on site.

CR 3.3 Technical Challenge

In order to promote the scientific goals of RoboCup and an according team attitude, the Middle Size League Committee will give recognition to specific scientific and engineering achievements tested by challenge competitions, as described in F-2000 Challenges.

Number and character of awards will be determined in accordance with the Executive Committee of the RoboCup Federation.

Mandatory and free challenges are identified in F-2000 Challenges.

CR 3.4 Preliminary Rounds

For the preliminary rounds, teams will be assigned to groups.

The number of groups will be determined by the League Organizing Committee, which takes into account the number of qualified teams as well as site and schedule constraints of the tournament.

The ranking for the initial organization of the groups that will participate in the competition will be done according to the following rules:

- For teams that have participated in the 2013 RoboCup edition, the final classification will be used;
- Teams that did not participate in the 2013 RoboCup edition will be ranked, after the previous ones, according to pre-qualification results.
- If a tie persists among one or more teams a draw will be performed. Team leaders must be present during the draw procedure.

Each group will play a single round of round-robin matches, i.e. each team will play once against every other team in its group.

As a guideline, every team should be prepared to play as many as eight (8) games within two days.

During the preliminary rounds, a match ending in a draw will NOT be decided by a penalty shoot-out.

The winner of a match will be awarded three (3) points, the loser will be awarded no points. If a match ends in a draw, each team will be awarded one point.

All points awarded to a team are added up.

Teams are ranked within their group by the points they gained during the preliminary round. If two or more teams have the same number of points, a decision is based upon the following criteria, in the order given:

1. Goal difference.
2. Absolute number of goals achieved.
3. Result of direct match-up.
4. Result of technical challenge between tied teams.

Note: Depending on competition factors such as number of teams, number of fields and availability of the venue during the night, the organizing Committee may decide on how to organize the next preliminary rounds after the first round robin.

The best eight teams of *all* groups qualify for the play-offs.

We assume the number of groups m to be less than eight. Then the first $8/m$ (rounded down to the next lower integer) teams in each group are qualified for the play-offs.

If eight is not a multiple of the number of groups m, then wild-card games may be played to determine the teams for remaining spots in the play-offs. Details will be announced prior to the tournament.

This schedule construction may be subject to changes of the League Organizing Committee due to the number of participating teams, the site and schedule constraints of the tournament and the principle of maximizing the number of games for each team. If the schedule is subject to changes this will be announced prior to the beginning of the tournament.

CR 3.5 Play-offs

The play-offs consist of quarter finals, half finals, and finals matches. Every play off round is decided by a single match (best-of-one). If, by decision of the League Organizing Committee, only four teams dispute the play-offs, then the play-offs consist only of half finals, and finals matches

If play-off matches end in a draw after the regular match time, an extra 10 minutes extra time will be played. This extra time is divided into two 5 minutes halves with no interval time. If the extra time still ends in a draw, the game must be decided by playing a penalty shoot-out. If play-off matches are still drawn after a penalty shoot-out, the penalty shoot-out will be repeated and after another draw the match will be decided according to the following set of priority rules:

- the team with the best goal difference in the tournament so far, wins;
- the team that made most goals in the tournament so far wins;
- the team that performed best during the technical challenge wins;
- the team that performed best at the scientific challenge wins.

A plan of play-off matches will be made available by the organizing committee before start of the tournament.

CR 3.6 Appearance at game start

If a team does not show up at the beginning of a game, a victory will be awarded to the opponent team with a score of 3-0.

CR 3.7 Withdrawal from game

If a team withdraws from a game after it started, a victory will be awarded to the opponent team, either by adding three goals to its current score, or by adding the necessary number of goals to ensure a minimum difference of three goals.

COMPETITION RULE 4 - Summary of Object Colours

Here is the list of colors for objects on the field (surface, field boundary, goals and robots):

Object	**Colour**
Field surface	GREEN
Field safety boundary	BLACK
Lines on the field	WHITE
Goals	WHITE
Robot bodies	BLACK
Markers of robots for team A	LIGHT BLUE
Markers of robots for team B	MAGENTA/PURPLE

COMPETITION RULE 5 - Referee Box

The official referee box is available at

- http://sourceforge.net/projects/msl-refbox

It is mandatory for all teams to use this referee box.

COMPETITION RULE 6 - Normalized data structure

Beginning in 2014 an effort will be made to standardize several aspects of the teams world model and data representation in order to make it easier in the future to play with ad-hoc mixed teams. With this in mind the referee box will be updated with a new command. This command will be issued ten times per second and requests each team's base station to send a data structure filled with the most relevant data regarding the world state perception of each robot and of the team base station itself. These data will be recorded during the games and made available to the teams after the end of the game.

The exact format of the data structure will be released to the competing teams right after the pre-selection procedure.

COMPETITION RULE 7 - Rules Updating Time Out

If the TC doesn't provide updated rules for the next tournament until Dec, 31st, the existing rules of the last tournament will apply.

Chapter

F-2000 Challenges

For RoboCup 2014 two challenges have been selected as mandatory: Challenge 1 - Technical Challenge and Challenge 2 - Scientific/Engineering Challenge. All teams participating in the tournament have to take part in these challenges. A team that does not participate in any of the challenges can be disqualified from the tournament. All results in the tournament will be cancelled.
Teams not participating in the tournament are welcome to participate in the challenge competitions.
The winner of each of the challenge competitions will be awarded a prize for his achievement.
A list of proposals for other challenges to be used in the future is also presented.

Challenge 1 - Technical Challenge

PLAYING IN AN ARTIFICIAL GRASS FIELD

The aim of this challenge is to encourage teams to improve their ability to play in a less even floor and tackle with surfaces with different dynamic behaviour. This challenge is carried out by one (or optionally two) active robots and one passive robot, uses the official tournament ball and is disputed in three runs.

The challenge will be played in a 12m x 9m MSL field made of artificial grass. The length of the grass leaves may be between 8mm and 10mm. The lines will be white and will be downsized from the original measures to fit the field size. The goals will be the official ones.

PRESET:

- he team's goal keeper is placed in the middle of the predefined opponent defending goal. It must be disconnected or static (this is the passive robot).
- The active robot is placed in the middle of its own penalty area.
- Three black obstacles (one on the own side and two on the opponent side), similar in size to a MSL robot, will be placed at random positions on the field.
- The ball is placed at a random position on the own side of the field.

PROCEDURE:

- After receiving a start command from the Referee Box, the active robot has to find the ball and start dribbling it.
- After getting the ball, the robot has to dribble it, avoiding the obstacles, enter the opponent field and then shoot it into the predetermined goal where the goalkeeper is standing.
- For the goal to be valid the shoot must be taken from a distance to the target goal less than 2m. For this purpose, the ball must be completely within the 2 meter distance limit.
- The robot has 90 seconds to complete each run of the challenge.

SCORING:

- One point is awarded if the active robot correctly identifies the ball, (i.e. the robot touches the ball for the first time and is able to dribble it afterwards for at least 1 meter).
- A second point is awarded if this robot is able to dribble the ball to the opponent side of the field.

- A third point is awarded if the robot, handling the ball, successfully enters a zone of the field where the ball is closer than 2m to the opponent goal.
- A fourth point is awarded if the robot successfully scores a goal in the predetermined goal.

EXTRA POINTS:

Any team my choose to use two robots instead of one robot to perform the challenge. The second robot can be placed anywhere in the opponent side of the field. In this case:

- One extra point is awarded if the first robot, after crossing the mid line dribbling the ball makes a valid pass to the second robot (i.e. the ball rolls freely for at least 2 meters and is touched by the second robot). In this case, the second robot assumes the place of the first one for the remainder of the challenge.
- Another two extra points are awarded if the pass is done with a lob shot over one of the obstacles in the field and if the receiving robot is able to control the ball before it goes out of the field.

PENALTIES:

- Every contact of any of the active robots with an obstacle will be punished with a point, which is subtracted from the amount of points in the current run. A continuous contact with an obstacle (even if it moves the obstacle) will count as a single contact.
- If the ball goes out the field delimiting lines at any time, the attempt is terminated and a point will be subtracted from the amount of points in the current run. This does not apply if the ball goes out of the field after bouncing off from the goalkeeper or from the goal posts.
- If a shot at the goal is taken from a distance greater than 2m, a point will be subtracted from the amount of points in the current run.

The minimum number of points in one run will be zero. In total this challenge is repeated three times with different ball start positions, but always with the same robots, which means that a team can be awarded with up to a maximum of twenty one points for this challenge. If teams end up with the same amount of points, the relative placement is decided by the sum of the time spent in each of the three trials (smaller time beats higher time). The total time available for a team to execute its challenge is ten minutes.

Challenge 2 - Scientific/Engineering Challenge

In this challenge teams are free to show one significant achievement each, and all the other team leaders, together with the TC members will evaluate them. Achievements in the list of proposal challenges that follows are encouraged but are not limited to them. The judgement will take into consideration the following specific issues, each one of which will be classified between 1 and 10 points:

- Presentation
- Novelty
- Interest for either the present or the future of the league
- Scientific/Technical complexity
- Scientific relevancy for the league
- Importance of demonstrated experimental results
- Relevance of the published results presented as a support for this challenge

The final classification is obtained as follows:

- The average of the classifications given by each of the team leaders will be calculated (TLav).
- A standard average will then be determined by the average of those averages:

$$CL_{av} = \frac{\sum_1^n TL_{av}}{n}$$

- Each of the team leaders ratings will then be multiplied by the ratio

$$C_r = \frac{CL_{av}}{TL_{av}}$$

- The resulting ratings will then be added up for obtaining the final score.

Other Challenges that can be used in Future Tournaments

Beyond the previously described challenges the following list is currently regarded as proposals. All teams are invited to contribute their ideas in order to add new challenges or further improve existing challenges.

Challenge 3 - Cooperative playing with three robots

The aim of this challenge is to encourage teams to improve their cooperative behaviour as well as their algorithms for planning, and obstacle recognition and avoidance. This challenge is carried out by three active robots and one passive robot, uses the official tournament ball and is disputed in three runs. PRESET:

- The goal keeper is placed in the middle of the predefined opponent defending goal. It must be disconnected or static (this is the passive robot).
- The first robot (named robot A) is placed in the middle of the penalty area line in the predefined own half of the field.
- The other two active robots (named robot B and robot C respectively) are positioned at the two restart points in front of the penalty area of the predefined opponent side of the field.
- Furthermore, black obstacles (at least one on the own side and two on the opponent side), similar in size to an MSL robot, will be placed on random positions of the field.
- The ball is placed at a random position on the opponent side of the field.

PROCEDURE:

- After receiving a start command from the Referee Box, one of the robots standing in the opponent side of the field (either B or C) has to find and dribble the ball and pass it to robot A.
- Dribbling done by the robot that first touch the ball (either B or C) must occur for no less than 3 meters.
- Neither of the three robots is allowed to cross the middle line.
- When the ball is passed to robot A, it must roll freely for at least 2 meters before it is intercepted by the latter.
- Robot A has to intercept the ball, dribble it around the obstacles for no less than 3 meters and then pass it back to the remaining robot on the opponent side of the field.
- For this pass to be considered valid, the robot that receives the ball must not be the one that found and dribbled the ball in the first place (i.e if robot B dribbled the ball and made the pass to robot A, then it is mandatory that robot A make his pass to the robot C and vice-versa).
- Again, when the ball is passed to the B/C robot, it must roll freely for at least 2 meters before it is intercepted by this robot.
- The third active robot has then to intercept the ball, dribble it around the obstacles for no less than 2 meters, and then shoot it into the predetermined goal where the goalkeeper is standing.
- All robots are allowed to move as soon as the challenge is started. The robot team has 90 seconds to complete each run of the challenge.

SCORING:

- One point is awarded if either robot B or C has correctly identified the ball, (i.e. the robot has touched the ball for the first time and is able to dribble the ball afterwards for at least 3 meters).
- A second point is awarded if this robot correctly executes a pass, (i.e. the ball crosses the mid line and roll freely for at least 2 meters).
- A third point is awarded if the robot A successfully intercepts the pass (i.e. robot A has touched the ball for the first time and is able to dribble it afterwards for no less than 3 meters).
- A fourth point is awarded if the robot A correctly executes a pass, (i.e. the ball crosses the mid line and roll freely for at least two meters).
- A fifth point is awarded if the third robot (B, if C has done the first dribble, or C otherwise) successfully intercepts the pass (i.e. the third robot has touched the ball for the first time and is able to dribble it afterwards for no less than 2 meters).
- A sixth point is awarded if the third robot has successfully scored a goal in the predetermined goal.
- For each of the two passes that have to be performed, two extra bonus points are awarded if the pass is done with a lob shot over one of the obstacles in the field and if the receiving robot is able to control the ball before it goes out of the field.

PENALTIES:

- Every contact of any of the three active robots with an obstacle will be punished with a point, which is subtracted from the amount of points in the current run. A continuous contact with an obstacle (even of it moves the obstacle) will count as a single contact.
- If the ball goes out the field delimiting lines at any time, the attempt is terminated and a point will be subtracted from the amount of points in the current run.
- If the sequence of passes is not done in the correct order, the attempt is terminated with the current amount of points.

- If any of the three robots crosses the middle line, the attempt is terminated with the current amount of points.

The minimum number of points in one run will be zero. In total this challenge is repeated three times with different ball start positions, but always with the same robots, which means that a team can be awarded up to a maximum of eighteen points for this challenge. If teams have the same amount of points, the total time needed for all runs decides on the placement.

Final note: If a team does not have three operational robots at the beginning of the challenge, the challenge can still be be completed with a minimum of two active robots (robots B and C are then replaced by a single robot B). In this case the second pass and the goal will not be considered, which means that the team can score a maximum of four points per run.

Challenge 4 - Ball Control and Planning

Five to eight black obstacles (length/width 40 cm, height 60 cm) are put at arbitrary positions on the field. The ball is put on the middle of the penalty area line, and a robot inside the same goal. The robot should dribble the ball into the opposite goal within 90 seconds, while it avoids all obstacles. One point is awarded to the robot if the ball has passed the center line, another point when a goal is scored. Penalty points are given each time the robot or the ball touches an obstacle. The challenge is repeated three times with various setups. An extra point is awarded to the team with the fastest robot. In order to be eligible for this extra point the robot may not have touched any of the obstacles. In total a team can be awarded up to seven points for this challenge.

Challenge 5 - Cooperative Mixed-Team Play

Teams should demonstrate cooperative mixed-team play between at least two robots from different teams. The selection of the activity to be performed is free, but it should last at most 90 seconds. A jury will evaluate the quality of cooperation and cooperative behaviour and will assign up to six points to each team.

Challenge 6 - Team play with an arbitrary FIFA ball

The aim of this challenge is to encourage teams to improve their vision routines as well as their algorithms for cooperation, arbitrary ball detection and obstacle recognition and avoidance. This challenge is carried out by two robots, three times, with three different standard previously unknown FIFA balls.

The first robot is placed in the middle of the goal area line in the predefined own half. A second robot is placed on a random position in the other half (opponent half) of the field at least 2m away from the middle line.

Furthermore, black obstacles, similar in size to an MSL robot, can be placed on random positions on the whole field.

- At the team leader request, the ball may be placed, for no longer than 10 seconds, in front of the first robot, and at a distance of no less than 50cm from it. After that, the ball is replaced in a random position within the predefined own half of the field.
- The first robot has to find and dribble the ball and pass it to the second robot in the other half.
- Neither of the robots is allowed to cross the middle line.
- When the ball is passed by the first robot it must roll freely for at least 2m before it is intercepted by the second robot.
- The second robot has to intercept the ball, dribble it around the obstacles and shoot it into the predefined goal.
- Both robots are allowed to move as soon as the challenge is started. The robot team has 90 seconds to complete the challenge.

- One point is awarded if the first robot has correctly identified the ball, i.e. the robot has touched the ball for the first time and is able to dribble the ball afterwards.
- A second point is awarded if the first robot correctly executes a pass. The valid pass will only be considered if the ball crosses the mid line of the field without previously going out of the field.
- A third point is awarded if the second robot successfully intercepts the pass (i.e. the second robot has touched the ball for the first time and is able to dribble it afterwards).
- A fourth point is awarded if the second robot has successfully scored a goal in the predefined goal.

Every contact of one of the robots with an obstacle will be restricted with a negative point, which is subtracted from the amount of points in the current run. A continuous contact with an obstacle (even of it moves the obstacle) will count as a single contact. The minimum number of points in one run can be zero. In total this challenge is repeated three times with different balls, but always with the same robots, which means that a team can be awarded up to a maximum of twelve points for this challenge. If teams have the same amount of points, the total time needed for all runs decides on the placement.

Challenge 7 - Play on an outside field

This challenge will only be offered, if it is possible by the organizers to provide a suitable field with respect to the actual field standards. Nevertheless, it is every time possible for teams to prepare by themselves an outside field of their choice which can also be smaller than the regular one. The team should be able to show solutions for typical problems on an outside field. To present these solutions it's not necessary to demonstrate them on a full size field, therefore a minimum field size is set to 10m×8m. If a team prepares a field by their own, even the type of ground can be selected by them, provided that it is the original type of ground of the chosen area, e.g. concrete, artificial turf, etc.
Teams that want to perform in this challenge have to contact the organizing or technical committee as soon as possible, because the resulting scenario needs to be approved by the technical committee.
Each of the above mentioned four abilities can be awarded with 2 points:

- The robots have to present that they are able to handle the ball according to the structure of the ground of the field.
- The robot has to approach a ball over distances of 2, 4, 6 and 8 meters.
- The robot has to avoid 3 obstacles.
- Summary of the above: The robot has to approach the ball over a distance of approx. 4m, then it has to score a goal after crossing a distance of approx. 6m, avoiding 3 randomly placed obstacles on the field and one obstacle in the goal.

A team has six minutes to demonstrate these abilities of their robots.

Appendix

Tables

A 1 Table of Network-Addresses

General Setup:

- WEP encryption is turned off.
- Broadcast of SSID is turned on.
- Subnet mask normal PC: 255.255.255.0.
- Subnet mask of a PC connected to the Refbox: 255.255.0.0.
- Access Point Beacon Interval should be set to 20-30.
- Access Point DTIM Interval should be set to 2-3.

Organization - Network Setup:

	Field A	Field B	Field C	Field D
SSID	MSL FIELD A	MSL FIELD B	MSL FIELD C	MSL FIELD D
Switch	172.16.1.1	172.16.2.1	172.16.3.1	172.16.4.1
Referee Box	172.16.1.2	172.16.2.2	172.16.3.2	172.16.4.2
Access Point 1 (802.11b)	172.16.1.3	172.16.2.3	172.16.3.3	172.16.4.3
Access Point 2 (802.11a)	172.16.1.4	172.16.2.4	172.16.3.4	172.16.4.4

Team - Network Setup for unicast communication:

Address	Team	Address	Team
172.16.32.*	Organization	172.16.54.*	Paderkicker
172.16.33.*	5DPO	172.16.55.*	Persian Gulf (IAUT)
172.16.34.*	AIS/BIT Robots	172.16.56.*	Philips Robocup Team
172.16.35.*	AllemaniACs	172.16.57.*	The Orient
172.16.36.*	Team Aros	172.16.58.*	Osaka University Trackies
172.16.37.*	Attempto Tubingen	172.16.59.*	Robofoot EPM
172.16.38.*	Brainstormers Tribots	172.16.60.*	Satrap
172.16.39.*	CAMBADA	172.16.61.*	Smoking Jays
172.16.40.*	Carpe Noctem Cassel (CNC)	172.16.62.*	Su-Spada
172.16.41.*	RFC Stuttgart	172.16.63.*	Tech United Eindhoven
172.16.42.*	EIGEN	172.16.64.*	TKU-ITRI
172.16.43.*	FU-Fighters	172.16.65.*	Ulm Sparrows
172.16.44.*	Hibikino-Musashi	172.16.66.*	WinKIT
172.16.45.*	ISePorto	172.16.67.*	Water
172.16.46.*	SocRob	172.16.68.*	Adro
172.16.47.*	Jiao Long	172.16.69.*	Endeavor
172.16.48.*	Khorasgan University	172.16.70.*	Hong Kong Dragons
172.16.49.*	MINHO	172.16.71.*	MU Penguins
172.16.50.*	Mostly Harmless	172.16.72.*	Strive-Legends
172.16.51.*	MRL	172.16.73.*	BIT-AC
172.16.52.*	MRT - Milan Robocup Team	172.16.74.*	
172.16.53.*	NuBot	172.16.75.*	

Team - Network Setup for multicast IPv4 communication:

Address	Team	Address	Team
224.16.32.32	Organization	224.16.32.54	Paderkicker
224.16.32.33	5DPO	224.16.32.55	Persian Gulf (IAUT)
224.16.32.34	AIS/BIT Robots	224.16.32.56	Philips Robocup Team
224.16.32.35	AllemaniACs	224.16.32.57	The Orient
224.16.32.36	Team Aros	224.16.32.58	Osaka University Trackies
224.16.32.37	Attempto Tubingen	224.16.32.59	Robofoot EPM
224.16.32.38	Brainstormers Tribots	224.16.32.60	Satrap
224.16.32.39	CAMBADA	224.16.32.61	Smoking Jays
224.16.32.40	Carpe Noctem Cassel (CNC)	224.16.32.62	Su-Spada
224.16.32.41	RFC Stuttgart	224.16.32.63	Tech United Eindhoven
224.16.32.42	EIGEN	224.16.32.64	TKU-ITRI
224.16.32.43	FU-Fighters	224.16.32.65	Ulm Sparrows
224.16.32.44	Hibikino-Musashi	224.16.32.66	WinKIT
224.16.32.45	ISePorto	224.16.32.67	Water
224.16.32.46	SocRob	224.16.32.68	Adro
224.16.32.47	Jiao Long	224.16.32.69	Endeavor
224.16.32.48	Khorasgan University	224.16.32.70	Hong Kong Dragons
224.16.32.49	MINHO	224.16.32.71	MU Penguins
224.16.32.50	Mostly Harmless	224.16.32.72	Strive-Legends
224.16.32.51	MRL	224.16.32.73	BIT-AC
224.16.32.52	MRT - Milan Robocup Team	224.16.32.74	
224.16.32.53	NuBot	224.16.32.75	

Tech United Eindhoven Team Description 2014

Middle Size League

Cesar Lopez Martinez, Ferry Schoenmakers, Gerrit Naus, Koen Meessen, Yanick Douven,
Harrie van de Loo, Dennis Bruijnen, Wouter Aangenent, Joost Groenen, Bob van Ninhuijs,
Matthias Briegel, Rob Hoogendijk, Patrick van Brakel, Rob van den Berg, Okke Hendriks,
René Arts, Frank Botden, Wouter Houtman, Marjon van 't Klooster, Jeroen van der Velden,
Camiel Beeren, Lotte de Koning, Olaf Klooster, Robin Soetens, René van de Molengraft

Eindhoven University of Technology,
Den Dolech 2, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
www.techunited.nl, techunited@tue.nl

Abstract. In this paper we discuss improvements in mechanical, electrical and software design of our middle-size league robots. Regarding hardware and control recent progress includes a prototype design of a ball clamping system, and first steps towards improved passing accuracy via velocity feedback control on the shooting lever. In terms of intelligent gameplay we have worked on creating possibilities for in-game optimization of strategic decisions. Via qr-code detection we can pass coaching instructions to our robots and with a basic machine learning algorithm success and failure after free-kicks is taken into account. In the final part of this paper we briefly discuss progress we have made in designing a four-wheeled soccer robot with a suspension system and on a smartphone application which real-time visualizes robot status and game state.

Keywords: RoboCup Soccer, Middle-Size League, Multi-Agent Coordination, Mechatronic Design, Motion Control, Machine Learning, Human-Robot Interaction, QR-Codes

1 Introduction

Tech United Eindhoven is a RoboCup team of Eindhoven University of Technology. Our team consists of PhD, MSc and BSc students, supplemented with academic staff members from different departments. The team was founded in 2005, originally only participating in the Middle-Size League (MSL). Six years later service robot AMIGO was added to the team, which since also participates in the RoboCup@Home league [4]. Knowledge acquired in designing our soccer robots proved to be an important resource in creating a service robot [3].

This paper describes our major scientific improvements over the past year. It is part of the qualification package for the RoboCup 2014 World Championships in Brazil, and contains five main sections. First we introduce our current robot platform, followed by a description of the robot skills we have improved recently (we will focus on ball clamping and accurate shooting). Hereafter we describe our progress in strategy and human-robot interaction, followed by a section on advancements in a new four-wheeled soccer robot platform we are designing in collaboration with an industrial partner. Lastly we show an iOS application dubbed 'iGreenField', which we wrote as a lightweight visualizer of robot status and perceived game-state.

Many of the points of improvement described in this paper are a direct result of rulechanges. In 2012 the mid-line passing rule was introduced, which was a large boost for the league in terms of stimulating smart team-play. Enforcing teams to make a pass before scoring provides an interesting academic challenge, but it also makes the matches more fun to watch for spectators.[1] This years rule-changes limit continuous dribbling distance, allow robot coaching along channels that are natural to human beings and replace the mid-line passing rule by a more general 'pass

---

[1] `http://youtu.be/UagXSjp9nfk` (Final Match RoboCup 2013 in Eindhoven)

before scoring' rule. The combination of rule changes introduce human-robot interaction to the middle-size league (Section 4.2), move the competition towards an even higher level of multi-agent coordination (Section 4) and push hardware design (Section 3.2).

2 Robot Platform

Our robots have been named TURTLEs (acronym for Tech United RoboCup Team: Limited Edition). Currently we are employing the fifth redesign of these robots, built in 2010, together with a goalkeeper robot which was built one year later (Fig. 1).

Three 12V Maxon motors, driven by Elmec Violin 25/60 amplifiers and two Makita 24V, 3.3Ah batteries, are used to power our omnidirectional platform. Our solenoid shooting mechanism, powered by a 450V, 4.7mF capacitor, provides an adjustable, accurate and powerful shot [5]. Each robot, except for the goalkeeper, is equipped with an active ball handling mechanism, enabling it to control the ball when driving forwards, while turning, and even when driving backwards [1]. Last year we redesigned this mechanism to improve its ball-catching abilities. As said before, this year we are aiming to improve its ball-clamping abilities (Section 3.2). We are also conducting experiments on directly catching lob balls with our ball handling system (Section 3.1.1).

To acquire information about its surroundings, the robot uses an omnivision unit, consisting of a camera focussed on a parabolic mirror [2]. An electronic compass is implemented to differentiate between omnivision images on our own side versus on the opponent side of the field. Recently we also added a kinect sensor to each robot. A detailed list of hardware specifications, along with CAD files of the base, upper-body, ball handling and shooting mechanism, has been published on a ROP wiki.[2]

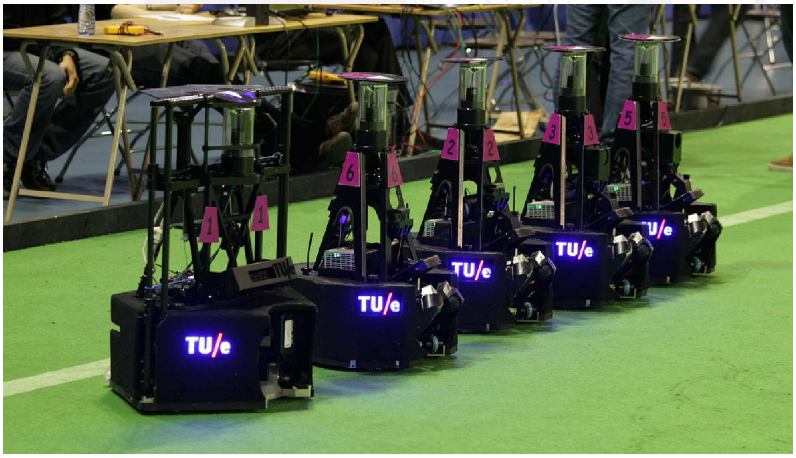


Fig. 1. Fifth generation TURTLE robots, with on the left the goalkeeper robot.

To facilitate data-acquisition and high-bandwidth motion control, the robots are equipped with EtherCAT devices provided by Beckhoff. These are connected to the onboard host computer via ethernet. Each robot is equipped with an industrial mini-pc running a preemptive Linux kernel. The software is automatically generated from Matlab/Simulink models via the RTW toolbox, recently renamed to 'Simulink Coder.' In order to allow asynchronous processing we have created a multitasking target for Simulinks code generation toolchain.[3]

Software for our robots is divided in three main executables: Vision, Worldmodel and Motion. On-board and robot-to-robot they communicate via a real-time database tool made by the CAMBADA team [6]. The vision module provides a localization of ball, obstacles and the robot itself.

---

[2] http://www.roboticopenplatform.org/wiki/TURTLE

[3] `http://www.techunited.nl/wiki/index.php?title=MultiTasking_Target_for_Linux`

Hereafter the worldmodel combines this information with data acquired from other team members to get a unified representation of the world. While vision runs at 60Hz and worldmodel at 20Hz, motion contains the controllers for shooting, ball handling en driving. Therefore it samples at a much higher rate (1000Hz). On top of the controllers, the motion executable also contains strategy and pathplanning, partly implemented as a subtask running at a much lower sample rate.

3 Improved Skills

Considering this years rule changes, it is likely that passing and catching will become increasingly important compared to dribbling. In the section below, we will describe how we are preparing for that by improving our accuracy for flat passing and by increasing our abilities to accurately catch and shoot a lob ball. The latter is not only beneficial for passing but also for shots at goal.

In the final paragraph of this section we describe an improved ball clamping mechanism, which could help winning or preventing scrums and reduce aim time for passing or shots on goal.

3.1 Shooting

The electrical scheme of our kicker consists of a battery pack charging a capacitor via a DC-DC converter (Fig. 2). Once fully charged, in roughly 20 seconds, the capacitor can be discharged via an IGBT switch, creating a pulse-width modulated signal. The energy of the capacitor drives a solenoid actuator connected to a mechanical transmission (a shooting lever). The lever can be adjusted in height to allow for lob- and flat shots.

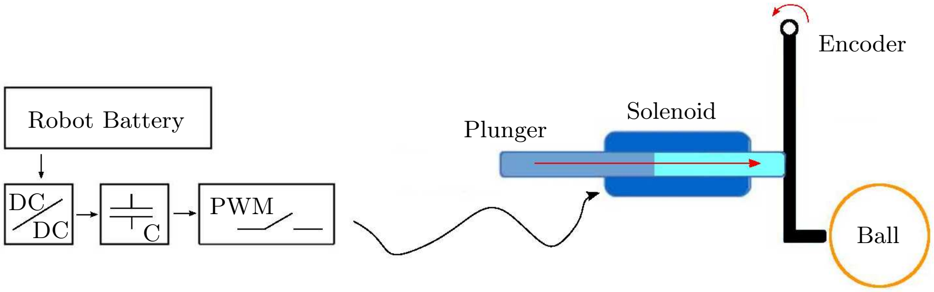


Fig. 2. Schematic overview of our shooting system. One half of the plunger is made of a non-magnetic material, the other half consists of a soft-magnetic material.

3.1.1 Shoot Lob Balls

To accurately shoot lob balls, the shooting system needs to be calibrated. Preferably we do this under conditions as close as possible to the conditions our robots face during the matches, i.e., on the official field with the same ball that will be used for competition. But during a tournament, testing time on the field is limited. Therefore our approach was to simply put the robot at the maximum distance it could take a lob shot from during a game, tune the PWM duty cycle until the ball lands exactly in the goal, and store the resulting duty cycle value. By linear interpolation between zero and the duty cycle we obtained during calibration, we could shoot from any spot within shooting range. The same calibration was used for all robots.

Although the above method is fast, it is also inaccurate. The relation between shooting distance and required duty cycle is non-linear, and since each robot has its own mechanical and electrical components, each robot has its own shooting characteristics. Therefore, calibration of each robot individually would be better.

For the upcoming season we designed and implemented a tool to quickly do robot-dependent calibration. Furthermore, empirically we identified the relation between the shooting distance (x) and the required duty cycle (u) is exponential for a lob-shot (Eq. 1). Parameters a, b and c are robot-dependent parameters. They have to be obtained by measuring the travelled distance for multiple duty cycles. To make a correct fit at least four measurements are required, though more are preferred.

$$u = b^{-1}\ln(a^{-1}(c - x)) \tag{1}$$

3.1.2 Catch Lob Balls

During last years technical challenge we showed an initial attempt to shoot and catch lob passes. In terms of catching the ball, our approach there was to simply wait until the ball bounces were low enough to simply intercept it as if it were a flat pass. Building on these first tries, this year we a working on a much more challenging lob pass approach. Our goal is to use our current ball handling system to grab the ball exactly when it hits the ground after a flight-phase. We call this coordinate the point of intercept (POI).

The teammate shooting the lob ball communicates to the receiving robot where the ball is expected to land (the feedforward position, FFP). When consecutive bounces are taken into account, multiple FFP's exist (example in Fig. 3). Each of them has a certain inaccuracy, for now modelled as a circle around the point itself. Based on the estimated time to reach each of the FFP's, the receiving robot drives towards one of the feedforward points when a lob ball is expected, but not actually shot yet.

Once the ball is in-air, a kinect camera mounted on the receiving robot is used to measure the ball position. Based on these observations, a simplified ball model, without drag and spin, predicts the ball trajectory. The receiving robot will respond to this ball-tracking based POI prediction, but only if it is located within the uncertainty circle. In case the estimated POI is located outside the circle, the robot will wait at the edge of the circle.

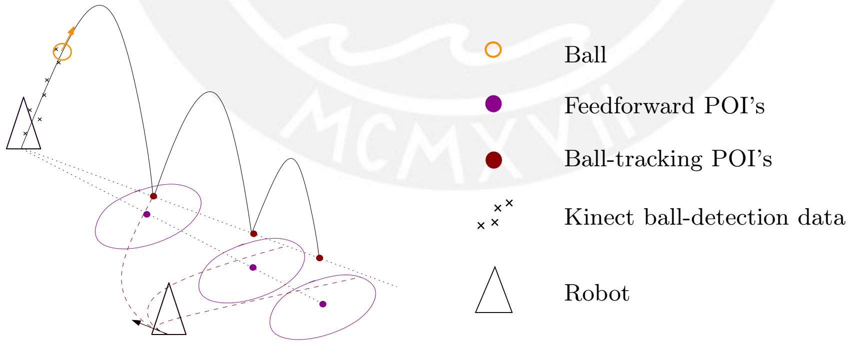


Fig. 3. Lob ball intercept strategy, the receiving robot chooses one of the points of intercept.

3.1.3 Shooting-Lever Velocity Feedback Control

Similar to what we described for lob shots, currently our control for flat shots and for flat passes is fully based on feedforward. As said before, many disturbances are robot-, ball- or field-dependent. Feedback control would allow to compensate for those.

This year we are planning to use an encoder mounted on the rotational joint of the shooting lever (Fig. 2) as a feedback signal for velocity control. For full-power shots the end-effector of the shooting lever is pushed into the ball almost entirely before the ball itself even starts to move.[4] Using lever angular velocity as a feedback signal to control the resulting ball velocity

---

[4] `http://youtu.be/MF7mfItBriA` (High-speed video of a full-power shot.

would be hard in this case, because it is hard to exactly predict the dynamic behaviour of the deformed ball. For slow shooting on the other hand, it is possible to make the lever and ball move as one body before the ball leaves the robot. Especially for passing, being able to accurately control ball-velocity would be of great help.

At the time of writing we are testing multiple controllers in simulation. What is particularly challenging is the limited time one has available (a shot takes between 20 and 50ms) and the limited spatial resolution of the encoder (130 ticks over the entire shooting lever stroke). Furthermore the solenoid actuator can only push in a single direction, therefore no overshoot is allowed.

In the upcoming months we are planning to finalize this controller. If necessary we will also use a redesigned shooting lever end-effector to improve passing accuracy (Fig. 4).

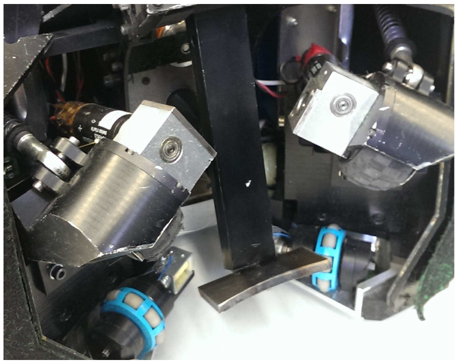


Fig. 4. Shooting lever end-effector for more accurate passing.

3.2 Ball Clamping

Temporary clamping the ball helps in winning scrum situations, but it also allows the robot to rotate faster around its vertical axis while in possession of the ball. Therefore it does not only help winning scrums, it can also help preventing them by allowing the robot to quickly turn away when an opponent approaches. Fast yaw-rotation could also decrease aiming time when giving a pass, or shooting on goal. Lastly, better clamping can also result in a more reproducible and more field-independent ball-retraction, which would make shooting and passing more accurate.

To achieve this clamping ability, a prototype system which works together with the original ball handling mechanism was designed. By actively controlling two contact wheels an additional force F_{cw} can be applied to the ball. Together with the force F_{bh} applied by the original ball handling system, a resulting force F_{res} lifts the ball from the ground and keeps it within the robot (Fig. 5).

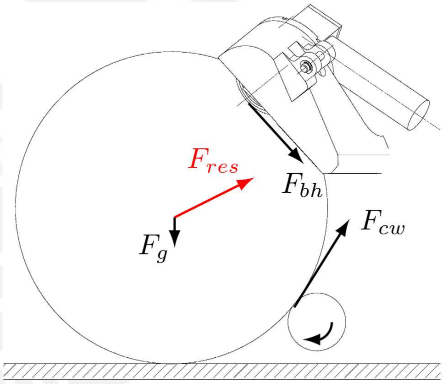


Fig. 5. Ball clamping forces.

In a recent test we matched up a robot equipped with the ball-clamping mechanism with a robot not having this mechanism. Ball-clamping resulted in winning an additional 25% of scrum situations. Currently we are working on improving control for the contact wheels such that they do not hinder normal ball-dribbling.

4 Improved Strategy

Our strategy takes into account the estimated positions of all peers and opponents, represented in a worldmodel. We are currently developing a method to also use velocities and estimated game state to assess the feasibility of various tactical actions (plans). Instead of instantaneously seeking the free space on the field, we would like to seek for an opportunity to create and employ it.

As a first step in moving to a more plan-based level of cognition, we have created a skill-selector, which we will describe in the upcoming section. Further we work on in-game optimization of decision making in refbox tasks, either via human coaching (Section 4.2) or via machine learning (Section 4.3).

4.1 Skill Selector GUI

In our strategy, first we assign a unique role to each of the robots. Every role contains a number of actions/skills which can be executed during play. The main attacker for instance has five different skills to choose from: Flat shot, lob shot, pass, dribble and push-attack (i.e., bouncing the ball towards the goal with the side of the robot).

To decide on which skill to use at a certain moment in time, hard-coded conditional statements are evaluated. For the original system, these conditions were solemnly true/false evaluations (e.g., to shoot at goal, there must be a clear path to the goal). They are evaluated in the order they appear in programming and therefore immediately discard all other possible actions. This creates situations for which the TURTLEs do not take the optimal action. In order to solve this problem, a more generic framework for skill-selection has been developed.

In our improved skill-selector framework, for each of the skills the hard-coded conditions are complemented with normalized ranking functions (e.g., while turning towards the goal, the ranking for shooting at the goal will increase). After evaluating all ranking functions the skill selector chooses the skill with the highest overall ranking. In case multiple rankings are the same, the default skill 'dribble' will be selected. To make sure the chosen skill consistently ranks higher than the current skill, a hysteresis function has been added.

For debugging and tuning purposes we have created a graphical user interface which visualises skill-selector output for a given game state (Fig. 6).

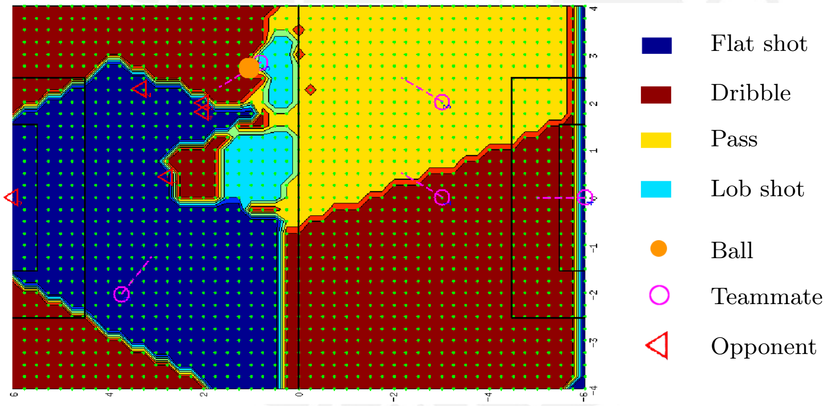


Fig. 6. Skill-selector visualization.

4.2 Human Coaching

This years rulebook opens doors for human-robot coaching in RoboCup MSL. Coaching instructions are intended to pass high-level instructions like 'shoot more often,' as opposed to low-level commands like 'shoot now.' As a first step, this year we will use qr-codes to tell our robots which predefined play to use, e.g., during a free-kick.

We use a freely available open source library to scan a video stream coming from our robots kinect sensor to scan for qr-codes.[5] With the maximum allowed qr-code size (i.e., 30x30cm), containing three chars of encoded information, we experimentally searched for the maximum distance for which the code could be scanned. Averaged over 35 trails, using seven different char-combinations, this distance turned out to be 5.1 metres (with a standard deviation of 0.29). False positives within the code detection regularly occurred, especially against a non-plain background. But since none of these false positives matched any of the known strings, we could simply keep scanning until a combination of symbols was recognized that was actually grounded in the robots knowledge base.

---

[5] `http://zbar.sourceforge.net/` (ZBar, open source bar- and qr-code reader)

In any trial of the experiment, if the code got detected, it was recognized within four seconds. Since in the current rules coaching is only allowed during 'dead time' between stop and start of a refbox task, we were interested how often a robot could actually get within five metres from the coaching spot and stay there for at least four seconds to receive a coaching instruction. Therefore we looked back at logged data of last years final match. In total this match involved 58 refbox tasks, 21 of them did not involve direct scoring risk (i.e., at least one of our robots was available to come to the side for coaching). Taking into account constraints on the robots acceleration and velocity, with our current qr-code detection system 17 coach moments would have succeeded. Probably enough to tell our robots to quit using those double passes that were continuously interrupted by the robots of team Water.

4.3 Learning Refbox Play Decisions

In the previous section we described a way to do a hard, human-imposed, reset within our robots decision making. On top of these hard resets, we are also working on a basic reinforcement learning algorithm for a more subtle optimization of strategic play-choice during refbox tasks.

A reinforcement learning algorithm is built around actions, states and rewards [7]. Applying this framework to our free-kick strategy, we use six existing refbox plays as our action-space (single kick and shoot, double kick and pass etc). Based on which opponent we face and the location of the free-kick (state), one play may result in slightly better scoring chances than the others. As a reward function we give high virtual reward for a scored goal, lower reward for a shot attempt, small punishment for loss of ball possession and severe punishment for a goal scored by the opponent (all weighted for time passed after the refbox task start signal).

Within this framework of rewards, states and actions we are currently able to store an expected reward for each state, based on past experience. In simulation, playing against our own team, this approach eventually outperformed our original heuristics based approach.

5 Four-Wheeled Platform with Suspension System

Fig. 7. Base structure.

Already since our first generation of soccer robots, we have been using a robot base with three omniwheels positioned in a triangle. Such a three-wheeled design makes control easier because, regardless of field irregularities, all of the wheels will maintain in touch with the ground. But disadvantages also exist. Although driving straight forward is the most common direction of acceleration, it is also the direction for which our three-wheeled robot experiences the least traction during acceleration. The robot tends to tilt backwards, putting most of its pressure on the only wheel that cannot be used to transfer a torque to the ground when driving forward. For our current robot-design, traction is the limiting factor in achieving higher acceleration.

For a four-wheeled base accelerating forward, i.e., in the direction of the ball-handling mechanism, additional pressure is put on wheels that are actively used in acceleration. We are currently collaborating with an industrial partner to realize a prototype of a four-wheeled robot.[6] On top of the RoboCup rulebook requirements with respect to weight and size, an additional requirement was created: Without any of the wheels losing contact with the floor, the robot should be able to take bumps of at least 10mm in any direction while maintaining a ground clearance of 15mm.

To meet this latter requirement, a suspension system is needed. In the current prototype design, each of the wheels is equipped with an independent suspension system. Wheels and motor are still directly connected via a gearbox but the combination of the two is connected to the base via a passive spring-damper combination. At the time of writing the prototype robot is being assembled.

---

[6] Prodrive-Technologies, Science Park Eindhoven

6 iGreenField app

We have created an iOS smartphone application. Initially this app was intended as a mobile version of our desktop 'GreenField' application for visualization. It shows the current position of peer and opponent robots and the position of the ball (Fig. 8). Also, the most recently planned path of each of the robots is visualized.

Fig. 8. iGreenfield

Originally, the app was fully based on the comm-framework developed by the CAMBADA team [6]. However, at some point we realized the app might also be nice to have for spectators during a match. To be able to handle a possible many-user scenario during RoboCup 2013 in Eindhoven, we decided to build a separate server. A laptop computer next to the field listens to the data packages from our robots, creates iGreenField-specific packages and sends them to the server application located at our university. Users of the app automatically connect to the server and receive the most recent packages.

A few weeks before RoboCup 2013 the app was released. In total it was downloaded 337 times during the tournament. A maximum of 35 simultaneous users has been observed. General users often do not recognise the actual purpose of the app. Having seen the visualisation they immediately conclude that the app offers a visualisation of the actual game state, where it represents the estimated game state as seen by the robots. As a result, the span of use was very limited for most users. Apps such as iGreenField have the potential to help explaining to the general public where RoboCup is all about, but apparently more thought must be given to the user interface in order to accomplish this. At the moment, the app is more useful for ourselves, as a lightweight replacement for the GreenField laptop application.

7 Conclusions

In our team description paper we have discussed concrete steps towards more accurate shooting which, together with better ball tracking abilities, could enable passing via lob balls. Also we have presented proof of concept experiments for qr-code based human coaching, for learning algorithms in refbox strategy and for a new ball clamping mechanism.

Altogether we hope our progress contributes to an even higher level of dynamic and scientifically challenging robot soccer during RoboCup 2014 in Brazil. While at the same time maintaining the attractiveness of our competition for a general audience.

References

1. J.J.T.H. de Best, M.J.G. van de Molengraft, and M. Steinbuch. A Novel Ball Handling Mechanism for the RoboCup Middle Size League. *Mechatronics*, 21(2), 2011.
2. Dennis Bruijnen, Wouter Aangenent, Jeroen van Helvoort, and René van de Molengraft. From Vision to Realtime Motion Control for the RoboCup Domain. In *IEEE International Conference on Control Applications*, pages 545–550, Singapore, 2007.
3. Janno Lunenburg, Robin Soetens, Ferry Schoenmakers, Paul Metsemakers, René van de Molengraft, and Maarten Steinbuch. Sharing Open Hardware through ROP, the Robotic Open Platform. In *RoboCup Symposium*, Eindhoven, 2013.
4. J.J.M. Lunenburg, S.A.M. Coenen, T.T.J. Derksen, S. van den Dries, J. Elfring, and M.J.G. van de Molengraft. Tech United Eindhoven @Home Team Description. *RoboCup*, 2014.
5. K.J. Meessen, J.J.H. Paulides, and E. Lomonova. A football kicking high speed actuator for a mobile robotic application. *Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society*, pages 1659–1664, 2010.
6. António JR Neves, José Luís Azevedo, Bernardo Cunha, Nuno Lau, João Silva, Frederico Santos, Gustavo Corrente, Daniel A Martins, Nuno Figueiredo, Artur Pereira, et al. Cambada soccer team: from robot architecture to multiagent coordination. *Robot Soccer*, pages 19–45, 2010.
7. Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning.* MIT Press, 1998.

Chapter 3

Wheel configuration

The wheel configuration determines the Turtle's base plate shape. Currently, three omniwheels are mounted under 120 ° with respect to each other. This is not the only wheel configuration that can be used. In this Chapter, a comparison between different wheel configurations is made. First the configurations are explained and their inverse kinematics are given. Next, they are compared based on top speed, acceleration and cornering speed.

3.1 Different wheel configurations and their kinematics

Most RoboCup middle size league robots nowadays have three or four omniwheels. In the past, there were also teams that used a three wheeled platform with two regular wheels and a castor wheel or omniwheel for support at the back. These three main wheel configurations are depicted in Figure 3.1.

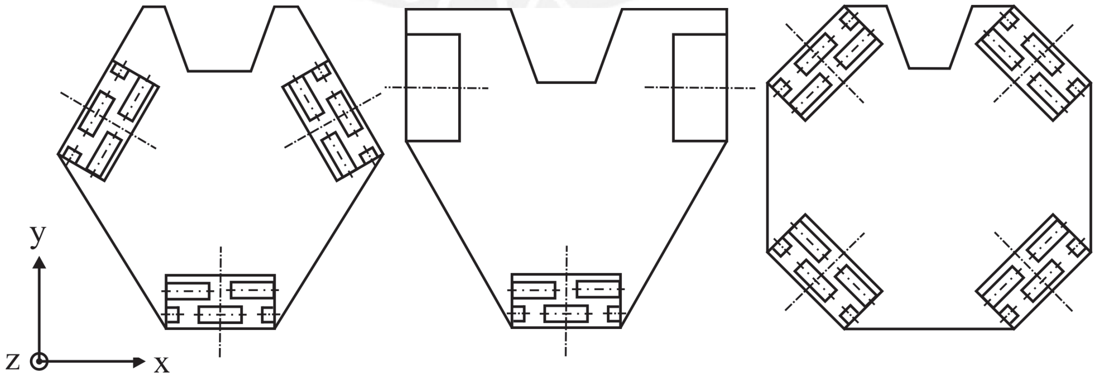


Figure 3.1: The three main wheel configurations with from left to right: The platform with three omniwheels, the three wheeled platform with two regular wheels and an omniwheel and the platform with four omniwheels.

3.1.1 Platform with three omniwheels

The base plate of this type of platform is an equilateral triangle. The omniwheels are attached to its corners. This makes this platform holonomic, it can move in every direction without turning first. The inverse kinematics are required to analyse the kinematic behavior of the platform. They determine the angular velocity of the wheels related to the global platform velocity and heading. It can also be used the other way around. The angular velocities of the wheels are the input, resulting in a velocity and heading of the platform. A kinematic diagram of the platform is required to derive the inverse kinematics. Figure 3.2 shows the kinematic diagram of the platform with three omniwheels. The kinematics were derived in [4].

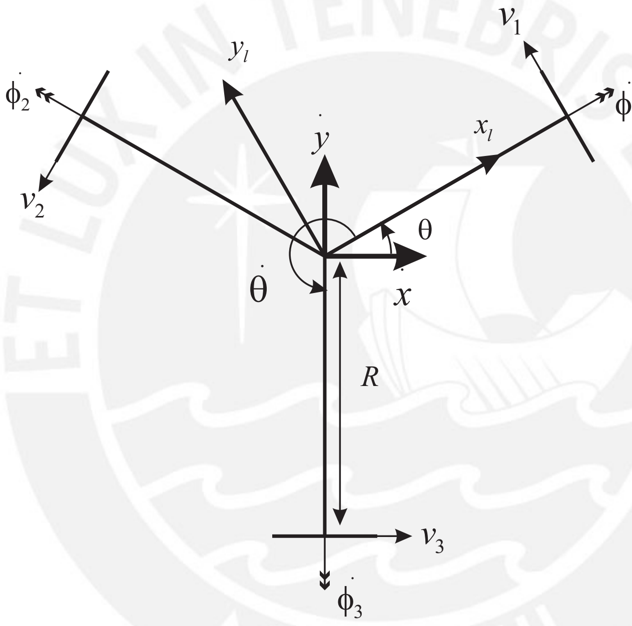


Figure 3.2: Kinematic diagram of the platform with three omniwheels.

The starting point of this derivation is the inverse kinematic equation of a single wheel (Equation 3.1). This determines the translational velocity v of the wheel with respect to the global velocity $[\dot{x}, \dot{y}.\dot{\theta}]$ of the platform. It is a combination of translation and rotation of the platform.

$$v_i = -\sin(\theta + \alpha_i)\dot{x} + \cos(\theta + \alpha_i)\dot{y} + R\dot{\theta} \tag{3.1}$$

Where θ is the angle between the global [x,y] and local $[x_l, y_l]$ frame attached to the platform (see Figure 3.2), α is the angle of wheel i relative to the local frame. Since this frame coincides with the C.O.G. of the platform and wheel 1 is located on the local axis (x_1), $\alpha_1 = 0\,^\circ$. This means that $\alpha_2 = 120\,^\circ$ and $\alpha_3 = 240\,^\circ$. Finally, R is the distance between the platform's C.O.G. and the center of a wheel.
The translational velocity of the wheel can be rewritten as its angular velocity ω_i using Equation 3.2 resulting in Equation 3.3.

$$v_i = r\omega_i \tag{3.2}$$

Where r is the radius of the wheel.

$$r\omega_i = -\sin(\theta + \alpha_i)\dot{x} + \cos(\theta + \alpha_i)\dot{y} + R\dot{\theta} \tag{3.3}$$

This can be transformed in to a matrix representation (Equation 3.4):

$$\vec{\omega_i} = \mathbf{J_{inv}}\vec{\dot{u}} \tag{3.4}$$

In this Equation, $\mathbf{J_{inv}}$ is the inverse Jacobian for the holonomic platform. It provides the relationship between the angular velocities of the wheels $\vec{\omega_i}$ and the global velocity vector $\vec{\dot{u}}$. The total matrix representation is described by Equation 3.5:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta + \alpha_1) & \cos(\theta + \alpha_1) & R \\ -\sin(\theta + \alpha_2) & \cos(\theta + \alpha_2) & R \\ -\sin(\theta + \alpha_3) & \cos(\theta + \alpha_3) & R \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{3.5}$$

3.1.2 Three wheeled platform with regular wheels

This type of platform has two regular wheels, which can have a higher friction coefficient than the omniwheels. Therefore, more traction and a theoretically higher acceleration is possible. This makes it interesting to investigate this type of platform.
The platform steers like a tank. It cannot move instantaneously in any direction. Therefore, it is called a non-holonomic platform. It turns by introducing a difference in the angular velocity of the front wheels. They are positioned in front of the C.O.G. Therefore, a third wheel is necessary to support the platform and to keep its back end of the ground. Generally, a castor wheel is used. It is also possible to replace it by an omniwheel. This does not affect the forward motion, but it can assist when making a turn. The θ rotation axis around which there is the lowest inertia, is a line through the C.O.G.. Because the front wheels are in front of that, they have to overcome a larger inertia when turning the platform. The extra omniwheel assists to speed up the θ rotation.

A kinematic diagram of this platform is depicted in Figure 3.3.

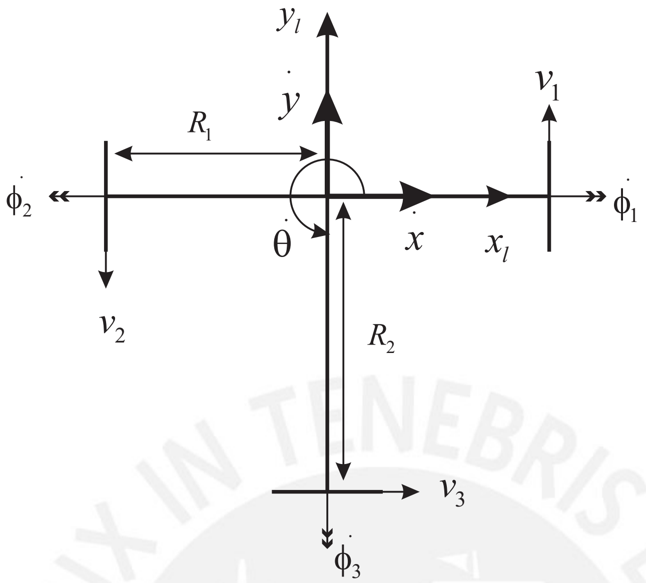


Figure 3.3: Kinematic diagram of the three wheeled platform with two regular wheels and an omniwheel.

The wheels of this platform are not positioned at an equal distance from the center of rotation, which is positioned in the origin of the local frame. The distance between the omniwheel and the center of rotation R_2 is larger than half the distance between the front wheels R_1. This results in a small change in the inverse kinematics with respect to that of the previous platform. The inverse kinematics are described by Equation 3.6.

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\end{bmatrix} = \frac{1}{r}\begin{bmatrix}-\sin(\theta+\alpha_1) & \cos(\theta+\alpha_1) & R_1\\-\sin(\theta+\alpha_2) & \cos(\theta+\alpha_2) & R_1\\-\sin(\theta+\alpha_3) & \cos(\theta+\alpha_3) & R_2\end{bmatrix}\begin{bmatrix}\dot{x}\\\dot{y}\\\dot{\theta}\end{bmatrix} \tag{3.6}$$

In this situation, α_1 again is at the local axis, therefore $\alpha_1 = 0\,°$. This means that $\alpha_2 = 180\,°$ and $\alpha_3 = 270\,°$.

3.1.3 Platform with four omniwheels

The last type of platform is the one with four omniwheels. The wheels are positioned at the corners of a square (see Figure 3.1), their axes along the diagonals. In the two previous concepts the base plate was triangular shaped. Both the square and the triangle have to fit in the 500 mm x 500 mm square described in Section 2.4. This means that the base plate of the platform with four omniwheels can be larger than the base plates of the other concepts. It is also a holonomic platform. For this reason, it is interesting to investigate this wheel configuration.

The kinematic diagram is depicted in Figure 3.4. All wheels are positioned at an equal distance R from the center of the platform. This is also its center of rotation.

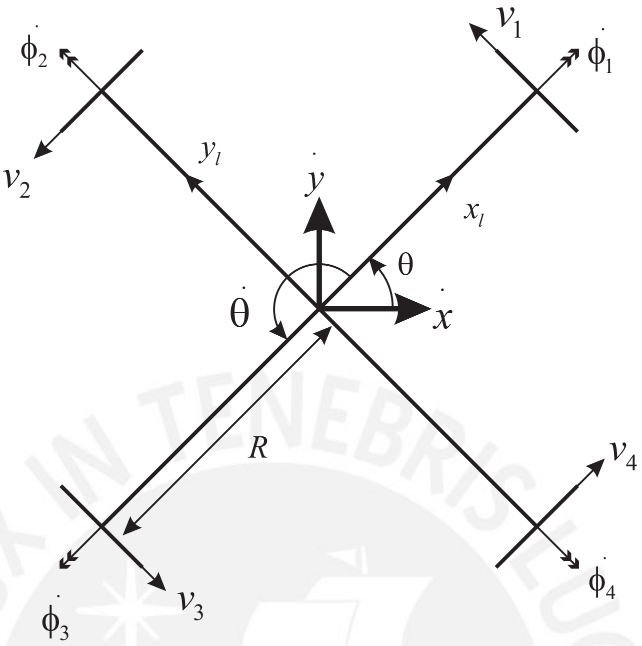


Figure 3.4: Kinematic diagram of the platform with four omniwheels.

The additional wheel introduces an extra Equation that describes the inverse kinematics of one wheel (Equation 3.1) in the matrix representation of the platform's inverse kinematics. This representation is given by Equation 3.7.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta+\alpha_1) & \cos(\theta+\alpha_1) & R \\ -\sin(\theta+\alpha_2) & \cos(\theta+\alpha_2) & R \\ -\sin(\theta+\alpha_3) & \cos(\theta+\alpha_3) & R \\ -\sin(\theta+\alpha_4) & \cos(\theta+\alpha_4) & R \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{3.7}$$

As in the previous inverse kinematic relations, α_1 is at the local axis. As a result, $\alpha_1 = 0\,°$. This implies that $\alpha_2 = 90\,°$, $\alpha_3 = 180\,°$ and $\alpha_4 = 270\,°$.

3.2 Rotation of the wheels

With the inverse kinematics it is possible to compare the rotation of the wheels of the different platforms when they perform a global translation. This will provide some insight in the behavior of the different concepts. To that purpose, the platforms have completed a simulated trajectory in y direction (specified in the kinematic diagrams) with a constant velocity. The angular velocities of the wheels of the different platforms are plotted. The trajectory and the plots are depicted in Figure 3.5.

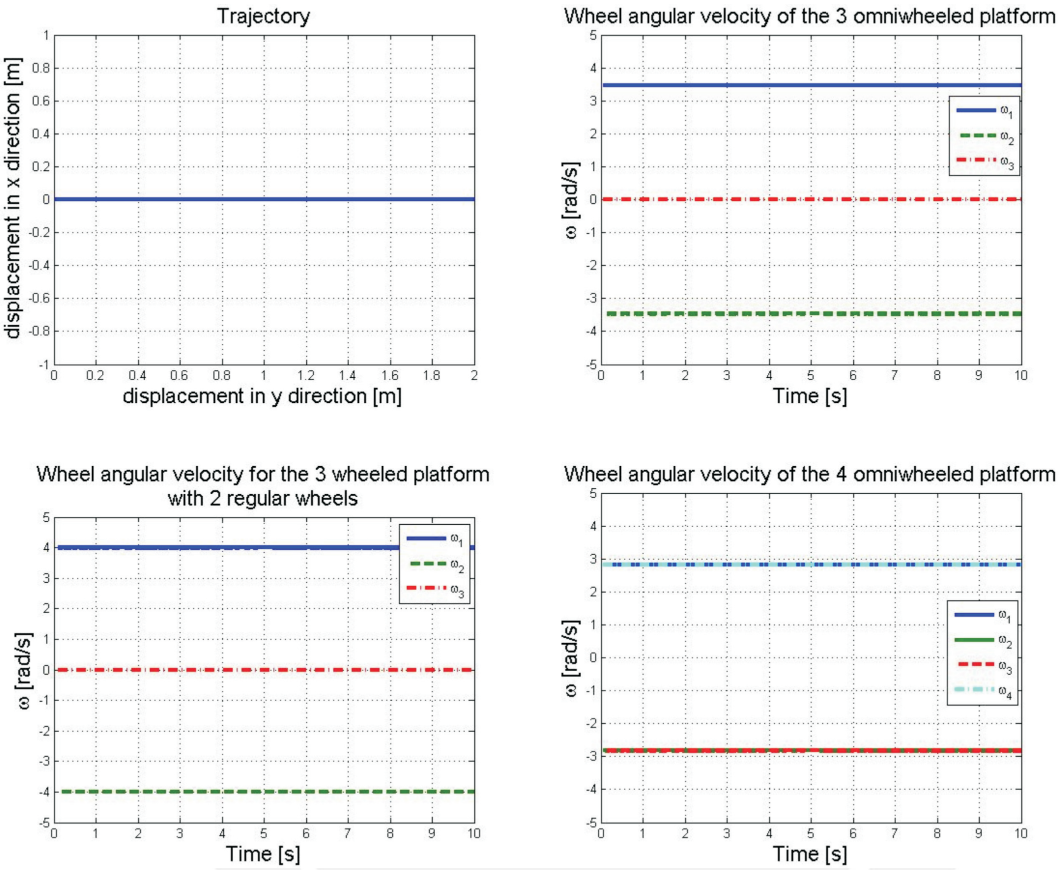


Figure 3.5: A straight line, constant speed trajectory and the resulting angular wheel velocities of the different platforms.

The plots show that the angular velocity of the third wheel of both the three wheeled platforms is zero. This was expected. These wheels are positioned perpendicular to the direction of motion and thereby, do not contribute to the propulsion of the platform in this particular direction. The left and right wheels of the platform with four omniwheels have equal but opposite angular velocities. This was also expected, because of the definition of the positive direction of the wheel's translational velocity in Figure 3.4.

The plots also show a difference in angular velocity of the wheels of the different platforms. The wheels of the three wheeled platform with two regular wheels rotate the fastest, followed by the wheels of the platform with three omniwheels. The wheels of the platform with four omniwheels rotate the slowest. This is against the expectations. The regular wheels are positioned in line with the direction of motion and it was expected that this would be the most effective way of propelling a vehicle in the forward direction. However, it seems that wheels positioned under an angle relative to the direction of motion, are more effective.

This phenomenon was described by Ashmore and Barnes [5]. They describe that a platform with omniwheels positioned under an angle relative to the direction of motion reaches a higher top speed compared to a platform with wheels in line with the direction of motion, assuming equal angular velocities of the wheels.

Like regular wheels, omniwheels transmit a torque in one direction. However, an omniwheel can move freely in the direction perpendicular to the torque vector. It then rolls over the small wheels on its perimeter. Regular wheels cannot do that because of the non-slip condition.

An omniwheel will perform a translation in both its actuated direction and in the direction perpendicular to this, when it is positioned under an angle relative to the direction of motion. This can be seen in Figure 3.6.

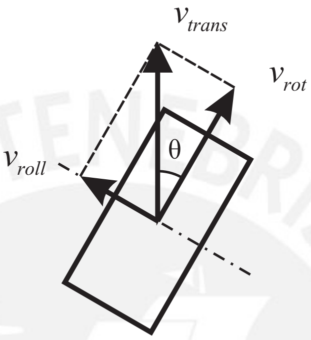


Figure 3.6: Translational velocity of an omniwheel under an angle θ relative to the direction of motion.

The translational velocity v_{trans} is a combination of the translational velocity in the regular rotational direction v_{rot} and the sideways rolling velocity v_{roll}. As a result, v_{trans} is larger than v_{rot} and increases when θ increases according to Equation 3.8.

$$v_{trans} = \frac{v_{rot}}{\cos\theta} \tag{3.8}$$

Ashmore and Barnes [5] state that a platform with four omniwheels (like the one described before) benefits the most from this increase in forward speed. In that case, θ varies between $0\,^{\circ}$ and $45\,^{\circ}$. If the platform moves in line with two omniwheels and perpendicular to the other two, $\theta = 0\,^{\circ}$ and it will reach the same velocity as the platform with two regular wheels. If it moves along the line of motion (both forwards and backwards) specified in Figure 3.5, or perpendicular to it, $\theta = 45\,^{\circ}$. This is a direct result of the symmetrical base plate.

A same distribution of θ around the platform exists for the platform with three omniwheels. However, in this situation θ varies between $0\,^{\circ}$ and $30\,^{\circ}$. If one wheel is in line with the direction of motion, $\theta = 0\,^{\circ}$ and if one wheel is perpendicular to the direction of motion, $\theta = 30\,^{\circ}$. The wheel with the smallest angle θ determines the translational velocity of the platform. For this reason it is of no use to add a fifth wheel. Locally this results in a θ larger than $45\,^{\circ}$, but at the same time there is also a θ that is smaller than $45\,^{\circ}$, which limit the velocity.

3.3 Acceleration

There is a downside to this obtained extra speed. The angle of the wheel with respect to the direction of motion can be regarded as an extra reduction that is added to the drive train of the wheel. It has a gear ratio equal to or larger than one depending of θ (Equation 3.9).

$$\frac{v_{trans}}{v_{roll}} = \frac{1}{\cos(\theta)} \tag{3.9}$$

The total reduction i_{tot} from the angular motor velocity ω_m to v_{trans} changes when θ changes. The total reduction is given by Equation 3.10.

$$\begin{aligned} i_{tot} &= \frac{v_{trans}}{\omega_m} \\ &= \frac{v_{trans}}{v_{roll}} \cdot \frac{v_{roll}}{\omega_m} \\ &= \frac{v_{trans}}{v_{roll}} \cdot \frac{v_{roll}}{\omega_w} \cdot \frac{\omega_w}{\omega_m} \\ &= \frac{1}{\cos(\theta)} \cdot r \cdot i_{gh} \\ &= \frac{i_{gh} r}{\cos(\theta)} \end{aligned} \tag{3.10}$$

Where i_{gh} is the gear ratio of the gear head and r is the radius of the wheel. ω_w Is the angular velocitie of the wheel. The total reduction of the drive train increases when θ is increased.

Each motor has to accelerate a load m, which is a part of the Turtle's total mass. It depends on the number of wheels that contribute to the propulsion. For example, if two wheels propel the Turtle, each motor has to accelerate a load equal to half the mass of the Turtle. The motor's angular acceleration α_m depends on the inertia of the rotor J_r and the reduced inertia of the load on the motor axle J_L it has to accelerate according to Equation 3.11.

$$\alpha_m = \frac{T_m}{J_r + J_L} \tag{3.11}$$

The angular acceleration also depends on the torque T_m produces by the motor. J_L depends on the load m and the total reduction of the drive train. It is described by Equation 3.12.

$$J_L = m i_{tot}^2 \tag{3.12}$$

$$a_w = \frac{\alpha_m}{i_{tot}} \tag{3.13}$$

From Equations 3.11 and 3.12 it is concluded that an increase of i_{tot} results in an increase of J_L. Correspondingly, α_m and the translational acceleration of the wheel a_w in the direction of motion decrease according to Equation 3.13. This means, though the platforms with omniwheels can reach higher end velocities, the platform with the regular wheels theoretically can accelerate faster.

3.3.1 Slippage of the wheels

The angle of the wheel is not the only factor that influences the acceleration of the platforms. The motor torque applied to the wheels is limited by the amount of torque the wheels can transfer to the field without slipping. The torque T_w applied produces a tangential force F_{tw} at the outer perimeter of the wheel at the point where it is in contact with the field (Equation 3.14).

$$F_{tw} = T_w r \tag{3.14}$$

$$T_w = \frac{T_m}{i_{gh}} \tag{3.15}$$

$$F_{fr} = F_N \mu \tag{3.16}$$

If this force exceeds the friction force (F_{fr} in Equation 3.16) in the contact, the wheel will slip. The magnitude of the friction force depends on the normal force F_N and the sliding friction coefficient μ_{kin} between the wheel and the field.
The small rollers of the omniwheels are made of rubber. To get an indication of sliding friction coefficient between omniwheel and the field, which is made of carpet, a simple experiment was performed (see Appendix B.2). This resulted in a μ_{kin} of 0.8. It is assumed that the regular wheels have a rubber outer surface which is rougher. The sliding friction coefficient of these wheels in contact with the field is estimated at 1.0 [8].

3.3.2 Normal force acting on the wheels

The normal force acting on a wheel depends on the number of wheels, the mass of the platform and the height of the C.O.G. when maneuvering. The masses of the different platforms are estimated referring to the mass of the current Turtle, some of its components and the number of wheels.
Currently, the mass of the Turtle is approximately 35 kg. The three wheeled platform with two regular wheels and the platform with three omniwheels have an equal number of wheels as the current Turtle. Therefore, their mass is estimated at 35 kg. The platform with four omniwheels has an extra wheel, drive train and amplifier. This results in an additional mass of approximately 1.6 kg. During forward motion, the platform with the four omniwheels uses twice the number of motors when driving in forward direction, compared to the other platforms. This results in an estimated double power use. To compensate for this, the number of batteries is doubled. This introduces an extra mass of 3 kg. Finally, its base plate larger. All together this platform has an additional mass of approximately 5 kg, giving it a total estimated mass of 40 kg.

In steady state, the platform's mass is equally distributed over the wheels. This is not the case during acceleration. The height of the C.O.G. above the field causes the platform to tilt. This is explained with the help of Figure 3.7 and Equation 3.17.

$$\Delta F_N = \frac{h}{l} mg \tag{3.17}$$

Equation 3.17 is the final result of an equilibrium of moments around one of the contact points with the field of the schematic platform in Figure 3.7. When the platform accelerates in forward direction, a force acts in opposite direction on the C.O.G.. Because of the equilibrium of moments, $F_{N,front}$ reduces with ΔF_N, while $F_{N,rear}$ increases with an equal amount. This difference is indicated in Figure 3.17 as $2\Delta F_N$. In this situation the front wheels can transmit less force from the motor to the field. The torque of the motor is

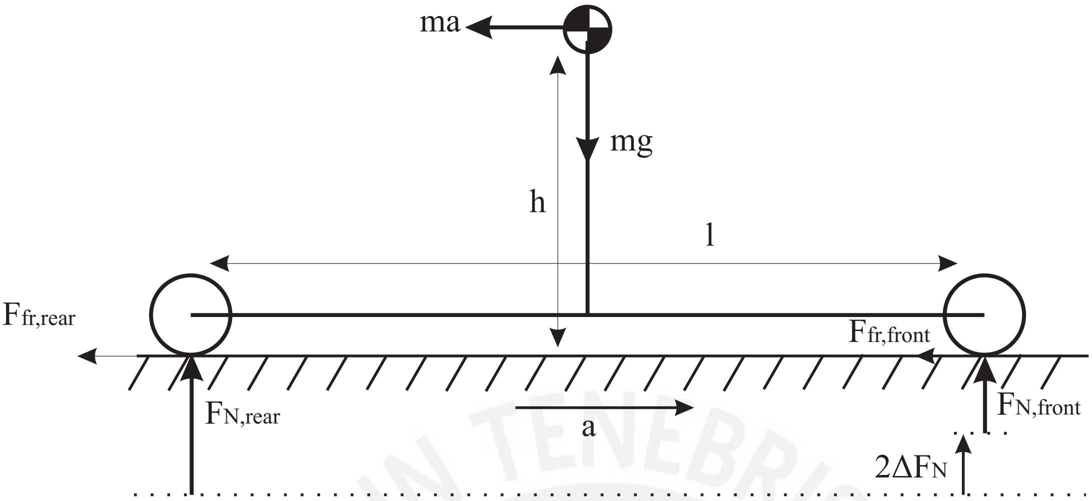


Figure 3.7: Height h of the C.O.G. of the platform above the field, the wheel base l of the platform and all forces that are applied to it during acceleration.

reduced to prevent slipping. This also affects the maximum acceleration possible, which decreases as well.
The height h of the C.O.G. of the current Turtle is determined, to get a reasonable indication which value should be used in the simulations. This was done with an experiment, which is described in Appendix B.1. The height is estimated at 150 mm above the field.
The wheel base l of the platforms are given in table 3.1.

Table 3.1: Wheel base of the different platforms.

Platform	Wheel base $[mm]$
Platform with three omniwheels	360
Three wheel platform with two regular wheels	400
Platform with four omniwheels	400

The wheel base of the platform with three omniwheels is shorter because of the dimensions of the largest equilateral triangle that fits in the 500 mm x 500 mm plane. The front wheels of the three wheeled platform with two regular wheels are positioned further to the front than those of the platform with three omniwheels, because they are not attached to an equilateral triangle.

3.3.3 Simulation of the acceleration

The motor torque is the input of the acceleration simulations. Currently, the Turtle's use Maxon RE40 150 W DC servo motors (see appendix A.1 for the specifications). These have to be simulated as well, because their maximum torque depends on the current applied to the motor and its angular velocity. The faster a motor rotates, the less torque it can deliver.

This relation is described by Equation 3.18 [6].

$$\omega = n_0 \frac{2\pi}{60} - \frac{\Delta n \frac{2\pi}{60}}{\Delta T_m} \frac{T_m}{1000} \tag{3.18}$$

Where ω is the current angular velocity of the motor, n_0 is the no load speed of the motor in rpm and $\frac{\Delta n}{\Delta T_m}$ is the torque gradient in rpm/mNm. The motor torque T_m (in Equation 3.18 in mNm) was determined with Equations 3.14, 3.15 and 3.16 by equating F_{tw} to F_{fr}. If the input torque is known, the angular acceleration of the motor (Equation 3.11) and the time it takes for the motor the reach the maximum angular velocity it can achieve with the current torque (Equation 3.19) can be determined [6].

$$\Delta t = \omega_m \frac{J_r + J_L}{T_m} \tag{3.19}$$

From this point on, the acceleration decreases linearly until its reaches zero at the time the motor reaches its nominal speed. This is regarded as the maximum angular velocity the motors can achieve in the simulations. With the known reduction of the drive train, this can be converted to a theoretical top speed of the platforms. The deceleration is linear, because the motor torque decreases linearly (Equations 3.11 and 3.18).
With Equation 3.19 it is calculated how much time it takes to reach the nominal speed. During each interval i of length Δt the acceleration of the motor is constant. This acceleration is converted to the platform's acceleration a_{wi} in that interval via Equation 3.13. Therefore it is possible to calculate the distance traveled by the platform during an interval (see Equation 3.20).

$$\begin{aligned} s_i &= 0.5 a_{wi} \Delta t_i^2 + v_{i-1} \Delta t_1 + s_{i-1} \\ v_i &= a_{wi} \Delta t_i + v_{i-1} \end{aligned} \tag{3.20}$$

s_i And v_i are calculated in an interval and function as s_{i-1} and v_{i-1} in the next interval. If i has reached the number of intervals, s_i gives the total traveled distance during acceleration. This method makes it possible to simulate a drag race between the different platforms. The result of this simulation is depicted in Figure 3.8.

The Figure shows that the three wheeled platform with two regular wheels accelerates the fastest, as was expected. It is the first one to reach its theoretical top speed of 3.0 m/s. The Figure also shows the higher top speeds of the other two platforms. The platform with four omniwheels is the first to catch up with the three wheeled platform. This happens after roughly 1.1 second. Both platforms have traveled 2.5 meter at that time. After roughly 1.7 second and a distance of 4 meter, the platform with three omniwheels catches up with the platform with the regular wheels. The platform with four omniwheels has a theoretical top speed of 4.3 m/s and the platform with three omniwheels has a theoretical top speed of 3.5 m/s.
Finally, the platform with four omniwheels accelerates faster than the platform with three omniwheels. This is caused by the number of driven wheels. The four motors of the platform with four omniwheels each have to accelerate less load compared to the two motors driving the platform with three omniwheels.
The catch up distances must be compared to the dimensions of the field to get an idea whether the acceleration differences are relevant. The official RoboCup middle size league

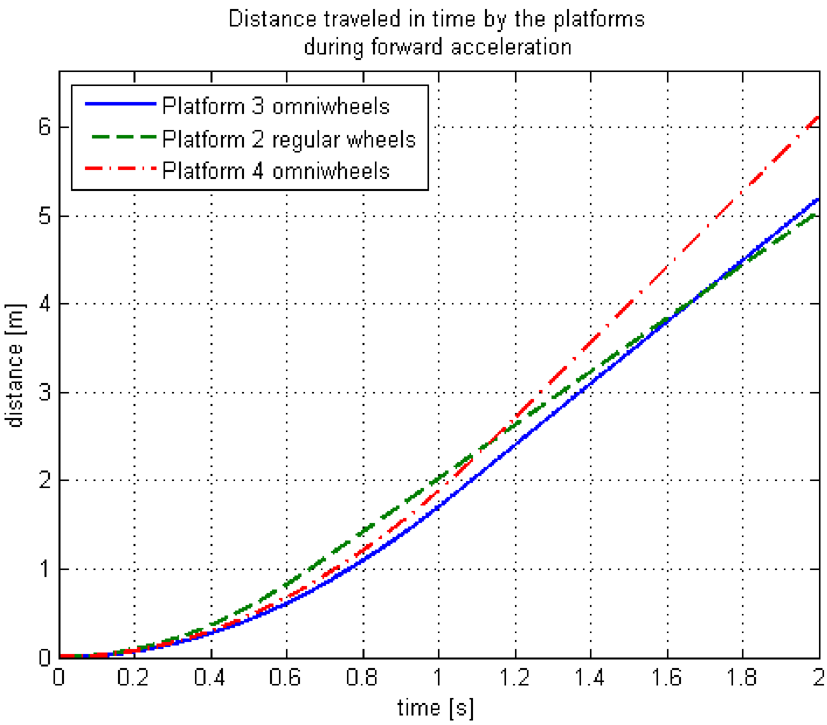


Figure 3.8: Distance traveled in time during forward acceleration by each of the platforms.

field measures 18 m x 12 m [3]. Robots mostly cover distances larger than 2.5 meters each time they head for a ball or try to score. Therefore, the platform with two regular wheels does not have that much of an advantage compared to the platform with four omniwheels. The advantage is more relevant compared to the platform with three omniwheels.
Usually, small distances are covered during defending. Robots move sideways over small distances to stay in line between the attacker and the own goal. As Figure 3.8 shows, there is not much difference between the platforms' performances when the traveled distance is less than one meter. The platforms also need to be vary maneuverable during defensive actions. The holonomic platforms satisfy this requirement, because they can instantly move in an other direction without turning first.

In the previous simulation, all platforms used the same motor. Therefore, they could not use the same amount of power during the forward acceleration. The platform with the four omniwheels has double the power compared the other two platforms. In the next simulation, this is compensated for by selecting different motors. The platform with regular wheels and the platform with three omniwheels have Maxon EC45 250 Watt motors [6] installed and the other platform is powered by Maxon EC40 120 Watt motors [6]. The simulation results are depicted in Figure 3.9.

Again, the platform with two regular wheels accelerates the fastest and the other two platforms still catch up with it. However, it takes more time and the they have traveled greater distances. Especially the platform with three omniwheels catches up with the other platform relatively late. It has to travel almost one third of the length of the field.

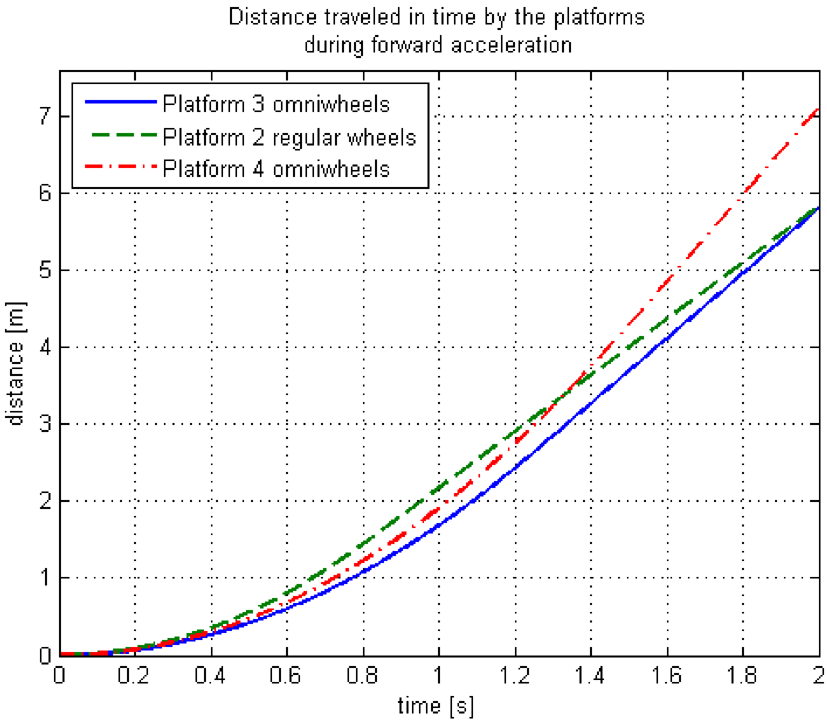


Figure 3.9: Distance traveled in time during forward acceleration by each of the platforms. All platforms use almost the same amount of power.

3.4 Cornering speed

Finally, it is investigated how fast the platforms can follow a random curve with constant radius without lifting a wheel off the ground or slipping out of the corner. These phenomenons are caused by the centrifugal force, which is calculated with Equation 3.21.

$$F_{cen} = \frac{mv^2}{r_{cur}} \tag{3.21}$$

Where v is the forward velocity of the platform and r_{cur} is the radius of the curve that the platform follows. The centrifugal force acts on the C.O.G. similar to the acceleration force in Section 3.3.2. It decreases the normal force acting on the inner wheel(s), until it reaches zero and the wheel(s) lose(s) contact with the field. Figure 3.10 shows the line over which the platforms roll when they lift a wheel off the field.

Both three wheeled platforms have a roll line that is not perpendicular to the centrifugal force. F'_{cen} (see Equation 3.22) is a component of F_{cen} and acts on the C.O.G. perpendicular to the roll line.

$$F'_{cen} = F_{cen}\cos(\beta) \tag{3.22}$$

In the simulation, the platforms follow a curve with a constant radius, while maintaining a constant velocity. Figure 3.11 shows at which velocity and curve radius the inner wheels lose contact with the field. The platform with the four omniwheels loses contact first,

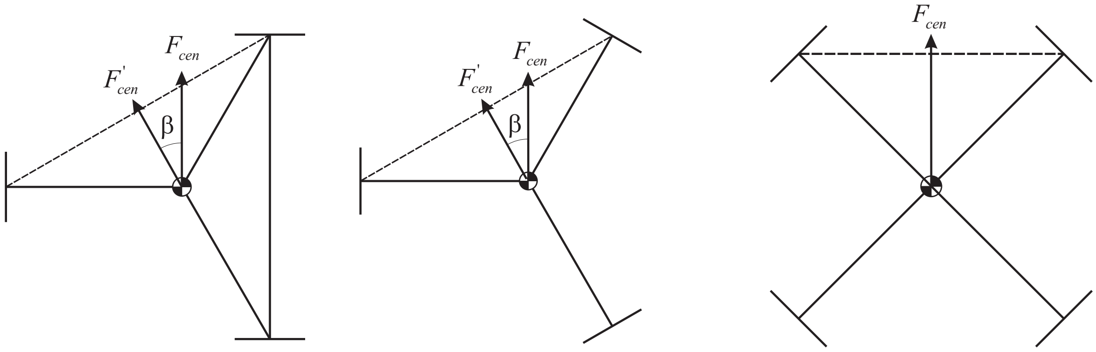


Figure 3.10: Centrifugal force acting on the platforms and the line over which they will roll when they lift the inner wheel(s) off the field.

followed by the platform with three omniwheels and the platform with regular wheels respectably. It has to be investigated whether the wheels can cope with the required lateral force to lift a wheel off the field, or that it starts to slip sideways before the inner wheel is lifted.
The regular wheels slip sideways if the centrifugal force is larger than the friction force produced by the tires in lateral direction. This is calculated with Equation 3.16.
The lateral force an omniwheel can support depends on the angle of the wheel relative to the direction of motion. This is explained with the help of figure 3.12.

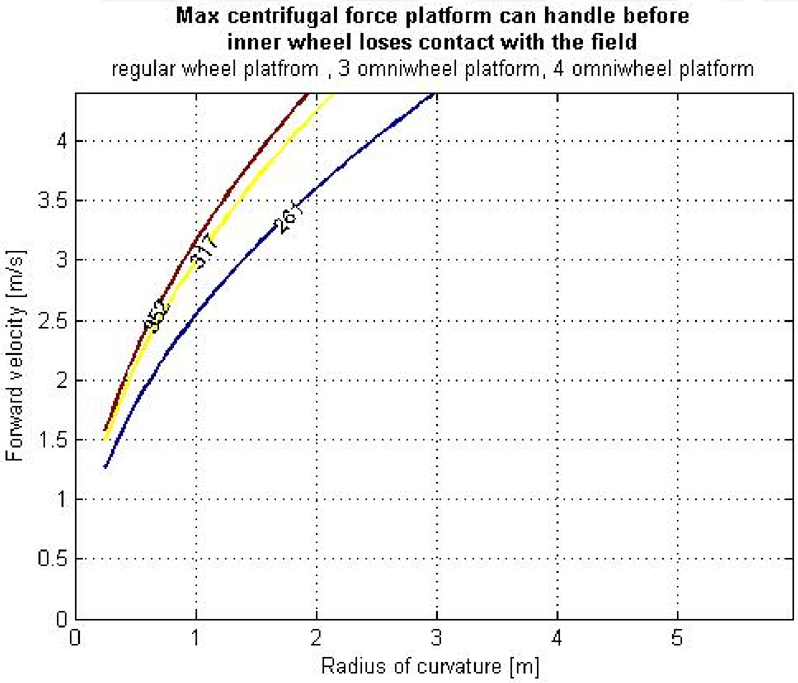


Figure 3.11: The radius of the corner and the velocity of the platform at which a wheel loses contact with the field. Left to right: platform with regular wheels, platform with three omniwheels and platform with four omniwheels.

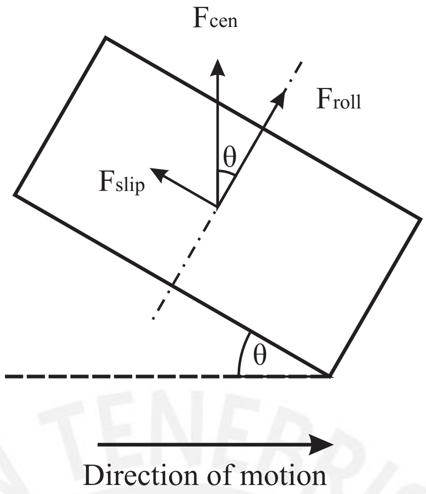


Figure 3.12: The relation between the lateral friction force produced by an omniwheel and the angle of this wheel with respect to the direction of motion.

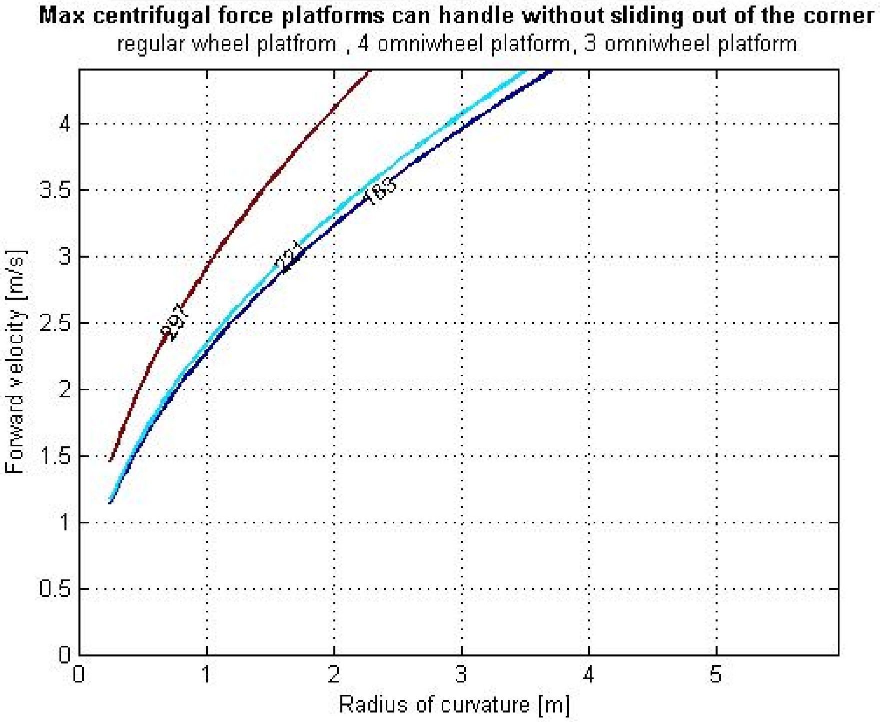


Figure 3.13: The radius of the corner and the velocity of the platform at which the platform starts to slip sideways. Left to right: platform with regular wheels, platform with four omniwheels and platform with three omniwheels.

If F_{cen} and F_{slip} are parallel, Equation 3.16 determines the force required to slip the wheel sideways. If F_{cen} and F_{slip} are not parallel (as depicted in Figure 3.12), the maximum centrifugal force the wheel can handle decrease with a factor $\sin(\theta)$. From θ and the normal load of all wheels, the total lateral force the wheels can produce is calculated.
Due to the shift in load distribution, the inner wheels can handle less lateral force. This is compensated by the outer wheels which can handle more. The robot starts slipping sideways if the centrifugal force is larger than the total lateral force the wheels can handle. Figure 3.13 shows at which forward velocity and at which curvature the robots start to slip sideways.

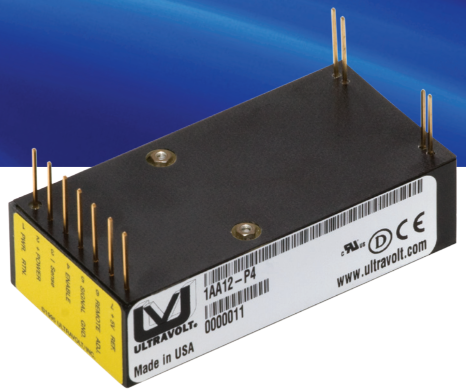
Figure 3.12 and Figure 3.13 show that all the platforms start to slip before a wheel loses contact with the field. They also show that the platform with the regular wheels can perform the tightest corners with the highest velocities, followed by the platform with the four omniwheels. There is not much difference between the platforms with omniwheels. Though the lateral forces (see Figure 3.13) of these platforms differ quite a lot, the platforms start to slip at almost the same velocity and curvature. The extra mass of the platform with four omniwheels causes extra centrifugal force. The extra wheel produces extra friction force to allow faster and tighter curves. This result in a cornering behavior like the platform with three omniwheels.

3.5 Concluding remarks

All the simulations give some insight in the behavior of the platforms in different situations. However, it is not possible to point out the best wheel configuration. There are too many variables left. The platform's performances are heavily depending on the software it runs on. Also, team tactics play an important role in the decision. In the Tech United tactics, maneuverability is important. Tech United is implementing dynamic passing. Therefore, the platform has to be able to make small position adjustments in all directions to position itself well to receive the pass and control the ball. This requirement cannot be fulfilled by the platform with two regular wheels. This leaves the platforms with three and four omniwheels. Besides forward motion, sideways motion is used a lot. The platform with four omniwheels reaches higher end velocities in both directions and can make faster and tighter turns. Therefore it is decided to design a platform with four omniwheels.

AA SERIES

High Voltage Biasing Supply



The AA Series of high-voltage regulated DC-DC converters addresses the needs of the miniature PCB-mount regulated high voltage power supply user. Designed and built utilizing state-of-the-art power-conversion topology, these units feature surface-mount technology and encapsulation techniques that provide high reliability and performance. Typical applications for the AA Series include the following: bias supplies, detectors, piezos, amplifiers, and photomultiplier tubes (PMT).

- 22% smaller than standard A Series
- 8 models from 0 to 62V through 0 to 6kV
- 4, 20 or 30 watts of output power
- Maximum Iout capability down to 0 Volts
- Wide input voltage range
- Indefinite output short-circuit protection
- Output current & voltage monitors
- Fixed-frequency, low-stored-energy design
- >1,250,000 hour MTBF @65°C
- UL/cUL Recognized Component; CE Mark (LVD & RoHS)

PARAMETER	CONDITIONS	MODELS																								UNITS
INPUT		12V												24V												
Voltage Range	Full Power	+ 11 to 16												+ 23 to 30												VDC
Voltage Range	Derated Power Range	+ 9 to 32												+ 9 to 32												VDC
Current	Standby / Disable	< 30												< 30												mA
Current	No Load, Max Eout	< 100												< 90												mA
Current	Max Load, Max Eout	~ 400												~ 1350												mA
AC Ripple Current	Nominal Input, Full Load	< 80												< 80												mA p-p
OUTPUT		1/16AA			1/8AA			1/4AA			1/2AA			1AA			2AA			4AA			6AA			
Voltage Range	Nominal Input	0 to 62			0 to 125			0 to 250			0 to 500			0 to 1,000			0 to 2,000			0 to 4,000			0 to 6,000			VDC
Nominal Input Voltage / Model		12	24	24	12	24	24	12	24	24	12	24	24	12	24	24	12	24	24	12	24	24	12	24	24	VDC
Power	Nominal Input, Max Eout	4	20	30	4	20	30	4	20	30	4	20	30	4	20	30	4	20	30	4	20	30	4	20	30	Watts
Current	Iout Entire Output Voltage Range	64	320	480	32	160	240	16	80	120	8	40	60	4	20	30	2	10	15	1	5	7.5	0.67	3.3	5	mA
Current Scale Factor	Full Load	42.67	969.7	960	11.64	237	258	3.27	70.48	72.7	.79	17.78	17.65	.37	4.60	4.62	.192	1.52	1.52	.090	.752	.76	.066	.490	.50	mA/V
Voltage Monitor Scaling		10:1 ± 2% into 10MΩ												100:1 ± 2% into 10MΩ												-
Ripple	Full Load, Max Eout	0.03	0.06	0.15	0.03	0.038	0.038	0.023	0.04	0.05	0.01	0.01	0.011	0.026	0.048	0.073	0.01	0.011	0.046	0.042	0.050	0.070	0.035	0.024	0.046	%V p-p
Dynamic Load Regulation	½ to Full Load, Max Eout per .1mA	<.12	<.12	<.12	<.12	<.12	<.12	<.20	<.20	<.20	<.50	<.50	<.50	<1.0	<1.0	<1.0	<2.0	<2.0	<2.0	<4.0	<4.0	<4.0	<6.0	<6.0	<6.0	V pk
Line Regulation	Nom. Input, Max Eout, Full Power	< 0.01 %																								VDC
Static Load Regulation	No Load to Full Load, Max Eout	< 0.01%																								VDC
Stability	30 Min. warmup, per 8 hr/ per day	< 0.01% / < 0.02%																								VDC
PROGRAMMING & CONTROLS		ALL TYPES																								
Input Impedance	Nominal Input	+ Output Models 1.1MΩ to GND, - Output Models 1.1MΩ to +5 Vref																								MΩ
Adjust Resistance	Typical Potentiometer Values	10K to 100K (Pot across Vref. & Signal GND, Wiper to Adjust)																								Ω
Adjust Logic	0 to +5 for +Out, +5 to 0 for - Out	+4.64 VDC for +Output or +0.36 for -Output = Nominal Eout																								-
Output Voltage & Impedance	T=+25°C	+ 5.00VDC ± 2%, Zout = 464Ω ± 1%																								-
Enable/Disable		0 to +0.5 Disable, +2.4 to 32 Enable (Default = Enable)																								VDC
ENVIRONMENTAL		ALL TYPES																								
Operating	Full Load, Max Eout, Case Temp.	-40 to +65																								°C
Coefficient	Over the Specified Temperature	±50 (±25 Optional)																								PPM/°C
Thermal Shock	Mil-Std 810, Method 503-4, Proc. II	-40 to +65																								°C
Storage	Non-Operating, Case Temp.	-55 to +105																								°C
Humidity	All Conditions, Standard Package	0 to 95% non-condensing																								-
Altitude	Standard Package, All Conditions	Sea Level through Vacuum																								-
Shock	Mil-Std-810, Method 516.5, Proc. IV	20																								G's
Vibration	Mil-Std-810, Method 514.5, Fig.514.5C-3	10																								G's

Specifications subject to change without notice.

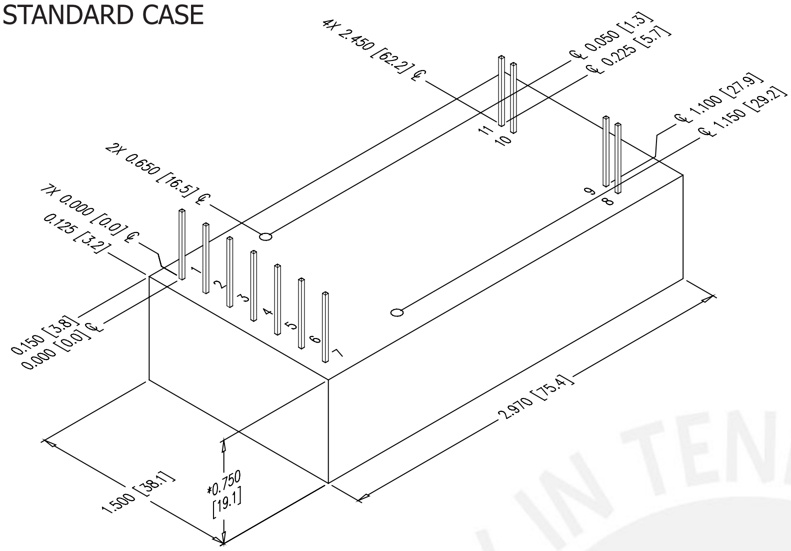
Making High Voltage Easier!®

Higher Service, Higher Performance, Higher Reliability

AA SERIES

High Voltage Biasing Supply

STANDARD CASE



CONSTRUCTION

Epoxy-filled DAP box certified to ASTM-D-5948

SIZE

Volume 3.34in^3 (54.8 cc)
Weight 4.0oz (114g)

TOLERANCE

Overall ±0.050″ (1.27)
Pin to Pin ±0.015″ (0.38)
Mounting hole location ±0.025″ (0.64)

NOTES

20W and 30W versions are an additional 0.062″ (1.57) in height.
-M equipped units are an additional 0.030″ (0.76) for all dimensions.
Contact UltraVolt's Customer Service Department for drawings of models equipped with -E or -H options.

Downloadable drawings (complete with mounting & pin information) and 3D models are available online.

CONNECTIONS

PIN	FUNCTION
1	Input-Power Ground Return
2	Positive Power Input
3	Iout Monitor
4	Enable/Disable
5	Signal Ground Return
6	Remote Adjust Input
7	+5VDC Reference Output
8	HV Ground Return
9	Eout Monitor
10 & 11	HV Output

All grounds joined internally. Power-supply mounting points isolated from internal grounds by >100kΩ, .01uF / 50V (Max) on all models except -M (20W and above), -M-E, and -M-H configurations which are 0Ω.

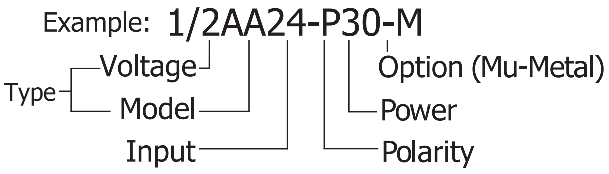
ORDERING INFORMATION

Type	0 to 62 VDC Output	1/16AA
	0 to 125 VDC Output	1/8AA
	0 to 250 VDC Output	1/4AA
	0 to 500 VDC Output	1/2AA
	0 to 1,000 VDC Output	1AA
	0 to 2,000 VDC Output	2AA
	0 to 4,000 VDC Output	4AA
	0 to 6,000 VDC Output	6AA
Input	12VDC Nominal	12
	24VDC Nominal	24
Polarity	Positive Output	-P
	Negative Output	-N
Power	Watts Output (12 V Only)	4
	Watts Output (24 V Only)	20
	Watts Output (24 V Only)	30
Case	Plastic Case - Diallyl Phthalate	(Standard)
	'Eared' Chassis Mounting Plate	-E
Heat Sink	.500″ High (sized to fit case)	-H
Shield	Six-sided Mu-Metal Shield	-M
Temp. Coefficient	25PPM Temperature Coefficient	-25PPM
Enhanced Interface	5V Control and Monitors	-I5
	10V Control and Monitors (24Vin only)	-I10

Note: For more information on the enhanced interface options, download the I5/I10 Option datasheet.

IEC-60950-1

Non-RoHS compliant units are available. Please contact the factory for more information.



Popular accessories ordered with this product include CONN-KIT and BR-18 mounting bracket kit.

Rev. W 2/14

ALUMINUM ELECTROLYTIC CAPACITORS

nichicon

QR series

Screw Terminal Type, 85°C High speed charge-discharge

Smaller | High Ripple Current | Long Life

- Suited for high frequency regenerative voltage for AC servomotor, general inverter.
- Suited for equipment used at valtage fluctuating area.
- Suited for rectifier circuit of voltage doubler
- Compliant to the RoHS directive (2002/95/EC).

QR

Hi speed charge-discharge

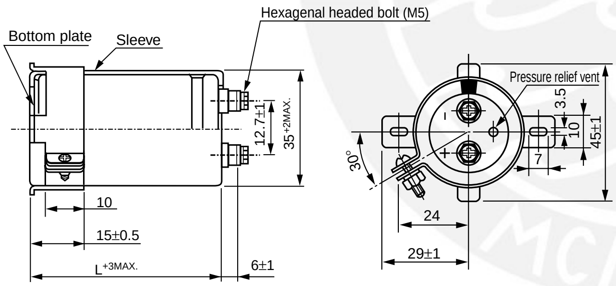
NX

Specifications

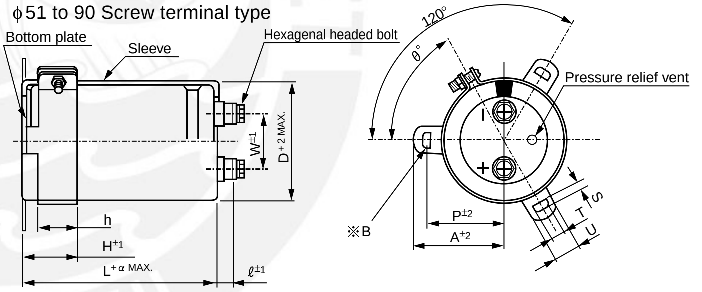
<table>
<tr><td>Item</td><td>Performance Characteristics</td></tr>
<tr><td>Category Temperature Range</td><td>– 25 to +85°C</td></tr>
<tr><td>Rated Voltage Range</td><td>350 to 450V</td></tr>
<tr><td>Rated Capacitance Range</td><td>680 to 15000μF</td></tr>
<tr><td>Capacitance Tolerance</td><td>±20% at 120Hz, 20°C</td></tr>
<tr><td>Leakage Current</td><td>After 5 minutes' application of rated voltage, leakage current is not more than $3\sqrt{CV}$ (μA) or 5 mA, whichever is smaller (at 20°C).
[C : Rated Capacitance(μF), V : Voltage (V)]</td></tr>
<tr><td>Tangent of loss angle (tan δ)</td><td>(Measurement frequency:120Hz, Temperature:20°C)
<table>
<tr><td>Rated voltage (V)</td><td>350</td><td>400</td><td>450</td></tr>
<tr><td>tan δ (MAX.)</td><td>0.15</td><td>0.15</td><td>0.15</td></tr>
</table></td></tr>
<tr><td>Stability at Low Temperature</td><td>
<table>
<tr><td>Rated voltage (V)</td><td>350 to 450</td></tr>
<tr><td rowspan="2">Impedance ratio ZT/Z20(MAX.)</td><td>Z – 25°C / Z+20°C</td></tr>
<tr><td>8</td></tr>
</table>
Measurement frequency : 120Hz</td></tr>
<tr><td>Endurance of charge-discharge behavior</td><td>After an application of charge-discharge voltage for 20million times (charge-discharge voltage difference(ΔV)=rated voltage × 0.3, cycle 3Hz)capacitors shell meet the characteristics requirement listed at right
<table>
<tr><td>Capacitance change</td><td>Within ±20% of the initial capacitance value</td></tr>
<tr><td>tan δ</td><td>300% or less than the initial specified value</td></tr>
<tr><td>Leakage current</td><td>Less than or equal to the initial specified value</td></tr>
<tr><td>Appearance</td><td>There shell be found to remarkable abnormality on the capacitor</td></tr>
</table></td></tr>
<tr><td>Endurance</td><td>The specifications listed at right shall be met when the capacitors are restored to 20°C after D.C. bias plus rated ripple current is applied for 5000 hours at 85°C, the peak voltage shall not exceed the rated voltage.
<table>
<tr><td>Capacitance change</td><td>Within ±20% of the initial capacitance value</td></tr>
<tr><td>tan δ</td><td>300% or less than the initial specified value</td></tr>
<tr><td>Leakage current</td><td>Less than or equal to the initial specified value</td></tr>
</table></td></tr>
<tr><td>Shelf Life</td><td>After storing the capacitors under no load at 85°C for 1000 hours and then performing voltage treatment based on JIS C 5101-4 clause 4.1 at 20°C, they shall meet the specified value for endurance characteristics listed above.</td></tr>
<tr><td>Marking</td><td>Printed with white color letter on darkbrown sleeve.</td></tr>
</table>

Drawing

ϕ35 Screw terminal type

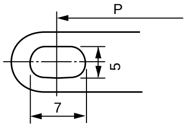


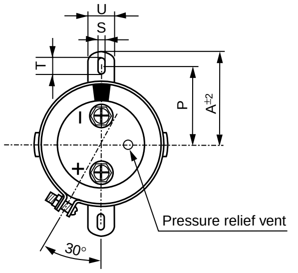
ϕ51 to 90 Screw terminal type



※ B

3-leg brackets for ϕ90 capacitors have different hole shapes from the ordinary ones illustrated below.





Type numbering system (Example : 400V 1800μF)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	Q	R	2	G	1	8	2	M	S	E	F		

- 1: Type
- 2–3: Series name
- 4–5: Rated voltage (400V)
- 6–8: Rated Capacitance (1800μF)
- 9: Capacitance tolerance (±20%)
- 10–11: Configuration※
- 12: Case dia. code
- 13–14: Mounting bracket

ϕD	Code
35	C
51	F
63.5	G
76.2	H
90	J

(ϕ35)

Code less	2-leg brackets
BN	No brackets

(ϕ51 to ϕ90)

Code less	3-leg brackets
BB	2-leg brackets
BN	No brackets

※Configuration

Cr (III) Plating (RoHS compliant)
SE

Resin bushing available upon request.

※Please contact us if PVCless products are required

● Dimension of terminal pitch (W) and length (ℓ) and Nominal dia. of bolt (mm)

ϕD	W	ℓ	α	Nominal dia. of bolt
51	22.0	6	3	M 5
63.5	28.6	6	3	M 5
76.2	31.8	6	3	M 5
90	31.8	6	3	M 5

● Dimension of mounting bracket (mm)

Leg shape / Symbol ϕD	3−Leg				2−Leg			
	51	63.5	76.2	90	51	63.5	76.2	90
P	32.5	38.1	44.5	50.8	33.2	40.5	46.5	53
A	38.5	43	49.2	58.5	40	46.5	53	59
T	7.5	8.0	7.0	8.0	6.0	7.0	6.0	6.0
S	5.0	5.0	5.0	5.0	4.5	4.5	4.5	4.5
U	12	14	14	18	14	14	14	14
θ °	60	60	60	60	30	30	30	30
H	20	25	30	35	25	35	35	35
h	15	20	24	25	15	20	20	20

● Dimension table in next page.

QR series

■Dimensions

350V (2V)					
Cap. (μF)	Size ϕD × L(mm)	Rated ripple (Arms)	tan δ	Leakage Current (mA)	Code
820	35×80	3.3	0.15	1.60	LQR2V821MSEC
1000	35×100	4.3	0.15	1.77	LQR2V102MSEC
1800	51×80	7.2	0.15	2.38	LQR2V182MSEF
2200	51×100	9.1	0.15	2.63	LQR2V222MSEF
2700	51×110	10.8	0.15	2.91	LQR2V272MSEF
	63.5×80	10.6	0.15	2.91	LQR2V272MSEG
3300	51×130	12.4	0.15	3.22	LQR2V332MSEF
	63.5×90	11.9	0.15	3.22	LQR2V332MSEG
3900	63.5×110	14.6	0.15	3.50	LQR2V392MSEG
	76.2×80	14.1	0.15	3.50	LQR2V392MSEH
4700	51×170	17.0	0.15	3.84	LQR2V472MSEF
	76.2×90	16.4	0.15	3.84	LQR2V472MSEH
5600	63.5×150	20.4	0.15	4.20	LQR2V562MSEG
	76.2×110	19.7	0.15	4.20	LQR2V562MSEH
6800	63.5×170	23.5	0.15	4.62	LQR2V682MSEG
	76.2×130	22.9	0.15	4.62	LQR2V682MSEH
	90×100	22.5	0.15	4.62	LQR2V682MSEJ
8200	63.5×190	27.1	0.15	5.00	LQR2V822MSEG
	76.2×150	26.4	0.15	5.00	LQR2V822MSEH
10000	76.2×170	31.1	0.15	5.00	LQR2V103MSEH
	90×130	30.2	0.15	5.00	LQR2V103MSEJ
12000	76.2×190	35.7	0.15	5.00	LQR2V123MSEH
15000	90×190	40.5	0.15	5.00	LQR2V153MSEJ

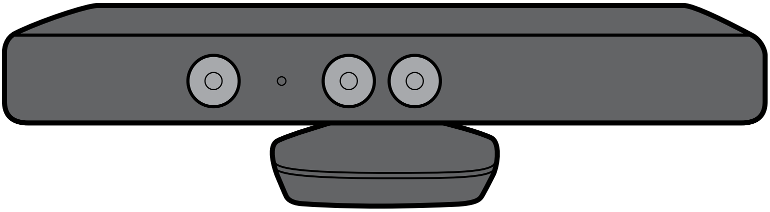
400V (2G)					
Cap. (μF)	Size ϕD × L(mm)	Rated ripple (Arms)	tan δ	Leakage Current (mA)	Code
680	35×80	3.2	0.15	1.56	LQR2G681MSEC
820	35×100	4.1	0.15	1.71	LQR2G821MSEC
1500	51×80	7.5	0.15	2.32	LQR2G152MSEF
1800	51×90	9.1	0.15	2.54	LQR2G182MSEF
2200	51×110	10.4	0.15	2.81	LQR2G222MSEF
2700	63.5×90	11.5	0.15	3.11	LQR2G272MSEG
3300	51×150	13.7	0.15	3.44	LQR2G332MSEF
	63.5×110	13.2	0.15	3.44	LQR2G332MSEG
3900	63.5×130	16.0	0.15	3.74	LQR2G392MSEG
	76.2×90	15.3	0.15	3.74	LQR2G392MSEH
4700	63.5×150	18.7	0.15	4.11	LQR2G472MSEG
	76.2×110	18.3	0.15	4.11	LQR2G472MSEH
5600	63.5×170	22.0	0.15	4.49	LQR2G562MSEG
	76.2×130	21.4	0.15	4.49	LQR2G562MSEH
6800	76.2×150	25.4	0.15	4.94	LQR2G682MSEH
8200	76.2×170	28.6	0.15	5.00	LQR2G822MSEH
	90×130	27.8	0.15	5.00	LQR2G822MSEJ
10000	90×150	32.7	0.15	5.00	LQR2G103MSEJ
12000	90×170	37.6	0.15	5.00	LQR2G123MSEJ
15000	90×220	43.0	0.15	5.00	LQR2G153MSEJ

450V (2W)					
Cap. (μF)	Size ϕD × L(mm)	Rated ripple (Arms)	tan δ	Leakage Current (mA)	Code
680	35×100	3.5	0.15	1.65	LQR2W681MSEC
820	35×110	3.9	0.15	1.82	LQR2W821MSEC
1200	51×80	5.2	0.15	2.20	LQR2W122MSEF
1500	51×100	6.3	0.15	2.46	LQR2W152MSEF
1800	51×110	7.4	0.15	2.70	LQR2W182MSEF
	63.5×80	7.9	0.15	2.70	LQR2W182MSEG
2200	51×130	8.7	0.15	2.98	LQR2W222MSEF
	63.5×100	8.6	0.15	2.98	LQR2W222MSEG
2700	51×150	10.2	0.15	3.30	LQR2W272MSEF
	76.2×80	10.0	0.15	3.30	LQR2W272MSEH
3300	63.5×130	12.4	0.15	3.65	LQR2W332MSEG
	76.2×100	11.8	0.15	3.65	LQR2W332MSEH
3900	63.5×150	13.7	0.15	3.97	LQR2W392MSEG
	76.2×110	14.1	0.15	3.97	LQR2W392MSEH
	90×90	13.6	0.15	3.97	LQR2W392MSEJ
4700	63.5×170	16.5	0.15	4.36	LQR2W472MSEG
	76.2×130	16.3	0.15	4.36	LQR2W472MSEH
	90×110	15.8	0.15	4.36	LQR2W472MSEJ
5600	63.5×190	19.4	0.15	4.76	LQR2W562MSEG
	90×130	19.1	0.15	4.76	LQR2W562MSEJ
6800	76.2×170	23.3	0.15	5.00	LQR2W682MSEH
8200	90×150	26.1	0.15	5.00	LQR2W822MSEJ
10000	90×190	31.3	0.15	5.00	LQR2W103MSEJ
12000	90×220	35.5	0.15	5.00	LQR2W123MSEJ

Rated ripple current (Arms) at 85°C 120Hz

● Frequency coefficient of rated ripple current

Frequency (Hz)	60	120	360	1k	10k or more
Coefficient	0.82	1.00	1.20	1.35	1.40



English

WARNING

Before using this product, read this manual, the Xbox 360® console instructions, and the manuals of any other accessories or games for important safety and health information. Keep all manuals for future reference. For replacement manuals, visit www.xbox.com/support (see "If You Need More Help").

The limited warranty covering this product appears in this manual, which is also available online at www.xbox.com/support.

WARNING

Before allowing children to use the Kinect sensor:

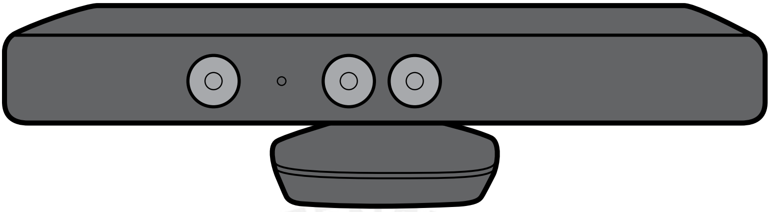
- Determine how each child is able to use the sensor (playing games, chatting or video messaging with other players online) and whether they should be supervised during these activities.
- If you allow children to use the sensor without supervision, be sure to explain all relevant safety and health information and instructions.

Make sure children using the Kinect sensor play safely.

Make sure children using the Kinect sensor play safely and within their limits, and make sure they understand proper use of the system.

This symbol identifies safety and health messages in this manual and Xbox 360 accessories manuals.

XBOX 360 KINECT SENSOR

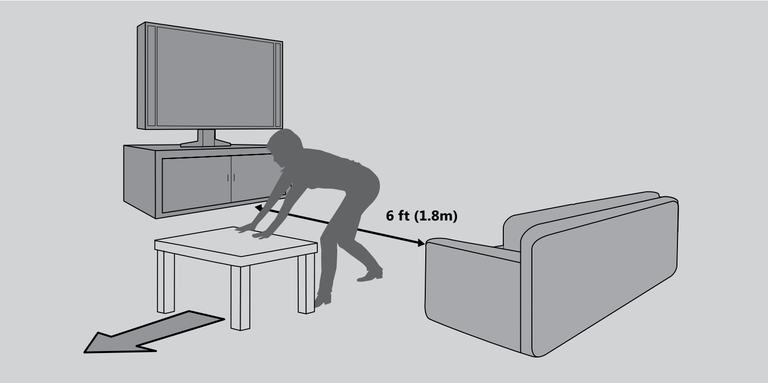


Xbox 360 Kinect Sensor

Thanks for choosing the Xbox 360® Kinect™ Sensor. The Kinect sensor offers a revolutionary new way to play: you're the controller. Just move around and see what happens. Control your Xbox 360 with a wave of your hand. The only experience you need is life experience.

The Kinect sensor is for use with the Xbox 360 video game and entertainment system. To learn more about using the Kinect sensor with a specific game, see the documentation that came with your game.

ADEQUATE SPACE FOR PLAYING



The Kinect sensor needs to be able to see you, and you need room to move. The sensor can see you when you play approximately 6 feet (2 meters) from the sensor. For two people, you should play approximately 8 feet (2.5 meters) from the sensor.

Play space will vary based on your sensor placement and other factors. See your game's instructions for more information about whether it requires only part of the sensor play space.

⚠ **WARNING**

Make sure you have enough space to move freely while playing

Gameplay with your Kinect sensor may require varying amounts of movement. Make sure you won't hit, run into, or trip over other players, bystanders, pets, furniture, or other objects when playing. If you will be standing and/or moving during gameplay, you will also need good footing.

Before playing:

- Look in all directions (right, left, forward, backward, down, and up) for things you might hit or trip over.
- Make sure your play space is far enough away from windows, walls, stairs, etc.
- Make sure there is nothing you might trip on—toys, furniture, or loose rugs, for example. Also, be aware of children and pets in the area. If necessary, move objects or people out of the play space.
- Don't forget to look up. Be aware of light fixtures, fans, and other objects overhead when assessing the play space.

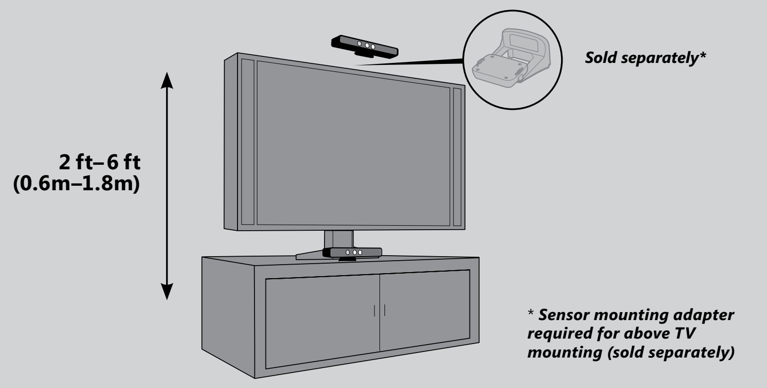
While playing:

- Stay far enough away from the television to avoid contact.
- Keep enough distance from other players, bystanders, and pets. This distance may vary between games, so take account of how you are playing when determining how far away you need to be.
- Stay alert for objects or people you might hit or trip on. People and objects can move into the area during gameplay, so always be alert to your surroundings.

Make sure you always have good footing while playing:

- Play on a level floor with enough traction for game activities.
- Make sure you have appropriate footwear for gaming (no high heels, flip flops, etc.) or are barefoot, if appropriate.

CHOOSE A LOCATION FOR YOUR SENSOR



* **_Sensor mounting adapter required for above TV mounting (sold separately)_**

For the best play space and sensor performance, place your sensor between 2 feet and 6 feet (0.6 and 2 meters) high, the closer to the low or high limit, the better. Also:

- Place the sensor on a stable surface.
- Make sure the sensor is aligned with the center of your TV, and as close as possible to the front edge of the table or shelf.
- Make sure to place the sensor in a location where it will not fall or be struck during gameplay.
- Do not put the sensor on your console.
- Do not place the sensor on or in front of a speaker or a surface that vibrates or makes noise.
- Keep the sensor out of direct sunlight.
- Do not use near any heat sources. Use the sensor within its specified operating temperature range of 41 °F – 95 °F (5 °C – 35 °C). If the sensor is exposed to an environment outside its prescribed range, turn it off and allow the temperature to stabilize within the specified range before using the sensor again.

IMPORTANT

Only adjust the sensor location by moving the base. Do not adjust the sensor viewing angle by hand, by tilting the sensor on its base. After setup is complete, let the sensor motors adjust the viewing angle, or you risk damaging your sensor.

⚠ **WARNING**

Arrange all cables and cords so that people and pets are not likely to trip over or accidentally pull on them as they move around or walk through the area. When the sensor and console are not in use, you may need to disconnect all cables and cords from the sensor and console to keep them out of the reach of children and pets. Do not allow children to play with cables and cords.

⚠ **WARNING**

Avoid Glare

To minimize eyestrain from glare, try the following:

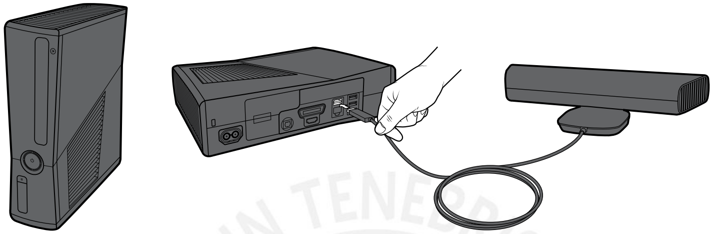
- Position yourself at a comfortable distance from your television or monitor and the Kinect sensor.
- Place your television or monitor and Kinect sensor away from light sources that produce glare, or use window blinds to control light levels.
- Choose soothing natural light that minimizes glare and eyestrain and increases contrast and clarity.
- Adjust your television or monitor brightness and contrast.

SET UP YOUR SENSOR

Before you can use your Kinect sensor, you need to connect it to your Xbox 360 console. For Xbox 360 S consoles, power is supplied by the console. For original Xbox 360 consoles, you'll also need to connect the sensor to a standard wall outlet.

Connect the Sensor to Your Xbox 360 S Console

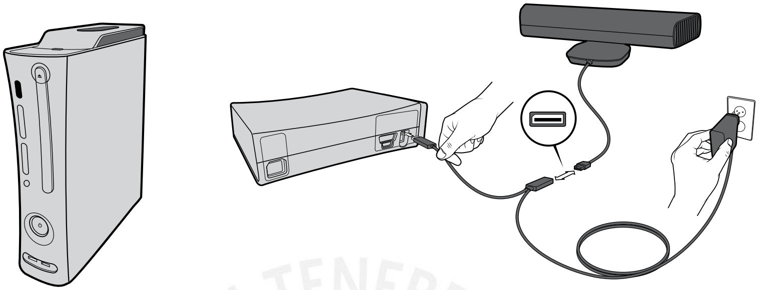
To connect to your Xbox 360 S console, simply plug the sensor into the console AUX port.



Xbox 360 S

Connect the Sensor to Your Original Xbox 360 Console

The sensor only works with the back USB port on an original Xbox 360 console.



Original Xbox 360

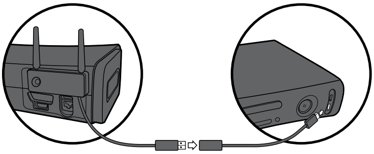
To connect to your original Xbox 360 console:

1. Unplug any accessories from the back USB port on your console.
2. Plug the sensor into the USB/power cable.
3. Plug the USB/power cable into your console's back USB port.
4. Plug the AC adapter end of the USB/power cable into a wall outlet.

Use only the USB/power cable that is shipped with the product or is given to you by an authorized repair center.

If you have an original Xbox 360 console with no hard drive, you should also attach a storage device with at least 256 MB free space. You can use an Xbox 360 Hard Drive, Xbox 360 Memory Unit, or a USB flash drive.

If you're using an Xbox 360 Wireless Networking Adapter that's already connected to the back USB port, you'll need to disconnect its USB cable and reconnect it to a front USB port using the WiFi extension cable, provided.



To reconnect a wireless networking adapter to a front USB port, if necessary:

1. Unplug the wireless networking adapter cable from the USB port on the back of the console, leaving the adapter itself attached to the console.
2. Attach the WiFi extension cable to the wireless networking adapter's USB cable.
3. Plug the other end of the extension cable into a USB port on the front of your console.

⚠ Electrical Safety

As with many other electrical devices, failure to take the following precautions can result in serious injury or death from electric shock or fire or damage to the sensor.

If you use AC power, select an appropriate power source:

- The sensor's power input is 12V DC @ 1.1A. Use only the AC adapter on the USB/power cable that came with your sensor or that you received from an authorized repair center.
- Confirm that your electrical outlet provides the type of power indicated on the USB/power cable, in terms of voltage (V) and frequency (Hz). If you aren't sure of the type of power supplied to your home, consult a qualified electrician.
- Do not use non-standard power sources, such as generators or inverters, even if the voltage and frequency appear acceptable. Use only AC power provided by a standard wall outlet.
- Do not overload your wall outlet, extension cord, power strip, or other electrical receptacle. Confirm that they are rated to handle the total current (in amps [A]) drawn by the sensor (indicated on the USB/power cable) and any other devices that are on the same circuit.

To avoid damaging the USB/power cable:

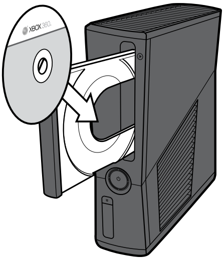
- Protect the cable from being pinched or sharply bent, particularly where it connects to the power outlet and the sensor.
- Protect the cable from being walked on.
- Do not jerk, knot, sharply bend, or otherwise abuse the cable.
- Do not expose the cable to sources of heat.
- Keep children and pets away from the cable. Do not allow them to bite or chew on the cable.
- When disconnecting the cable, pull on the plug—do not pull on the cable.

If the USB/power cable becomes damaged in any way, stop using it immediately and contact Xbox Customer Support for a replacement.

Unplug your sensor's USB/power cable during lightning storms or when unused for long periods of time.

Install the Sensor Software on Your Console

Your console needs a system update before you can use it with your Kinect sensor.



To update your console:

1. Turn on your console and insert the supplied disc. The update will install automatically. If it doesn't start automatically, select the disc tray from the dashboard (as if you were playing a game from a disc).
2. When the installation confirmation message appears, remove the disc and begin setting up your sensor.

⚠ WARNING

Don't overexert yourself

Gameplay with the Kinect sensor may require varying amounts of physical activity.

Consult a doctor before using the sensor if you have any medical condition or issue that affects your ability to safely perform physical activities, or if:

- you are or may be pregnant,
- you have heart, respiratory, back, joint, or other orthopedic conditions,
- you have high blood pressure,
- you have difficulty with physical exercise, or
- you have been instructed to restrict physical activity.

Consult your doctor before beginning any exercise routine or fitness regimen that includes using your sensor.

Do not play under the influence of drugs or alcohol, and make sure your balance and physical abilities are sufficient for any movements while gaming.

Take breaks periodically

- Stop and rest if your muscles, joints, or eyes become tired or sore.
- If you experience excessive fatigue, nausea, shortness of breath, chest tightness, dizziness, discomfort, or pain, STOP USING IMMEDIATELY and consult a doctor.

Adults — attend to children

Make sure children using your sensor play within their limits.

Do not use unlicensed accessories or unauthorized props or other objects with the Kinect sensor

Use of these accessories or objects may result in injury to yourself or others and/or in damage to the sensor or other property.

CLEAN YOUR SENSOR

If you clean the sensor:

- Clean the outside of the sensor only.
- Use a dry cloth—do not use abrasive pads, detergents, scouring powders, solvents (for example, alcohol, gasoline, paint thinner, or benzene), or other liquid or aerosol cleaners.
- Do not use compressed air.
- Do not attempt to clean connectors.
- Clean the surface on which the sensor rests with a dry cloth.
- Do not allow the sensor to become wet. To reduce the risk of fire or shock, do not expose the sensor to rain or other types of moisture.

TROUBLESHOOTING

If you encounter problems, try the possible solutions provided below.

Sensor Doesn’t Work

- Make sure cables are connected. When used with an original Xbox 360 console, make sure the sensor is connected to the back USB port, and the USB/power cable is plugged in. The light on the front of the sensor will light up when the sensor is on.
- Make sure the sensor is in a well-ventilated area.
- If the sensor software hasn’t been set up, insert the sensor software disc for a system update.

Sensor Doesn’t See Player

- Play in the sensor’s play space.
- Turn on lights to brighten the play space.
- Prevent lights, including sunlight, from shining directly on the sensor.
- Try wearing different clothing that contrasts with the background of your play space.
- Clean the sensor lens with a dry cloth.
- Make sure nothing is blocking the sensor’s viewing angle.

Sensor Loses Player

Try leaving and reentering the play space if the sensor has stopped tracking you.

Sensor Doesn’t Hear Player

- Don’t place the sensor near sources of vibration, TV speakers, or other audio sources.
- Make sure nothing is blocking the sensor’s microphone array.

Sensor Motors Don’t Adjust Sensor Viewing Angle

- Make sure cables are connected and the sensor light is on. When used with an original Xbox 360 console, the sensor must be connected to the back USB port, not a front USB port, and the USB/power cable must be connected to a power outlet.
- Make sure the sensor’s motion isn’t blocked by anything.
- Don’t tilt your sensor on its base or adjust the sensor viewing angle by hand. Let the sensor motors adjust the viewing angle automatically, or you risk damaging your sensor.

IF YOU NEED MORE HELP

For answers to common questions, troubleshooting steps, and Xbox Customer Support contact information, visit www.xbox.com/support.

Do not take your Xbox 360 console or its accessories to your retailer for repair. Please visit www.xbox.com/support for troubleshooting and service information.

⚠ **WARNING**

Do Not Attempt Repairs

Do not attempt to take apart, open, service, or modify the hardware device or power supply. Doing so could present the risk of electric shock or other hazard. Any evidence of any attempt to open and/or modify the device, including any peeling, puncturing, or removal of any of the labels, will void the Limited Warranty.

LIMITED WARRANTY

BY USING YOUR KINECT SENSOR YOU AGREE TO THIS WARRANTY. BEFORE SETTING IT UP, PLEASE READ THIS WARRANTY CAREFULLY. IF YOU DO NOT ACCEPT THIS WARRANTY, DO NOT USE YOUR KINECT SENSOR. RETURN IT UNUSED TO YOUR RETAILER OR MICROSOFT FOR A REFUND. Contact Microsoft at http://www.xbox.com/ (800) 4MY-XBOX or (800) 469-9269. This warranty gives You specific legal rights. You may also have other rights which vary from State to State or Province to Province.

1. Definitions

(a) "Kinect Sensor" means a new Kinect Sensor purchased from an authorized retailer.

(b) "Warranty Period" means 1 year from the date You purchased Your Kinect Sensor.

(c) "You" means the original end-user.

(d) "Normal Use Conditions" means ordinary consumer use under normal home conditions according to the instruction manual for the Kinect Sensor.

(e) "State" means a State, the District of Columbia, and any other United States territory or possession. "The United States of America" includes all of them.

2. Warranty

(a) During the Warranty Period, Microsoft warrants, only to You, that the Kinect Sensor will not malfunction under Normal Use Conditions.

(b) This is the only warranty Microsoft gives for Your Kinect Sensor and Microsoft gives no other guarantee, warranty, or condition. No one else may give any guarantee, warranty, or condition on Microsoft's behalf.

(c) IF YOUR STATE'S OR PROVINCE'S LAW GIVES YOU ANY IMPLIED WARRANTY, INCLUDING AN IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ITS DURATION IS LIMITED TO THE WARRANTY PERIOD. Some States or Provinces do not allow limitations on how long an implied warranty lasts, so this limitation may not apply to You.

3. How to Get Warranty Service

(a) Before starting the warranty process, please use the trouble-shooting tips at http://www.xbox.com/en-US (United States) or http://www.xbox.com/en-CA (Canada).

(b) If the troubleshooting tips don't resolve Your problem, then follow the online process at http://www.xbox.com/en-US (United States) or http://www.xbox.com/en-CA (Canada). If You don't have access to the Internet, You can call (800) 4MY-XBOX or (800) 469-9269.

4. Microsoft's Responsibility

(a) After You return Your Kinect Sensor Microsoft will inspect it.

(b) If Microsoft determines that the Kinect Sensor malfunctioned during the Warranty Period under Normal Use Conditions, Microsoft will (at its option) repair or replace it, or refund the purchase price to You. Repair may use new or refurbished parts. Replacement may be with a new or refurbished unit.

(c) After repair or replacement, Your Kinect Sensor will be covered by this warranty for the longer of the remainder of Your original Warranty Period, or 95 days after Microsoft ships it to You.

(d) MICROSOFT'S RESPONSIBILITY TO REPAIR OR REPLACE YOUR KINECT SENSOR, OR TO REFUND THE PURCHASE PRICE, IS YOUR EXCLUSIVE REMEDY.

(e) If Your Kinect Sensor malfunctions after the Warranty Period expires, there is no warranty of any kind. After the Warranty Period expires, Microsoft may charge You a fee for its efforts to diagnose and service any problems with Your Kinect Sensor.

5. Warranty Exclusions

Microsoft is not responsible and this warranty does not apply if Your Kinect Sensor is:

(a) damaged by use with products not sold or licensed by Microsoft (including, for example, games and accessories not manufactured or licensed by Microsoft, and "pirated" games, etc.);

(b) used for commercial purposes (including, for example, rental, pay-per-play, etc.);

(c) opened, modified, or tampered with (including, for example, any attempt to defeat any Kinect Sensor technical limitation, security, or anti-piracy mechanism, etc.), or its serial number is altered or removed;

(d) damaged by any external cause (including, for example, by being dropped, used with inadequate ventilation, etc., or failure to follow instructions in the instruction manual for the Kinect Sensor); or

(e) repaired by anyone other than Microsoft.

6. EXCLUSION OF CERTAIN DAMAGES

MICROSOFT IS NOT RESPONSIBLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES; ANY LOSS OF DATA, PRIVACY, CONFIDENTIALITY, OR PROFITS; OR ANY INABILITY TO USE YOUR KINECT SENSOR. THESE EXCLUSIONS APPLY EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF THESE DAMAGES, AND EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. Some States or Provinces do not allow the exclusion or limitation of incidental or consequential damages, so this limitation or exclusion may not apply to You.

7. Additional Terms

If You attempt to defeat or circumvent any Kinect Sensor technical limitation, security, or anti-piracy system, You may cause Your Kinect Sensor to stop working permanently. You will also void Your warranty, and make Your Kinect Sensor ineligible for authorized repair, even for a fee.

8. Choice of Law

(a) If You acquired Your Kinect Sensor in the United States, Washington State law governs the interpretation of this warranty and any claim that Microsoft has breached it, regardless of conflict of law principles.

(b) If You acquired Your Kinect Sensor in Canada, Ontario Provincial law governs the interpretation of this warranty and any claim that Microsoft has breached it, regardless of conflict of law principles.

(c) The laws of the State or Province where You live govern all other claims (including consumer protection, unfair competition, implied warranty, and tort claims).

All parts of this Limited Warranty apply to the maximum extent permitted by law or unless prohibited by law

9. This warranty is valid only in the United States of America and Canada.

Microsoft's address in the United States: Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

Microsoft's address in Canada: Microsoft Canada Inc., 1950 Meadowvale Blvd., Mississauga, Ontario, L5N 8L9

SOFTWARE LICENSE

BY USING YOUR KINECT SENSOR YOU AGREE TO THIS SOFTWARE LICENSE. BEFORE SETTING IT UP, PLEASE READ THIS SOFTWARE LICENSE CAREFULLY. IF YOU DO NOT ACCEPT THIS SOFTWARE LICENSE, DO NOT USE YOUR KINECT SENSOR. RETURN IT TO UNUSED TO YOUR RETAILER OR MICROSOFT FOR A REFUND. Contact Microsoft at http://www.xbox.com/ (800) 4MY-XBOX or (800) 469-9269.

1. Definitions

(a) "Xbox 360 S" means an Xbox 360 S console.

(b) "Authorized Accessory" means a Microsoft branded Xbox 360 or Xbox 360 S hardware accessory, and a Microsoft licensed, third party branded, Xbox 360 or Xbox 360 S hardware accessory whose packaging bears the official "Licensed for Xbox" logo. The Kinect Sensor is an Authorized Accessory solely for purpose of this software license.

(c) "Authorized Games" means Xbox 360 or Xbox 360 S games on game discs published or licensed by Microsoft, and game content downloaded from Microsoft's Xbox LIVE service or Xbox.com Web site (for example, avatars, downloadable games, game add-ons, etc.).

(e) "Software" means the software pre-installed in the Kinect Sensor, including any updates Microsoft may make available from time to time.

(e) "Unauthorized Accessories" means all hardware accessories other than an Authorized Accessory, except that USB memory sticks, digital cameras used to make photographs or movies, and music players used to play music or display photographs or videos are not Unauthorized Accessories.

(f) "Unauthorized Games" means all game discs, game downloads, and game content or media other than Authorized Games.

(g) "Unauthorized Software" means any software not distributed by Microsoft through Xbox 360 or Xbox 360 S game discs published or licensed by Microsoft, Microsoft's Xbox LIVE service, or Xbox.com Web site.

(h) "You" means the user of a Kinect Sensor.

2. License

(a) The Software is licensed to You, not sold. You are licensed to use the Software only as pre-installed in Your Kinect Sensor, and updated by Microsoft from time to time. You may not copy or reverse engineer the Software.

(b) As conditions to this Software license, You agree that:

  (i) **You will use Your Kinect Sensor with Xbox 360 or Xbox 360 S only and not with any other device (including, for example, personal computers, other video game consoles, etc.). You will use only Authorized Games with Your Kinect Sensor. You will not use Unauthorized Accessories or Unauthorized Games. They may not work or may stop working permanently after a Software update.**

  (ii) **You will not use or install any Unauthorized Software. If You do, Your Kinect Sensor may stop working permanently at that time or after a later Software update.**

  (iii) You will not attempt to defeat or circumvent any Kinect Sensor technical limitation, security, or anti-piracy system. If You do, Your Kinect Sensor may stop working permanently at that time or after a later Software update.

  (iv) Microsoft may use technical measures, including Software updates, to limit use of the Kinect Sensor to Xbox 360 or Xbox 360 S, to prevent use of Unauthorized

Accessories and Unauthorized Games, and to protect the technical limitations, security and anti-piracy systems in the Kinect Sensor.

(v) **Microsoft may update the Software from time to time without further notice to You, for example, to update any technical limitation, security, or anti-piracy system**.

3. Warranty.

The Software is covered by the Limited Warranty for Your Kinect Sensor, and Microsoft gives no other guarantee, warranty, or condition for the Software. No one else may give any guarantee, warranty, or condition on Microsoft's behalf.

4. EXCLUSION OF CERTAIN DAMAGES

MICROSOFT IS NOT RESPONSIBLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES; ANY LOSS OF DATA, PRIVACY, CONFIDENTIALITY, OR PROFITS; OR ANY INABILITY TO USE THE SOFTWARE. THESE EXCLUSIONS APPLY EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF THESE DAMAGES, AND EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. Some States or Provinces do not allow the exclusion or limitation of incidental or consequential damages, so this limitation or exclusion may not apply to You.

5. Choice of Law

(a) If You acquired Your Kinect Sensor in the United States, Washington State law governs the interpretation of this Software license and any claim that Microsoft has breached it, regardless of conflict of law principles.

(b) If You acquired Your Kinect Sensor in Canada, Ontario Provincial law governs the interpretation of this Software license and any claim that Microsoft has breached it, regardless of conflict of law principles.

(c) The laws of the State or Province where You live govern all other claims (including consumer protection, unfair competition, implied warranty, and tort claims).

This agreement applies to the maximum extent permitted by law and unless prohibited by law. This agreement does not change your rights under the laws of your State or country if the laws of your State or country do not permit it to do so.

Microsoft's address in the United States: Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

Microsoft's address in Canada: Microsoft Canada Inc., 1950 Meadowvale Blvd., Mississauga, Ontario, L5N 8L9

REGULATIONS

Disposal of Waste Batteries and Electrical & Electronic Equipment

This symbol on the product or its batteries or its packaging means that this product and any batteries it contains must not be disposed of with your household waste. Instead, it is your responsibility to hand this over to an applicable collection point for the recycling of batteries and electrical and electronic equipment. This separate collection and recycling will help to conserve natural resources and prevent potential negative consequences for human health and the environment due to the possible presence of hazardous substances in batteries and electrical and electronic equipment, which could be caused by inappropriate disposal. For more information about where to drop off your batteries and electrical and electronic waste, please contact your local city/municipality office, your household waste disposal service, or the shop where you purchased this product.

This product is for use with NRTL-listed (UL, CSA, ETL, etc.) and/or IEC/EN 60950 compliant (CE marked) Information Technology equipment.

Laser Specifications

⚠ **CAUTION**

Use of controls or adjustments, or performance of procedures other than those specified herein may result in hazardous radiation exposure.

This device complies with International Standard IEC 60825-1:2007:03 for a Class 1 laser product. This device also complies with 21 CFR 1040.10 and 1040.11 except for deviations pursuant to Laser Notice No. 50, dated June 24, 2007.

The following Class 1 laser label is located on the foot of the sensor.

CLASS 1 LASER PRODUCT

For Customers in the United States

FCC Declaration of Conformity (DoC):

Trade Name:	Microsoft Corp.
Models:	1414
Responsible Party:	Microsoft Corporation
Address:	One Microsoft Way, Redmond, WA 98052 U.S.A.
Telephone No.:	(800) 4MY-XBOX

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy.

If not installed and used in strict accordance with the instructions given in the printed documentation and/or on-screen help files, the device may cause harmful interference with other radio-communications devices (AM/FM radios, televisions, baby monitors, cordless phones, etc.). There is, however, no guarantee that RF interference will not occur in a particular installation.

To determine if your hardware device is causing interference to other radio-communications devices, disconnect the device from your computer or remove the device's batteries (for a battery operated device). If the interference stops, it was probably caused by the device. If the interference continues after you disconnect the device or remove the batteries, turn the device off and then on again. If the interference stopped when the device was off, check to see if one of the input/output (I/O) devices is causing the problem. Disconnect the I/O devices one at a time and see if the interference stops.

If this hardware device does cause interference, try the following measures to correct it:

- Relocate the antenna of the other radio-communications device (AM/FM radio, television, baby monitor, cordless phone, etc.) until the interference stops.
- Move the hardware device farther away from the radio or TV, or move it to one side or the other of the radio or TV.
- Plug the device into a different power outlet so that the device and radio or TV are on different circuits controlled by different circuit breakers or fuses.
- If necessary, consult the dealer or an experienced radio/TV technician for more suggestions.

⚠ **CAUTION**

Any changes or modifications made on the system not expressly approved by the manufacturer could void the user's authority to operate the equipment.

For Customers in Canada

This Class B digital apparatus complies with Canadian ICES-003.

Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

COPYRIGHT

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved.

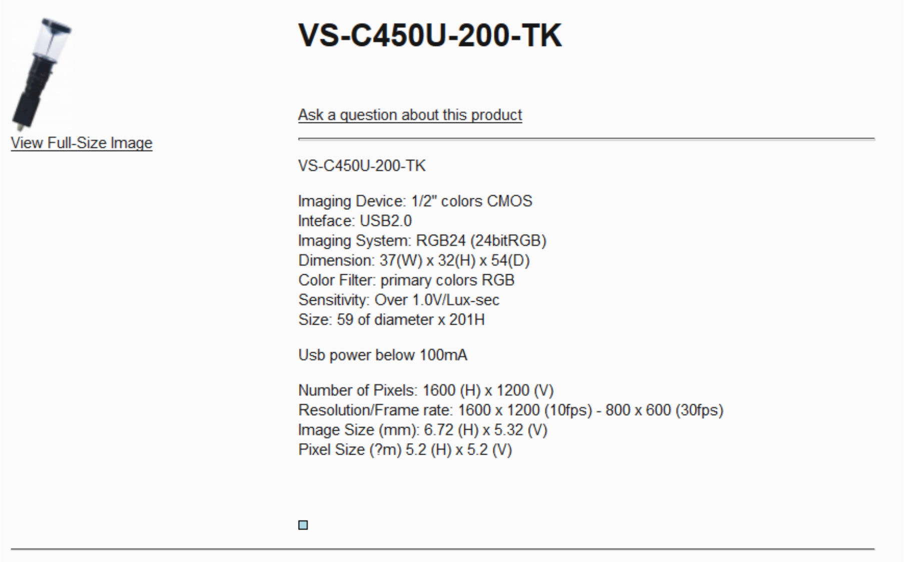
Microsoft, Xbox, Xbox 360, Xbox LIVE, the Xbox logos, the Xbox LIVE logo, and Kinect are trademarks of the Microsoft group of companies.

CUSTOMER SUPPORT

For answers to common questions, troubleshooting steps, and Xbox Customer Support contact information, visit **www.xbox.com/support**.

Datasheet

Cámara Omnidireccional



View Full-Size Image

VS-C450U-200-TK

Ask a question about this product

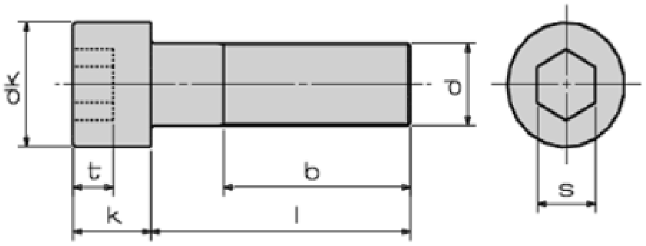
VS-C450U-200-TK

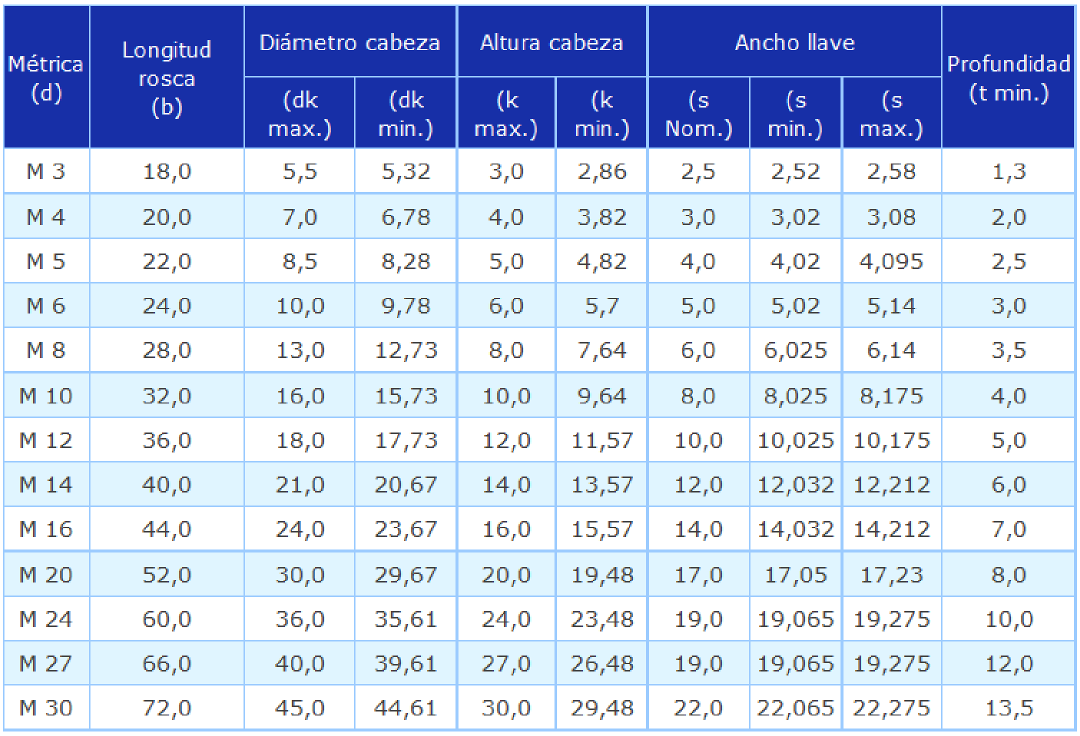
Imaging Device: 1/2" colors CMOS
Inteface: USB2.0
Imaging System: RGB24 (24bitRGB)
Dimension: 37(W) x 32(H) x 54(D)
Color Filter: primary colors RGB
Sensitivity: Over 1.0V/Lux-sec
Size: 59 of diameter x 201H

Usb power below 100mA

Number of Pixels: 1600 (H) x 1200 (V)
Resolution/Frame rate: 1600 x 1200 (10fps) - 800 x 600 (30fps)
Image Size (mm): 6.72 (H) x 5.32 (V)
Pixel Size (?m) 5.2 (H) x 5.2 (V)

Dimensiones Tornillo Allen DIN 912





Métrica (d)	Longitud rosca (b)	Diámetro cabeza		Altura cabeza		Ancho llave			Profundidad (t min.)
		(dk max.)	(dk min.)	(k max.)	(k min.)	(s Nom.)	(s min.)	(s max.)	
M 3	18,0	5,5	5,32	3,0	2,86	2,5	2,52	2,58	1,3
M 4	20,0	7,0	6,78	4,0	3,82	3,0	3,02	3,08	2,0
M 5	22,0	8,5	8,28	5,0	4,82	4,0	4,02	4,095	2,5
M 6	24,0	10,0	9,78	6,0	5,7	5,0	5,02	5,14	3,0
M 8	28,0	13,0	12,73	8,0	7,64	6,0	6,025	6,14	3,5
M 10	32,0	16,0	15,73	10,0	9,64	8,0	8,025	8,175	4,0
M 12	36,0	18,0	17,73	12,0	11,57	10,0	10,025	10,175	5,0
M 14	40,0	21,0	20,67	14,0	13,57	12,0	12,032	12,212	6,0
M 16	44,0	24,0	23,67	16,0	15,57	14,0	14,032	14,212	7,0
M 20	52,0	30,0	29,67	20,0	19,48	17,0	17,05	17,23	8,0
M 24	60,0	36,0	35,61	24,0	23,48	19,0	19,065	19,275	10,0
M 27	66,0	40,0	39,61	27,0	26,48	19,0	19,065	19,275	12,0
M 30	72,0	45,0	44,61	30,0	29,48	22,0	22,065	22,275	13,5

Cortex™-A7 MPCore™

Revision: r0p3

Technical Reference Manual

ARM®

ARM DDI 0464D (ID052812)

Cortex-A7 MPCore

Technical Reference Manual

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
03 October 2011	A	Non-Confidential	First release for r0p0
09 November 2011	B	Non-Confidential	First release for r0p1
11 January 2012	C	Non-Confidential	First release for r0p2
15 May 2012	D	Non-Confidential	First release for r0p3

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM® in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means "ARM or any of its subsidiaries as appropriate".

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Contents

Cortex-A7 MPCore Technical Reference Manual

Preface

Chapter 1 Introduction

Chapter 2 Functional Description

Chapter 3 Programmers Model

Appendix A Signal Descriptions

Appendix B Revisions

Preface

This preface introduces the *Cortex-A7 MPCore Technical Reference Manual*. It contains the following sections:

- *About this book* on page vii
- *Feedback* on page xi.

About this book

This book is for the Cortex-A7 MPCore processor. This is a multiprocessor device that has between one to four processors.

Product revision status

The r*n*p*n* identifier indicates the revision status of the product described in this book, where:

r*n*	Identifies the major revision of the product.
p*n*	Identifies the minor revision or modification status of the product.

Intended audience

This book is written for hardware and software engineers implementing the Cortex-A7 MPCore processor in system designs. It provides information that enables designers to integrate the Cortex-A7 MPCore processor into a target system.

Using this book

This book is organized into the following chapters:

Chapter 1 *Introduction*

Read this for an introduction to the Cortex-A7 MPCore processor and descriptions of the major features.

Chapter 2 *Functional Description*

Read this for a description of the functionality of the Cortex-A7 MPCore processor.

Chapter 3 *Programmers Model*

Read this for a description of the programmers model.

Chapter 4 *System Control*

Read this for a description of the system control registers, their structure, operation, and how to use them.

Chapter 5 *Memory Management Unit*

Read this for a description of the *Memory Management Unit* (MMU) and the address translation process.

Chapter 6 *L1 Memory System*

Read this for a description of the *Level 1* (L1) memory system, including caches, *Translation Lookaside Buffers* (TLB), and store buffer.

Chapter 7 *L2 Memory System*

Read this for a description of the *Level 2* (L2) memory system, including the *Snoop Control Unit* (SCU) and the AXI Coherency Extensions (ACE) attributes.

Chapter 8 *Generic Interrupt Controller*

Read this for a description of the *Generic Interrupt Controller* (GIC).

Chapter 9 *Generic Timer*

Read this for a description of the timers.

Chapter 10 *Debug*

Read this for a description of the support for debug.

Chapter 11 *Performance Monitoring Unit*

Read this for a description of the *Performance Monitoring Unit* (PMU) and associated events.

Appendix A *Signal Descriptions*

Read this for a description of the input and output signals.

Appendix B *Revisions*

Read this for a description of the technical changes between released issues of this book.

Glossary

The *ARM glossary* is a list of terms used in ARM documentation, together with definitions for those terms. The *ARM glossary* does not contain terms that are industry standard unless the ARM meaning differs from the generally accepted meaning.

See *ARM Glossary*, `http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014-/index.html`.

Conventions

Conventions that this book can use are described in:

- *Typographical*
- *Timing diagrams* on page ix
- *Signals* on page ix.

Typographical

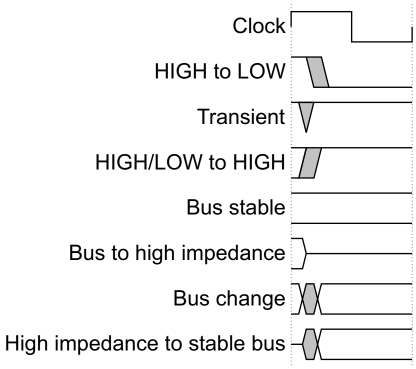
The typographical conventions are:

italic	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`monospace italic`	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.
< and >	Enclose replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the *ARM glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

The figure named *Key to timing diagram conventions* explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.



Key to timing diagram conventions

Timing diagrams sometimes show single-bit signals as HIGH and LOW at the same time and they look similar to the bus change shown in *Key to timing diagram conventions*. If a timing diagram shows a single-bit signal in this way then its value does not affect the accompanying description.

Signals

The signal conventions are:

Signal level	The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means: • HIGH for active-HIGH signals • LOW for active-LOW signals.
Lower-case n	At the start or end of a signal name denotes an active-LOW signal.

Additional reading

This section lists publications by ARM and by third parties.

See Infocenter, `http://infocenter.arm.com`, for access to ARM documentation.

ARM publications

This book contains information that is specific to this product. See the following documents for other relevant information:

- *AMBA® APB Protocol Specification* (ARM IHI 0024)
- *AMBA AXI and ACE Protocol Specification, AXI3, AXI4, and AXI4-Lite, ACE and ACE-Lite* (ARM IHI 0022)
- *ARM® Architecture Reference Manual ARMv7-A and ARMv7-R edition* (ARM DDI 0406)
- *ARM Generic Interrupt Controller Architecture Specification* (ARM IHI 0048)

- *CoreSight™ Architecture Specification* (ARM IHI 0029)
- *CoreSight ETM-A7 Technical Reference Manual* (ARM DDI 0468)
- *CoreSight ETM-A7 Configuration and Sign-off Guide* (ARM DII 0261)
- *CoreSight SoC Technical Reference Manual* (ARM DDI 0480)
- *Embedded Trace Macrocell Architecture Specification* (ARM IHI 0014)
- *Cortex-A Series Programmer's Guide* (ARM DEN0013B)
- *Cortex-A7 MPCore Configuration and Sign-off Guide* (ARM DII 0256)
- *Cortex-A7 MPCore Floating-Point Unit Technical Reference Manual* (ARM DDI 0463)
- *Cortex-A7 MPCore Integration Manual* (ARM DIT 0017)
- *Cortex-A7 MPCore NEON Media Processing Engine Technical Reference Manual* (ARM DDI 0462).

Other publications

This section lists relevant documents published by third parties:

- *ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic*
- *ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic.*

Feedback

ARM welcomes feedback on this product and its documentation.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to `errata@arm.com`. Give:

- the title
- the number, ARM DDI 0464D
- the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Chapter 1
Introduction

This chapter introduces the Cortex-A7 MPCore processor and its features. It contains the following sections:

1.1 About the Cortex-A7 MPCore processor

The Cortex-A7 MPCore processor is a high-performance, low-power processor that implements the ARMv7-A architecture. The Cortex-A7 MPCore processor has one to four processors in a single multiprocessor device with a L1 cache subsystem, an optional integrated GIC, and an optional L2 cache controller.

Figure 1-1 shows an example of a Cortex-A7 MPCore configuration with four processors.

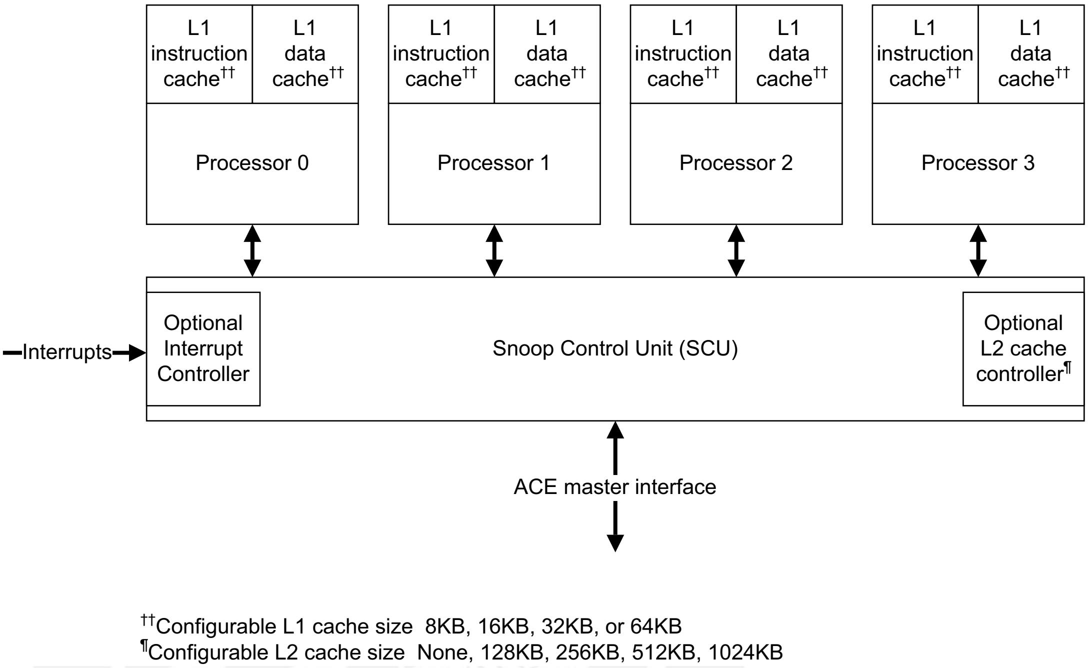


Figure 1-1 Example multiprocessor configuration

You can implement one processor in a Cortex-A7 MPCore processor design. In this configuration, the SCU is still provided. See *Configurable options* on page 1-7 for more information.

See *Processor components* on page 2-2 for more information about the functional components.

1.2 Compliance

The Cortex-A7 MPCore processor complies with, or implements, the specifications described in:

- *ARM architecture*
- *Advanced Microcontroller Bus Architectures*
- *Debug architecture*
- *Generic Interrupt Controller architecture* on page 1-4
- *Generic Timer architecture* on page 1-4.

This TRM complements architecture reference manuals, architecture specifications, protocol specifications, and relevant external standards. It does not duplicate information from these sources.

1.2.1 ARM architecture

The Cortex-A7 MPCore processor implements the ARMv7-A architecture with the following architecture extensions:

- *Advanced Single Instruction Multiple Data version 2* (SIMDv2) architecture extension for integer and floating-point vector operations.

  Note

  The Advanced SIMD architecture extension, its associated implementations, and supporting software, are commonly referred to as NEON technology.

- *Vector Floating-Point version 4* (VFPv4) architecture extension for floating-point computation that is fully compliant with the IEEE 754 standard.
- Security Extensions for implementation of enhanced security.
- Virtualization Extensions for the development of virtualized systems that enable the switching of guest operating systems.
- *Large Physical Address* (LPA) Extension for address translation of up to 40 bit physical addresses.
- Multiprocessing Extensions for multiprocessing functionality.

See the *ARM Architecture Reference Manual, ARMv7-A and ARMv7-R edition* for more information.

1.2.2 Advanced Microcontroller Bus Architectures

The Cortex-A7 MPCore processor ACE and debug interfaces comply with the:

- AMBA 4 *Advanced eXtensible Interface* (AXI) protocol. See the *AMBA AXI Protocol Specification*.
- AMBA *Advanced Peripheral Bus* (APB) protocol. See the *AMBA APB Protocol Specification.*

1.2.3 Debug architecture

The Cortex-A7 MPCore processor implements the ARMv7.1 Debug architecture that includes support for CoreSight. For more information, see the:

- *CoreSight Architecture Specification*
- *ARM Architecture Reference Manual, ARMv7-A and ARMv7-R edition*.

1.2.4 Generic Interrupt Controller architecture

The Cortex-A7 MPCore processor implements the *Generic Interrupt Controller* (GIC) v2.0 architecture that includes support for the Virtualization Extensions. See the *ARM Generic Interrupt Controller Architecture Specification*.

1.2.5 Generic Timer architecture

The Cortex-A7 MPCore processor implements the ARM Generic Timer architecture that includes support for the Virtualization Extensions. See the *ARM Architecture Reference Manual, ARMv7-A and ARMv7-R edition*.

1.3 Features

The processor includes the following features:

- Full implementation of the ARMv7-A architecture instruction set with the architecture extensions listed in *Compliance* on page 1-3
- In-order pipeline with direct and indirect branch prediction
- Harvard *Level 1* (L1) memory system with a *Memory Management Unit* (MMU)
- *Level 2* (L2) memory system
- APB debug interface that supports integer processor clock ratios up to and including 1:1.
- Trace support through an *Embedded Trace Macrocell* (ETM) interface
- Optional VFPv4-D16 FPU with trapless execution or *Media Processing Engine* (MPE) with NEON technology.

Note

When FPU option is selected without NEON, the FPU is VFPv4-D16 and uses 16-bit double-precision registers. When the FPU is implemented with NEON, the FPU is VFPv4-D32 and uses 32 double-precision registers. This register bank is shared with NEON.

1.4 Interfaces

The Cortex-A7 MPCore processor has the following external interfaces:

- ACE
- APB
- ETM
- *Design For Test* (DFT)
- *Memory Built-In Self Test* (MBIST) controller.

See *Interfaces* on page 2-8 for more information on these interfaces.

1.5 Configurable options

Table 1-1 shows the Cortex-A7 MPCore RTL configurable options.

Table 1-1 Configurable options for the Cortex-A7 MPCore RTL

Feature	Range of options	Default value
Processor-level configuration options:[a]		
Floating-Point Unit (FPU) or *Media Processing Engine* (NEON)	Neither, FPU only, FPU and NEON[b]	FPU and NEON
Global configuration options:		
Instruction cache size	8KB, 16KB, 32KB, or 64KB	32KB
Data cache size	8KB, 16KB, 32KB, or 64KB	32KB
L2 cache controller	Present or not present	Present
L2 cache sizes	None, 128KB, 256KB, 512KB, 1024KB	256KB
L2 data RAM cycle latency	2 cycles or 3 cycles	2 cycles
Shared Peripheral Interrupts	0-480 in steps of 32	0
Number of processors	1, 2, 3, or 4	2
Integrated GIC	Present or not present	Not present

[a] These options can be configured independently for each processor in the Cortex-A7 MPCore processor.

[b] If NEON is selected, FPU is included.

1.6 Test features

The Cortex-A7 MPCore processor is delivered as fully-synthesizable RTL and is a fully-static design. Scan-chains and test wrappers for production test can be inserted into the design by the synthesis tools during implementation. See the relevant implementation reference methodology documentation for more information.

You can perform production test of the cache and TLB RAMs, and the L2 cache RAMs if present, through a dedicated MBIST interface. See the *Cortex-A7 MPCore Integration Manual* for more information about this interface, and how to control it.

1.7 Product documentation and design flow

This section describes the Cortex-A7 MPCore books, how they relate to the design flow, and the relevant architectural standards and protocols.

See *Additional reading* on page ix for more information about the books described in this section.

1.7.1 Documentation

The Cortex-A7 MPCore documentation is as follows:

Technical Reference Manual

The *Technical Reference Manual* (TRM) describes the functionality and the effects of functional options on the behavior of the Cortex-A7 MPCore processor. It is required at all stages of the design flow. The choices made in the design flow can mean that some behavior described in the TRM is not relevant. If you are programming the Cortex-A7 MPCore processor then contact:

- the implementer to determine the build configuration of the implementation
- the integrator to determine the pin configuration of the *System-on-Chip* SoC that you are using.

Configuration and Sign-off Guide

The *Configuration and Sign-off Guide* (CSG) describes:

- the available build configuration options and related issues in selecting them
- how to configure the *Register Transfer Level* (RTL) description with the build configuration options
- how to integrate RAM arrays
- how to run test vectors
- the processes to sign-off the configured design.

The ARM product deliverables include reference scripts and information about using them to implement your design. Reference methodology flows supplied by ARM are example reference implementations. Contact your EDA vendor for EDA tool support.

The CSG is a confidential book that is only available to licensees.

Integration Manual

The *Integration Manual* (IM) describes how to integrate the Cortex-A7 MPCore processor into a SoC. It includes a description of the pins that the integrator must tie off to configure the macrocell for the required integration. Some of the integration is affected by the configuration options used when implementing the Cortex-A7 MPCore processor.

The IM is a confidential book that is only available to licensees.

1.7.2 Design flow

The Cortex-A7 MPCore processor is delivered as synthesizable Verilog RTL. Before it can be used in a product, it must go through the following process:

Implementation	The implementer configures and synthesizes the RTL to produce a hard macrocell. This might include integrating the RAMs into the design.

Integration	The integrator connects the implemented design into a SoC. This includes connecting it to a memory system and peripherals.
Programming	The system programmer develops the software required to configure and initialize the processor, and tests the required application software.

Each stage of the process:

- can be performed by a different party
- can include implementation and integration choices that affect the behavior and features of the Cortex-A7 MPCore processor.

The operation of the final device depends on:

Build configuration

The implementer chooses the options that affect how the RTL source files are pre-processed. They usually include or exclude logic that can affect the area or maximum frequency of the resulting macrocell.

Configuration inputs

The integrator configures some features of the processor by tying inputs to specific values. These configurations affect the start-up behavior before any software configuration is made. They can also limit the options available to the software.

Software configuration

The programmer configures the processor by programming particular values into software-visible registers. This affects the behavior of the processor.

Note

This manual refers to implementation-defined features that are applicable to build configuration options. References to a feature that is *included* means that the appropriate build and pin configuration options have been selected, while references to an *enabled* feature means one that has also been configured by software.

1.8 Product revisions

This section describes the differences in functionality between product revisions:

r0p0-r0p1 Functional changes are:

- ID register value changed to reflect product revision status:

  Main ID Register `0x410FC071`

  Debug Peripheral ID 2 `0x1B`

  Performance Monitors Peripheral ID2 `0x1B`
- support for a redundant internal GIC, see *GIC configuration* on page 8-5
- various engineering errata fixes.

r0p1-r0p2 Functional changes are:

- ID register value changed to reflect product revision status:

  Main ID Register `0x410FC072`

  Debug Peripheral ID 2 `0x2B`

  Performance Monitors Peripheral ID2 `0x2B`
- various engineering errata fixes.

r0p2-r0p3 Functional changes are:

- ID register value changed to reflect product revision status:

  Main ID Register `0x410FC073`

  Debug Peripheral ID 2 `0x3B`

  Performance Monitors Peripheral ID2 `0x3B`
- enhancement of the read allocate mode operation to optimize performance
- various engineering errata fixes.

Chapter 2

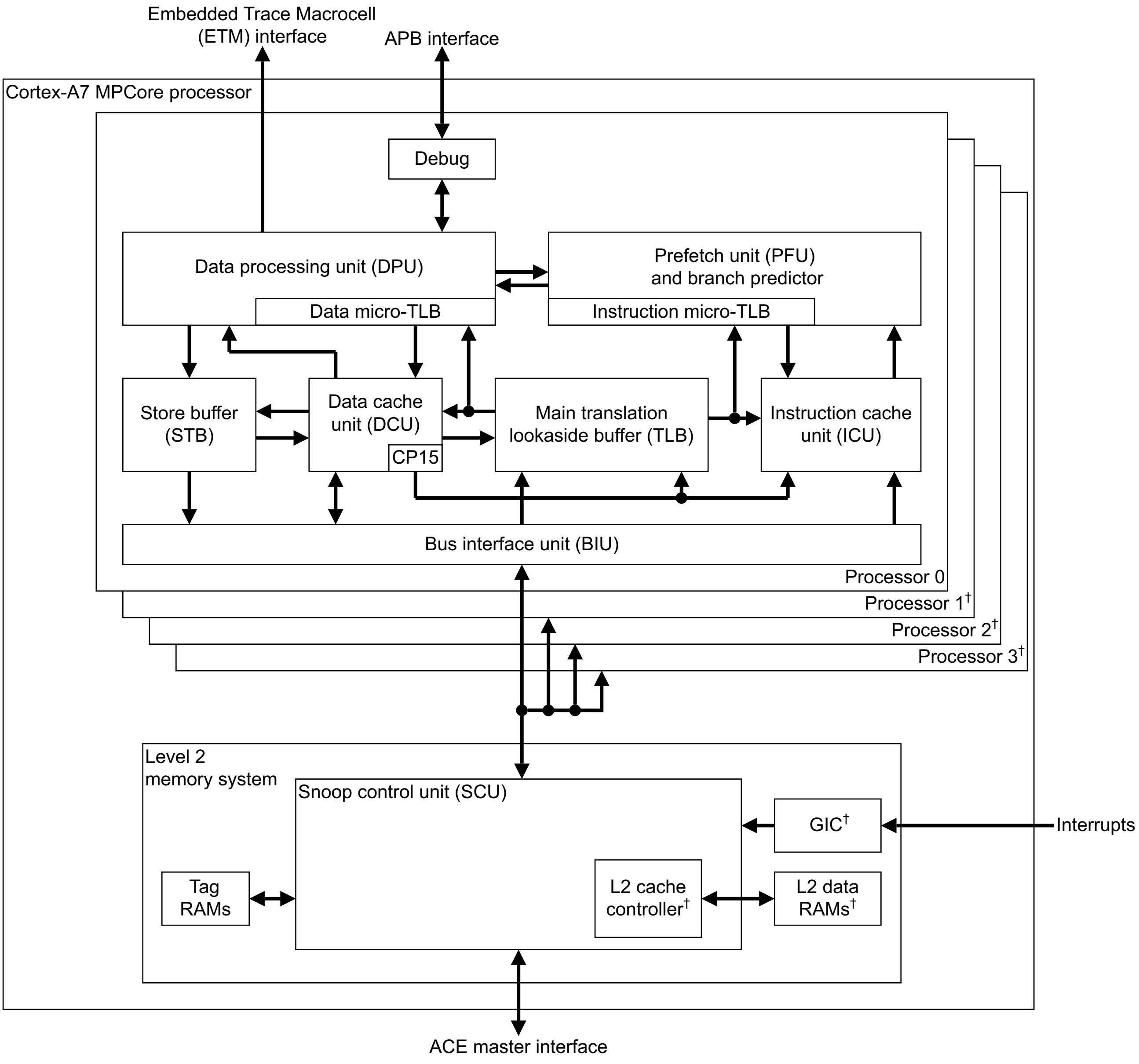
Functional Description

This chapter describes the functionality of the Cortex-A7 MPCore processor. It contains the following sections:

- *About the Cortex-A7 MPCore processor functions* on page 2-2
- *Interfaces* on page 2-8
- *Clocking and resets* on page 2-9
- *Power management* on page 2-12.

2.1 About the Cortex-A7 MPCore processor functions

Figure 2-1 shows a top-level functional diagram of the Cortex-A7 MPCore processor.



†Optional

Figure 2-1 Cortex-A7 MPCore processor top-level diagram

2.1.1 Processor components

The following sections describe the main components and their functions:

- *Data Processing Unit* on page 2-3
- *System control coprocessor* on page 2-3
- *Instruction side memory system* on page 2-3
- *Data side memory system* on page 2-4
- *L1 memory system* on page 2-6

- *Media Processing Engine* on page 2-7
- *Floating-Point Unit* on page 2-7
- *L2 memory system* on page 2-6
- *Debug* on page 2-7
- *Performance monitoring* on page 2-7.

Data Processing Unit

The *Data Processing Unit* (DPU) holds most of the program-visible state of the processor, such as general-purpose registers, status registers and control registers. It decodes and executes instructions, operating on data held in the registers in accordance with the ARM Architecture. Instructions are fed to the DPU from the *Prefetch Unit* (PFU). The DPU executes instructions that require data to be transferred to or from the memory system by interfacing to the *Data Cache Unit* (DCU), which manages all load and store operations. See Chapter 3 *Programmers Model* for more information.

System control coprocessor

The system control coprocessor, CP15, provides configuration and control of the memory system and its associated functionality.

See Chapter 4 *System Control* for more information.

Instruction side memory system

The instruction side memory system includes:

- *Instruction Cache Unit*
- *Prefetch Unit*.

Instruction Cache Unit

The *Instruction Cache Unit* (ICU) contains the Instruction Cache controller and its associated linefill buffer. The Cortex-A7 MPCore ICache is two-way set associative and uses *Virtually Indexed Physically Tagged* (VIPT) cache-lines holding up to 8 ARM or Thumb 32-bit instructions or up to 16 Thumb 16-bit instructions.

Prefetch Unit

The *Prefetch Unit* (PFU) obtains instructions from the instruction cache or from external memory and predicts the outcome of branches in the instruction stream, then passes the instructions to the DPU for processing. In any given cycle, up to a maximum of four instructions can be fetched and two can be passed to the DPU.

Branch Target Instruction Cache

The PFU also contains a four-entry deep *Branch Target Instruction Cache* (BTIC). Each entry stores up to two instruction cache fetches and enables the branch shadow of predicted taken B and BL instructions to be eliminated. The BTIC implementation is architecturally transparent, so it does not have to be flushed on a context switch.

Branch Target Address Cache

The PFU also contains a eight-entry deep *Branch Target Address Cache* (BTAC) used to predict the target address of certain indirect branches. The BTAC implementation is architecturally transparent, so it does not have to be flushed on a context switch.

Branch prediction	The branch predictor is a global type that uses history registers and a 256-entry pattern history table.
Return stack	The PFU includes an 8-entry return stack to accelerate returns from procedure calls. For each procedure call, the return address is pushed onto a hardware stack. When a procedure return is recognized, the address held in the return stack is popped, and the PFU uses it as the predicted return address. The return stack is architecturally transparent, so it does not have to be flushed on a context switch.

Data side memory system

This section describes the following:

- *Data Cache Unit*
- *Store Buffer* on page 2-6
- *Bus Interface Unit and SCU interface* on page 2-6.

Data Cache Unit

The *Data Cache Unit* (DCU) consists of the following sub-blocks:

- The *Level 1* (L1) data cache controller, which generates the control signals for the associated embedded tag, data, and dirty memory (RAMs) and arbitrates between the different sources requesting access to the memory resources. The data cache is 4-way set associative and uses a *Physically Indexed Physically Tagged* (PIPT) scheme for lookup which enables unambiguous address management in the system.

- The load/store pipeline that interfaces with the DPU and main TLB.

- The system coprocessor controller that performs cache maintenance operations directly on the data cache and indirectly on the instruction cache through an interface with the ICU.

- An interface to receive coherency requests from the *Snoop Control Unit* (SCU).

The DCU contains a combined local and global exclusive monitor. This monitor can be set to the exclusive state only by a `LDREX` instruction executing on the local processor, and can be cleared to the open access state by:

- a `STREX` instruction on the local processor or a store to the same shared cache line on another processor

- the cache line being evicted for other reasons

- a `CLREX` instruction.

The Cortex-A7 MPCore processor uses the MOESI protocol, with ACE modified equivalents of MOESI states, to maintain data coherency between multiple processors. MOESI describes the state that a shareable line in a L1 data cache can be in:

M	Modified/UniqueDirty (UD). The line is only in this cache and is dirty.
O	Owned/SharedDirty (SD). The line is possibly in more than one cache and is dirty.
E	Exclusive/UniqueClean (UC). The line is only in this cache and is clean.
S	Shared/SharedClean (SC). The line is possibly in more than one cache and is clean.
I	Invalid/Invalid (I). The line is not in this cache.

The DCU stores the MOESI state of the cache line in the tag and dirty RAMs.

Read allocate mode

The L1 data cache only supports a Write-Back policy. It normally allocates a cache line on either a read miss or a write miss. However, there are some situations where allocating on writes is undesirable, such as executing the C standard library `memset()` function to clear a large block of memory to a known value. Writing large blocks of data like this can pollute the cache with unnecessary data. It can also waste power and performance if a linefill must be performed only to discard the linefill data because the entire line was subsequently written by the `memset()`.

To prevent this, the *Bus Interface Unit* (BIU) includes logic to detect when a full cache line has been written by the processor before the linefill has completed. If this situation is detected on three consecutive linefills, it switches into read allocate mode. When in read allocate mode, loads behave as normal and can still cause linefills, and writes still lookup in the cache but, if they miss, they write out to L2 rather than starting a linefill.

The BIU continues in read allocate mode until it detects either a cacheable write burst to L2 that is not a full cache line, or there is a load to the same line as is currently being written to L2.

A secondary read allocate mode applies when the L2 cache is integrated. After 127 consecutive cache line sized writes to L2 are detected, L2 read allocate mode is entered. When in L2 read allocate mode, loads behave as normal and can still cause linefills, and writes still lookup in the cache but, if they miss, they write out to L3 rather than starting a linefill. L2 read allocate mode continues until there is a cacheable write burst that is not a full cache line, or there is a load to the same line as is currently being written to L3.

Note

The number of consecutive cache line sized writes to enter a secondary read allocate mode was 7 prior to product revision r0p3.

Data cache invalidate on reset

The ARMv7 *Virtual Memory System Architecture* (VMSA) does not support a CP15 operation to invalidate the entire data cache. If this function is required in software, it must be constructed by iterating over the cache geometry and executing a series of individual CP15 invalidate by set/way instructions.

In normal usage the only time the entire data cache must be invalidated is on reset. The processor provides this functionality by default. If it is not required on reset the invalidate operation can be disabled by asserting and holding the appropriate external **L1RSTDISABLE** signal for a processor when the corresponding reset signal is deasserted.

In parallel to the data cache invalidate, the DCU also sends an invalidate-all request to the ICU and the TLB, unless **L1RSTDISABLE** is asserted.

Store Buffer

The *Store Buffer* (STB) holds store operations when they have left the load/store pipeline and have been committed by the DPU. From the STB, a store can request access to the cache RAMs in the DCU, request the BIU to initiate linefills, or request the BIU to write the data out on the external write channel. External data writes are through the SCU.

The STB can merge:

- several store transactions into a single transaction if they are to the same 64-bit aligned address. The STB is also used to queue up CP15 maintenance operations before they are broadcast to other processors in the multiprocessor device.
- Multiple writes into an AXI write burst.

Bus Interface Unit and SCU interface

The *Bus Interface Unit* (BIU) contains the SCU interface and buffers to decouple the interface from the cache and STB. The BIU interface and the SCU always operate at the processor frequency.

A write buffer is available to hold:

- data from cache evictions or non-cacheable write bursts before they are written out to the SCU.
- the addresses of outstanding ACE write transactions to permit hazard checking against other outstanding requests in the system.

L1 memory system

The processor L1 memory system includes the following features:

- separate instruction and data caches
- export of memory attributes for system caches.

The caches have the following features:

- support for instruction and data cache sizes between 8KB and 64KB
- pseudo-random cache replacement policy
- ability to disable each cache independently
- streaming of sequential data from `LDM` and `LDRD` operations, and sequential instruction fetches
- critical word first linefill on a cache miss
- all the cache RAM blocks and associated tag and valid RAM blocks if implemented using standard RAM compilers.

See Chapter 6 *L1 Memory System* for more information.

L2 memory system

The L2 memory system contains:

- The SCU that connects between one to four processors to the external memory system through the ACE master interface. The SCU maintains data cache coherency between the processors and arbitrates L2 requests from the processors.

  The SCU includes support for data security using the implemented Security Extensions.

Note

The SCU does not support hardware management of coherency of the instruction caches.

- An optional L2 cache that:
  - has configurable cache RAM sizes of 128KB, 256KB, 512KB, or 1MB
  - is 8-way set associative
  - supports 64-byte cache lines.
- One ACE master interface. All transactions are routed through the interface.

See Chapter 7 *L2 Memory System* for more information.

Optional Generic Interrupt Controller

The optional integrated GIC manages interrupt sources and behavior, and can route interrupts to individual processors. It permits software to mask, enable and disable interrupts from individual sources, to prioritize, in hardware, individual sources and to generate software interrupts. It also provides support for the Security and Virtualization Extensions. The GIC accepts interrupts asserted at the system level and can signal them to each processor it is connected to. This can result in an IRQ or FIQ exception being taken.

See Chapter 8 *Generic Interrupt Controller* for more information.

Media Processing Engine

The optional *Media Processing Engine* (MPE) implements ARM NEON technology, a media and signal processing architecture that adds instructions targeted at audio, video, 3-D graphics, image, and speech processing. Advanced SIMD instructions are available in both ARM and Thumb states.

See the *Cortex-A7 MPCore NEON Media Processing Engine Technical Reference Manual* for more information.

Floating-Point Unit

The optional *Floating-Point Unit* (FPU) implements the ARMv7 VFPv4-D16 architecture and includes the VFP register file and status registers. It performs floating-point operations on the data held in the VFP register file.

See the *Cortex-A7 MPCore Floating-Point Unit Technical Reference Manual* for more information.

Debug

The Cortex-A7 MPCore processor has a CoreSight compliant *Advanced Peripheral Bus version 3* (APBv3) debug interface. This permits system access to debug resources, for example, the setting of watchpoints and breakpoints. The processor provides extensive support for real-time debug and performance profiling.

See Chapter 10 *Debug* for more information.

Performance monitoring

The Cortex-A7 MPCore processor provides performance counters and event monitors that can be configured to gather statistics on the operation of the processor and the memory system. See Chapter 11 *Performance Monitoring Unit* for more information.

2.2 Interfaces

The Cortex-A7 MPCore processor has the following external interfaces:

- *ACE*
- *APB*
- *ETM*
- *DFT*
- *MBIST controller*.

2.2.1 ACE

The processor implements the AMBA 4 *AXI Coherency Extension*s (ACE) interface.

ACE is an extension to the AXI protocol and provides the following enhancements:

- support for hardware coherent caches
- barrier transactions that guarantee transaction ordering
- distributed virtual memory messaging, enabling management of a virtual memory system.

See *ACE master interface signals* on page A-8 for more information on the signals.

See the *AMBA AXI and ACE Protocol Specification* for more information.

2.2.2 APB

The processor implements an AMBA 3 APB slave interface that enables access to the debug registers. See the *CoreSight Architecture Specification* for more information.

2.2.3 ETM

The *Embedded Trace Macrocell* (ETM) interface enables you to connect an external ETM unit to the Cortex-A7 MPCore processor for real-time code and data tracing of the processor in an embedded system.

Each processor in the Cortex-A7 MPCore processor exports its own ETM interface. Each interface runs at the full speed of the Cortex-A7 MPCore processor. The ETM interface connects directly to the external CoreSight ETM-A7 without any additional glue logic.

See *ETM interface signals* on page A-16 for more information on the signals.

See the *CoreSight ETM-A7 Technical Reference Manual* for more information on the ETM interface.

2.2.4 DFT

The processor implements a *Design For Test* (DFT) interface that enables an industry standard *Automatic Test Pattern Generation* (ATPG) tool to test logic outside of the embedded memories. See *DFT interface* on page A-17 for information on these test signals.

2.2.5 MBIST controller

The *Memory Built In Self Test* (MBIST) controller interface provides support for manufacturing test of the memories embedded in the Cortex-A7 MPCore processor. MBIST is the industry standard method of testing embedded memories. MBIST works by performing sequences of reads and writes to the memory based on test algorithms. See *MBIST interface* on page A-17 for information on this interface.

2.3 Clocking and resets

The following sections describe clocking and resets:

- *Clocking*
- *Resets* on page 2-10.

2.3.1 Clocking

The Cortex-A7 MPCore processor has a single clock input, **CLKIN**. All processors in the Cortex-A7 MPCore processor and the SCU are clocked with a distributed version of **CLKIN**. The Cortex-A7 MPCore processor synchronizes the input signals:

- **nCOREPORESET[3:0]**
- **nCORERESET[3:0]**
- **nDBGRESET[3:0]**
- **nL2RESET**
- **nIRQ[3:0]**
- **nFIQ[3:0]**
- **nVFIQ[3:0]**
- **nVIRQ[3:0]**
- **DBGEN[3:0]**
- **SPIDEN[3:0]**
- **EDBGRQ[3:0]**.

All other external signals must be synchronous with reference to **CLKIN**.

ACE master interface clocking

The SCU interface supports integer ratios of the **CLKIN** frequency , for example 1:1, 2:1, 3:1. These ratios are configured through external clock enable signals. In all cases AXI transfers remain synchronous. The ACE master interface includes the **ACLKENM** clock enable signal.

ACLKENM asserts one **CLKIN** cycle prior to the rising edge of the external ACE clock signal, **ACLKM**. Software can change the **CLKIN** to **ACLKM** frequency ratio dynamically using **ACLKENM**.

Figure 2-2 shows a timing example of **ACLKENM** that changes the **CLKIN** to **ACLKM** frequency ratio from 3:1 to 1:1.

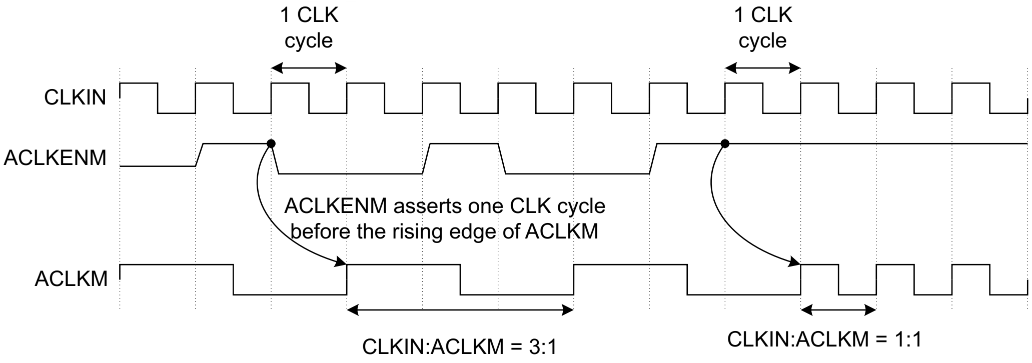


Figure 2-2 ACLKENM with CLKIN:ACLKM ratio changing from 3:1 to 1:1

Note

Figure 2-2 on page 2-9 shows the timing relationship between the AXI master clock, **ACLKM** and **ACLKENM**, where **ACLKENM** asserts one clock cycle before the rising edge of **ACLKM**. It is important that the relationship between **ACLKM** and **ACLKENM** is maintained.

Debug interface clocking

The processor includes an APB interface to access the debug and performance monitoring registers. Internally this interface is driven from **CLKIN**. A separate enable signal, **PCLKENDBG**, is provided to enable the external APB bus to be driven at a lower frequency, which must be an integer ratio of **CLKIN**. If the debug infrastructure in the system is required to be fully asynchronous to the processor clock, you can use a synchronizing component to connect the external AMBA APB to the processor.

2.3.2 Resets

The Cortex-A7 MPCore processor has multiple reset domains with the following reset input signals:

nCOREPORESET[3:0]	These power-on reset signals initialize all the processor logic, including CPU Debug, and breakpoint and watchpoint logic in the processor power domains. They do not reset debug logic in the debug power domain.
nCORERESET[3:0]	These are the primary reset signals which initialize the processor logic in the processor power domains, not including the debug, breakpoint and watchpoint logic.
nDBGRESET[3:0]	At the Cortex-A7 level, these signals reset only the debug, and breakpoint and watchpoint logic in the processor power domain. At the Cortex-A7 integration layer level, these signals also reset the debug logic for each processor, which is in the debug power domain.

These reset signals are 4-bit signals, where each bit represents one processor in the multiprocessor device.

The following reset input signals are single-bit fanouts to all the processors in the multiprocessor device:

nMBISTRESET	This signal resets the device for entry into MBIST mode.
nL2RESET	This signal resets the L2 memory system and the logic in the SCU.

All of these resets:

- can be asynchronously asserted and de-asserted.
- are active-LOW signals that reset logic in the appropriate domain of the Cortex-A7 MPCore processor.

In Table 2-1 on page 2-11, [3:0] specifies the processor configuration and [n] designates the processor that is reset.

Table 2-1 Valid reset combinations

Reset combination	Signals	Value	Description
All processor power-on reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	all = 0 all = 0[a] all = 0[a] 0	All logic is held in reset.
Individual processor power-on reset with Debug reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	[n] = 0 [n] = 0[a] [n] = 0[a] 1	Individual processor and Debug are held in reset.
All processor power-on and L2 reset with Debug active	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	all = 0 all = 0 all = 1 0	All processors and L2 are held in reset, so they can be powered up. This enables external debug over power down for all processors.
Individual processor power-on reset with Debug active	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	[n] = 0 [n] = 0 [n] = 1 1	Individual processor is held in reset, so that the processor can be powered up. This enables external debug over power down for the processor that is held in reset.
All processors software reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	all = 1 all = 0 all = 1 1	All logic excluding Debug and L2 memory system is held in reset. All breakpoints and watchpoints are retained.
All processors software reset and L2 reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	all = 1 all = 0 all = 1 0	All logic excluding Debug is held in reset. All breakpoints and watchpoints are retained.
Individual processor software reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	[n] = 1 [n] = 0 [n] = 1 1	Individual processor logic excluding Debug and ETM is held in reset. Breakpoints and watchpoints for that processor are retained.
All processors debug reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	all = 1 all = 1 all = 0 1	Debug is held in reset.
Individual processor Debug reset	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	[n] = 1 [n] = 1 [n] = 0 1	Individual processor Debug is held in reset.
Run mode	**nCOREPORESET [3:0]** **nCORERESET [3:0]** **nDBGRESET [3:0]** **nL2RESET**	1 1 1 1	No logic is held in reset.

[a] For power-on reset or processor reset, **nCOREPORESET** must be asserted. The remaining processor resets, **nCORERESET** and **nDBGRESET** can be asserted, but is not required.

2.4 Power management

The Cortex-A7 MPCore processor provides mechanisms and support to control both dynamic and static power dissipation. The individual processors in the Cortex-A7 MPCore processor support four main levels of power management. This section describes:

- *Power domains*
- *Power modes* on page 2-13
- *Event communication using WFE or SEV* on page 2-20
- *Communication to the Power Management Controller* on page 2-20.

2.4.1 Power domains

The power domains that the Cortex-A7 MPCore processor supports are:

- Vcore<n>, that represents the processor <n>, which includes the optional Advanced SIMD and VFP Extensions and the L1 TLB and cache RAM.
- Vscu that represents the Cortex-A7 L2 with the duplicate tag RAMs, which includes the SCU, the optional L2 cache controller, and the optional integrated GIC.
- Vscu_ram, which includes the L2 cache RAM.

The Cortex-A7 integration layer supports the following power domain:

- Vdebug, which includes support for v7.1 Debug architecture, CoreSight debug and trace components.

The separate SCU power domains can remain active even when all the processors are powered down. This enables the Cortex-A7 MPCore processor to accept snoops from an external device to control the L2 cache.

Figure 2-3 on page 2-13 shows an example of the domains embedded in a *System-on-Chip* (SoC) power domain.

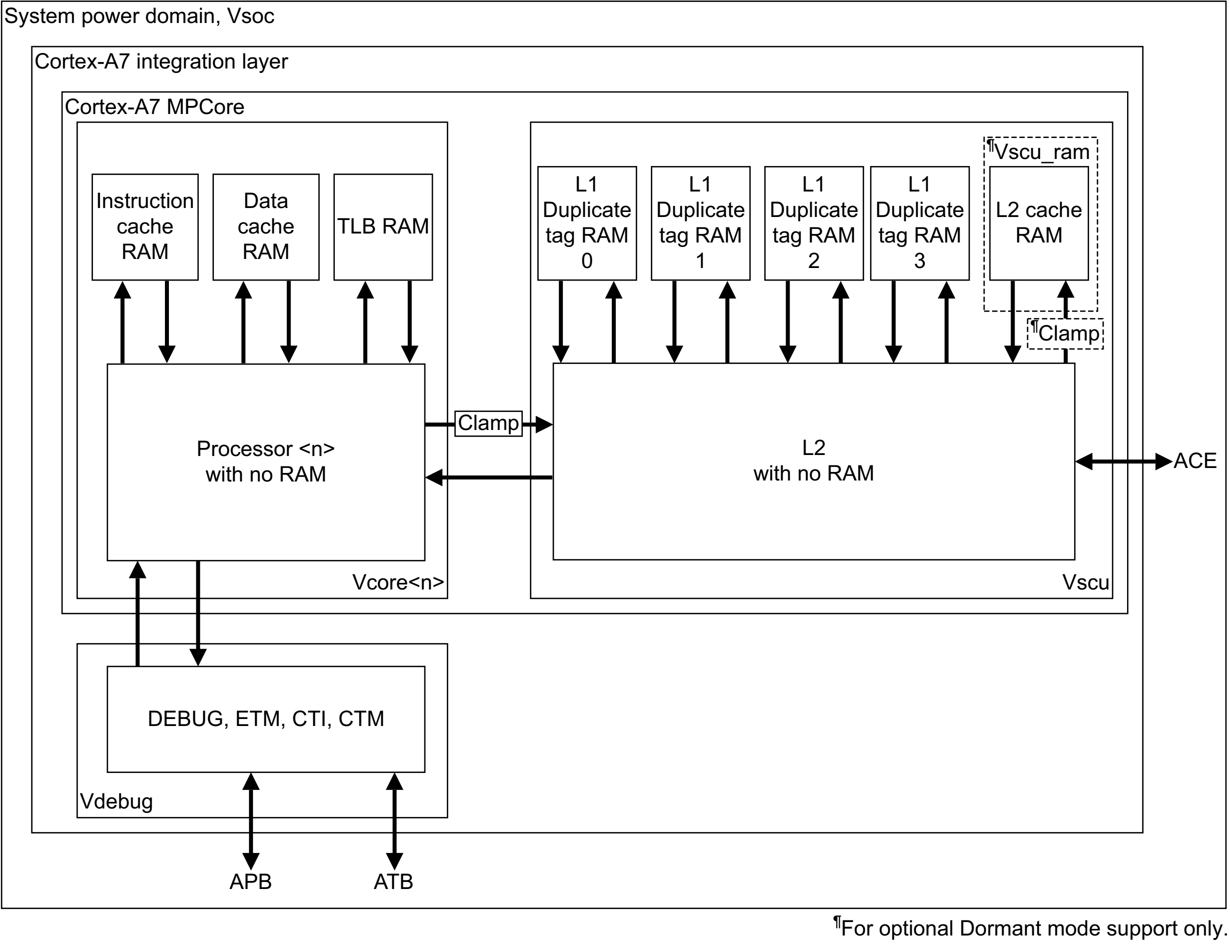


Figure 2-3 Power domains

Note

Figure 2-3 shows the Vdebug power domain for example only. It is a domain in the Cortex-A7 integration layer and is not described in this manual.

Each Vcore<n> domain uses a single clock that is architecturally gated at the top level of the processor to minimize dynamic power consumption without removing power completely from the processor.

At the SoC integration level, the processor logic in Vcore<n> and Vscu can be isolated and powered down completely through the instantiation of clamps on all of the external interfaces inside the Vsoc domain. .

2.4.2 Power modes

The power domains can be controlled independently to give different combinations of power-up and power-down domains. However, only some power-up and power-down domain combinations are valid and supported.

Table 2-2 .shows the supported power configurations for the different possible modes of operation.

Table 2-2 Supported power modes

Mode	Vsoc	Vscu	Vscu_ram	Vcore<n[a]>
Run mode	Powered up	Powered up	Powered up	Powered up
Processor n[a] in shutdown mode	Powered up	Powered up	Powered up	Powered down
Cortex-A7 MPCore processor in shutdown mode	Powered up	Powered down	Powered down	Powered down
Cortex-A7 MPCore processor in Dormant mode	Powered up	Powered down	Powered up	Powered down

[a] Where n represents 0-3.

There are specific requirements that you must meet to power up and power down each power domain within the processor. Not adhering to these requirements can lead to UNPREDICTABLE results.

The dynamic power management and power-up and power-down sequences in the following sections are the only power sequences that ARM recommends. Any deviation from these sequences can lead to UNPREDICTABLE results.

The supported power modes are:
- *Run mode*
- *Standby mode*
- *Individual processor shutdown mode* on page 2-17
- *Multiprocessor device shutdown mode* on page 2-18
- *Dormant mode* on page 2-19.

Run mode

This is the normal mode of operation where all of the processor functionality is available. The Cortex-A7 MPCore processor uses gated clocks and gates to disable inputs to unused functional blocks. Only the logic in use to perform an operation consumes any dynamic power.

Standby mode

The following sections describes the methods of entering standby mode:
- *Processor Wait for Interrupt*
- *Processor Wait for Event* on page 2-15
- *L2 Wait for Interrupt* on page 2-16.

Processor Wait for Interrupt

Wait for Interrupt is a feature of the ARMv7-A architecture that puts the processor in a low power state by disabling most of the clocks in the processor while keeping the processor powered up. This reduces the power drawn to the static leakage current, leaving a small clock power overhead to enable the processor to wake up from WFI mode.

A processor enters into WFI mode by executing the `WFI` instruction.

When executing the `WFI` instruction, the processor waits for all instructions in the processor to retire before entering the idle or low power state. The `WFI` instruction ensures that all explicit memory accesses occurred before the `WFI` instruction in program order, have retired. For example, the `WFI` instruction ensures that the following instructions received the required data or responses from the L2 memory system:

- load instructions
- cache and TLB maintenance operations
- store exclusives instructions.

In addition, the `WFI` instruction ensures that store instructions have updated the cache or have been issued to the L2 memory system.

While the processor is in WFI mode, the clocks in the processor are temporarily enabled without causing the processor to exit WFI mode, when any of the following events are detected:

- an L2 snoop request that must be serviced by the processor L1 data cache
- a cache or TLB maintenance operation that must be serviced by the processor L1 instruction cache, data cache, or TLB
- an APB access to the debug or trace registers residing in the processor power domain.

Exit from WFI mode occurs when the processor detects a reset or one of the WFI wake up events as described in the *ARM Architecture Reference Manual*.

On entry into WFI mode, **STANDBYWFI** for that processor is asserted. Assertion of **STANDBYWFI** guarantees that the processor is in idle and low power state. **STANDBYWFI** continues to assert even if the clocks in the processor are temporarily enabled because of an L2 snoop request, cache, or TLB maintenance operation or an APB access.

Figure 2-4 shows the upper bound for the **STANDBYWFI** deassertion timing after the assertion of **nIRQ** or **nFIQ** inputs.

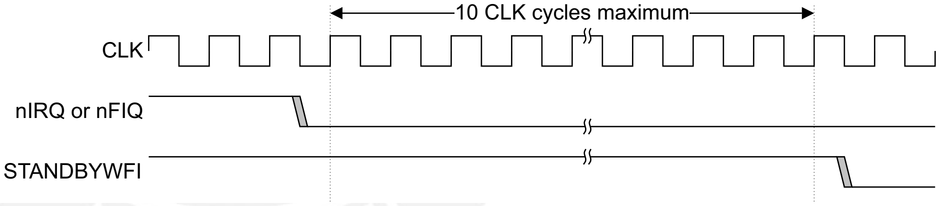


Figure 2-4 STANDBYWFI deassertion timing

Processor Wait for Event

Wait for Event is a feature of the ARMv7-A architecture that uses a locking mechanism based on events to put the processor in a low power state by disabling most of the clocks in the processor while keeping the processor powered up. This reduces the power drawn to the static leakage current, leaving a small clock power overhead to enable the processor to wake up from WFE mode.

A processor enters into WFE mode by executing the `WFE` instruction. When executing the `WFE` instruction, the processor waits for all instructions in the processor to complete before entering the idle or low power state.

While the processor is in WFE mode, the clocks in the processor are temporarily enabled without causing the processor to exit WFE mode, when any of the following events are detected:

- an L2 snoop request that must be serviced by the processor L1 data cache
- a cache or TLB maintenance operation that must be serviced by the processor L1 instruction cache, data cache, or TLB

- an APB access to the debug or trace registers residing in the processor power domain.

Exit from WFE mode occurs when the processor detects a reset, the assertion of the **EVENTI** input signal, or one of the WFI wake up events as described in the *ARM Architecture Reference Manual*.

On entry into WFE mode, **STANDBYWFE** for that processor is asserted. Assertion of **STANDBYWFE** guarantees that the processor is in idle and low power state. **STANDBYWFE** continues to assert even if the clocks in the processor are temporarily enabled because of an L2 snoop request, cache, and TLB maintenance operation or an APB access.

The upper bound for the **STANDBYWFE** deassertion timing after the assertion of **nIRQ** or **nFIQ** inputs is identical to **STANDBYWFI** as shown in Figure 2-4 on page 2-15.

L2 Wait for Interrupt

When all the processors are in WFI mode, the shared L2 memory system logic that is common to all the processors can also enter a WFI mode. In L2 WFI mode, all internal clocks in the processor are disabled.

Entry into L2 WFI mode can only occur if specific requirements are met and the following sequence applied:

- All processors are in WFI mode and therefore, all the processors **STANDBYWFI** outputs are asserted. Assertion of all the processors **STANDBYWFI** outputs guarantee that all the processors are in idle and low power state. All clocks in the processor, with the exception of a small amount of clock wake up logic, are disabled.
- The SoC asserts the input pin **ACINACTM** to idle the AXI master interface. This prevents the L2 memory system from accepting any new requests from the AXI master interface.
- When the L2 memory system completed the outstanding transactions for AXI interfaces, it can then enter the low power state, L2 WFI mode. On entry into L2 WFI mode, **STANDBYWFIL2** is asserted. Assertion of **STANDBYWFIL2** guarantees that the L2 memory system is in idle and does not accept any new transactions.

Exit from L2 WFI mode occurs on one of the following WFI wake up events:

- a physical IRQ or FIQ interrupt
- a debug event
- power-on or soft reset.

When the processor exits from WFI mode, **STANDBYWFI** for that processor is deasserted. When the L2 memory system logic exits from WFI mode, **STANDBYWFIL2** is deasserted. The SoC must continue to assert **ACINACTM** until **STANDBWFIL2** has deasserted.

Figure 2-5 on page 2-17 shows the L2 WFI timing for a 4-processor configuration.

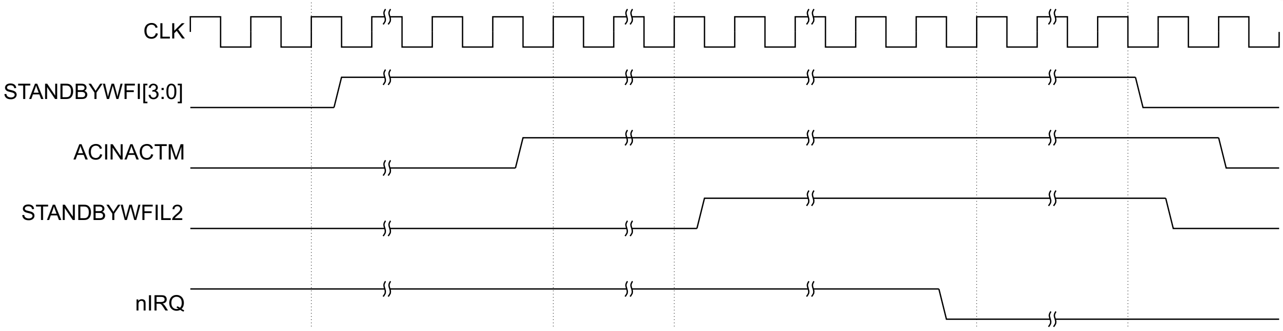


Figure 2-5 L2 Wait For Interrupt timing

Individual processor shutdown mode

This is the mode where the Vcore power domain for an individual processor is shut down and all state is lost.

To enable a processor to be powered down, the implementation must place the processor on a separately controlled power supply. In addition, you must clamp the outputs of the processor to benign values while the entire processor is powered down, to indicate that the processor is idle.

To power down the processor, apply the following sequence:

1. Clear the SCTLR.C bit, or HSCTLR.C bit if in Hyp mode, to prevent further data cache allocation.
2. Clean and invalidate all data from the L1 data cache. The L2 duplicate snoop tag RAM for this processor is now empty. This prevents any new data cache snoops or data cache maintenance operations from other processors in the multiprocessor device being issued to this processor.
3. Execute a `CLREX` instruction.
4. Switch the processor from *Symmetric Multiprocessing* (SMP) mode to *Asymmetric Multiprocessing* (AMP) mode by clearing the ACTLR.SMP bit. Clearing the SMP bit enables the processor to be taken out of coherency by preventing the processor from receiving cache or TLB maintenance operations broadcast by other processors in the multiprocessor device.
5. Execute an `ISB` instruction to ensure that all of the CP15 register changes from the previous steps have been committed.
6. Execute a `DSB` instruction to ensure that all cache, TLB and branch predictor maintenance operations issued by any processor in the multiprocessor device before the SMP bit was cleared have completed.
7. Execute a WFI instruction and wait until the **STANDBYWFI** output is asserted to indicate that the processor is in idle and low power state.
8. Deassert **DBGPWRDUP** LOW. This prevents any external debug access to the processor.

   Note

   The **DBGPWRDUP** signal is an integration layer signal. See the *Cortex-A7 MPCore Integration Manual* for more information.

9. Activate the processor output clamps.

10. Remove power from the Vcore power domain.

To power up the processor, apply the following sequence:

1. Assert **nCOREPORESET** LOW and hold **L1RSTDISABLE** LOW. Ensure **DBGPWRDUP** is held LOW to prevent any external debug access to the processor.
2. Apply power to the Vcore power domain. Keep the state of the signals **nCOREPORESET**, **L1RSTDISABLE** and **DBGPWRDUP** LOW.
3. Release the processor output clamps.
4. De-assert resets.
5. Assert **DBGPWRDUP** HIGH to allow external debug access to the processor.
6. If required use software to restore the state of the processor prior to power-down.
7. Assert ACTLR.SMP bit HIGH for SMP mode. Continue a normal power-on reset sequence.

Note

The **DBGPWRDUP** signal is an integration layer signal. See the Cortex-A7 MPCore Integration Manual for more information.

Multiprocessor device shutdown mode

This is the mode where the Vscu, Vscu_ram, and Vcore power domains are shut down and all state is lost. To power down the multiprocessor device, apply the following sequence:

1. Ensure all non-lead processors are in shutdown mode, see *Individual processor shutdown mode* on page 2-17.
2. Follow steps 1. to 3. in *Individual processor shutdown mode* on page 2-17.
3. Clean and invalidate all data from L2 data cache.
4. Follow steps 4 to 9. in *Individual processor shutdown mode* on page 2-17.
5. Assert **ACINACTM** and wait until the **STANDBYWFIL2** output is asserted to indicate that the L2 memory system is idle.
6. Activate the multiprocessor device output clamps.
7. Remove power from the Vscu and Vscu_ram power domains.

Note

For device power-down, all operations on a lead processor must occur after the equivalent step on all non-lead processors.

To power up the multiprocessor device, apply the following sequence:

1. For each processor in the multiprocessor device, assert **nCOREPORESET** LOW and hold **L1RSTDISABLE** LOW.
2. For the lead processor in the multiprocessor device, assert **nL2RESET** LOW and hold **L2RSTDISABLE** LOW.
3. Apply power to the Vscu, Vscu_ram, and Vcore domains while keeping the signals described in steps 1. and 2. LOW.

4. Release the processor output clamps.

5. Continue a normal power-on reset sequence.

6. For each processor in the multiprocessor device, set the ACTLR.SMP bit to 1 for SMP mode.

   Note

   You must ensure the ACTLR.SMP bit is set to 1 before the caches and MMU are enabled, or any cache and TLB maintenance operations are performed. The only time this bit is set to 0 is during a processor power-down sequence.

Dormant mode

Optionally, the Dormant mode is supported in the multiprocessor device. In this mode all the processors and L2 control logic are powered down while the L2 cache RAMs are powered up and retain state. The RAM blocks that remain powered up during Dormant mode are:

- L2 tag RAMs
- L2 data RAMs.

To support Dormant mode, you must ensure:

- That the L2 cache RAMs are in a separate power domain.
- To clamp all inputs to the L2 cache RAMs to benign values. This avoids corrupting data when the processors and L2 control power domains enter and exit power down state.

Before entering Dormant mode the architectural state of the multiprocessor device, excluding the contents of the L2 cache RAMs that remain powered up, must be saved to external memory.

To exit from Dormant mode to Run mode, the SoC must perform a full power-on reset sequence. The SoC must assert the reset signals until power is restored. After power is restored, the processor exits the power-on reset sequence, and the architectural state must be restored.

To enter Dormant mode, apply the following sequence:

1. Clear the SCTLR C bit to prevent further data cache allocation.

2. Clean and invalidate all data from the L1 data cache. The L2 duplicate snoop tag RAM for this processor is now empty. This prevents any new data cache snoops or data cache maintenance operations from other processors in the multiprocessor device being issued to this processor.

3. Execute a `CLREX` instruction.

4. Switch the processor from SMP mode to AMP mode by clearing the ACTLR.SMP bit. Clearing the SMP bit enables the processor to be taken out of coherency by preventing the processor from receiving cache or TLB maintenance operations broadcast by other processors in the multiprocessor device.

5. Save architectural state, if required. These state saving operations must ensure that the following occur:
   - all ARM registers, including the CPSR and SPSR, are saved
   - all system registers are saved
   - all debug related state is saved.

6. Execute an `ISB` instruction to ensure that all of the CP15 register changes from the previous steps have been committed.

7. Execute a `DSB` instruction to ensure that all cache, TLB and branch predictor maintenance operations issued by any processor in the multiprocessor device before the SMP bit was cleared have completed. In addition, this ensures that all state saving has completed.

8. Execute a `WFI` instruction and wait until the **STANDBYWFI** output is asserted, to indicate that the processor is in idle and low power state.

9. Repeat the previous steps for all processors, and wait for all **STANDBYWFI** outputs to be asserted.

10. The SoC asserts the input pin **ACINACTM** to idle the AXI master interface after all responses are received and before it sends any new transactions on the interface. When the L2 has completed the outstanding transactions for the AXI master and slave interfaces, **STANDBYWFIL2** is asserted to indicate that L2 memory system is idle.

11. When all processors **STANDBYWFI** and **STANDBYWFIL2** are asserted, the multiprocessor device is ready to enter Dormant mode.

12. Activate the L2 cache RAM input clamps.

13. Remove power from the Vcore and Vscu power domains.

To exit Dormant mode, apply the following sequence:

1. Apply a normal power-on reset sequence. You must apply resets to the processors and the L2 memory system logic until power is restored. During this reset sequence, **L2RSTDISABLE** must be held HIGH to disable the L2 cache hardware reset mechanism.

2. When power has been restored, release the L2 cache RAM input clamps.

3. Continue a normal power-on reset sequence with **L2RSTDISABLE** held HIGH.

4. The architectural state must be restored, if required.

2.4.3 Event communication using WFE or SEV

An external agent can use the the **EVENTI** pin to participate in a WFE or SEV event communication of the Cortex-A7 MPCore processor. When this pin is asserted, it sends an event message to all the Cortex-A7 MPCore processors in the cluster. This is similar to executing a `SEV` instruction on one processor in the cluster. This enables the external agent to signal to the processors that it has released a semaphore and that the processors can leave the WFE standby power saving mode. The **EVENTI** input pin must remain HIGH at least one **CLK** clock cycle to be visible by the processors.

The external agent can determine that at least one of the Cortex-A7 MPCore processors in the cluster has executed an `SEV` instruction by checking the **EVENTO** pin. When `SEV` is executed by any of the processors in the cluster, an event is signaled to all the processors in the cluster, and the **EVENTO** pin is asserted. This pin is asserted HIGH for one **CLK** clock cycle when any Cortex-A7 MPCore processor in the cluster executes an `SEV` instruction.

2.4.4 Communication to the Power Management Controller

Communication between the Cortex-A7 MPCore processor and the system Power Management Controller can be performed using the **STANDBYWFI[n]** or **STANDBYWFE[n]** signals.

Chapter 3
Programmers Model

This chapter describes the processor registers and provides information for programming the processor. It contains the following sections:

3.1 About the programmers model

The Cortex-A7 MPCore processor implements the ARMv7-A architecture. This includes:

- the 32-bit ARM instruction set
- the Thumb instruction set that has both 16-bit and 32-bit instructions
- the ThumbEE instruction set
- a trivial implementation of the Jazelle Extension
- the Advanced SIMD and VFP Extensions
- the Security Extensions
- the Virtualization Extensions
- the Large Physical Address Extension
- the Multiprocessing Extensions.

See the *ARM Architecture Reference Manual* for more information.

3.2 ThumbEE architecture

The *Thumb Execution Environment* (ThumbEE) extension is a variant of the Thumb instruction set that is designed as a target for dynamically generated code.

See the *ARM Architecture Reference Manual* for information.

3.3 Jazelle Extension

The Cortex-A7 MPCore processor provides a trivial implementation of the Jazelle Extension. This means that the processor does not accelerate the execution of any bytecodes.

In the trivial implementation of the Jazelle Extension:

- Jazelle state is not supported
- the `BXJ` instruction behaves as a `BX` instruction.

See the *ARM Architecture Reference Manual* for information.

3.3.1 Register summary

Table 3-1 gives a summary of the Cortex-A7 Jazelle Extension registers.

Table 3-1 Summary of Jazelle Extension registers

CRn	Op1	CRm	Op2	Name	Reset	Description
c0	7	c0	0	JIDR	`0x00000000`	*Jazelle Identity Register*
c1	7	c0	0	JOSCR	`0x00000000`	*Jazelle OS Control Register* on page 3-5
c2	7	c0	0	JMCR	`0x00000000`	*Jazelle Main Configuration Register* on page 3-5

3.3.2 Register description

This section describes the Cortex-A7 Jazelle Extension registers. Table 3-1 provides cross references to individual register bits.

Jazelle Identity Register

The JIDR characteristics are:

Purpose	Enables software to determine the implementation of the Jazelle Extension provided by the processor.
Usage constraints	The JIDR is: • a read-only register • accessible from all privilege levels.
Configurations	Available in all configurations.
Attributes	See the register summary in Table 3-1.

Figure 3-1 shows the JIDR bit assignments.

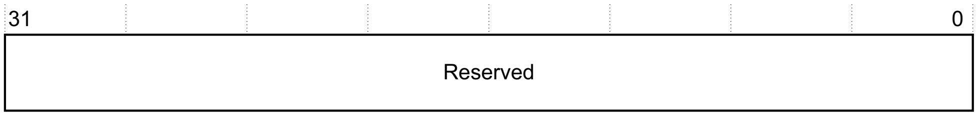


Figure 3-1 JIDR Register bit assignments

Table 3-2 shows the JIDR bit assignments.

Table 3-2 JIDR Register bit assignments

Bits	Name	Function
[31:0]	-	Reserved, RAZ

To access the JIDR, read the CP14 register with:

```
MRC p14, 7, <Rd>, c0, c0, 0; Read Jazelle Identity Register
```

Jazelle OS Control Register

The JOSCR characteristics are:

Purpose Provides operating system control of the use of the Jazelle Extension.

Usage constraints The JOSCR is:

- a read/write register
- accessible only from PL1 or higher.

Configurations Available in all configurations.

Attributes See the register summary in Table 3-1 on page 3-4.

Table 3-3 shows the JOSCR bit assignments.

Table 3-3 JOSCR Register bit assignments

Bits	Name	Function
[31:0]	-	Reserved, RAZ/WI

To access the JOSCR, read or write the CP14 register with:

```
MRC p14, 7, <Rd>, c1, c0, 0; Read Jazelle OS Control Register
MCR p14, 7, <Rd>, c1, c0, 0; Write Jazelle OS Control Register
```

Jazelle Main Configuration Register

The JMCR characteristics are:

Purpose Provides control of the Jazelle Extension features.

Usage constraints The JMCR is:

- a read/write register, with access rights that depend on the current privilege level:
  - write-only in unprivileged level
  - read-write at PL1 or higher.

Configurations Available in all configurations.

Attributes See the register summary in Table 3-1 on page 3-4.

Table 3-4 shows the JMCR bit assignments.

Table 3-4 JMCR Register bit assignments

Bits	Name	Function
[31:0]	-	Reserved, RAZ/WI

To access the JMCR, read or write the CP14 register with:

```
MRC p14, 7, <Rd>, c2, c0, 0; Read Jazelle Main Configuration Register
MCR p14, 7, <Rd>, c2, c0, 0; Write Jazelle Main Configuration Register
```

3.4 Advanced SIMD and VFP Extensions

The Advanced SIMD extension is a media and signal processing architecture that adds instructions targeted primarily at audio, video, 3-D graphics, image, and speech processing.

The VFP extension performs single-precision and double-precision floating-point operations.

Note

The Advanced SIMD architecture extension, its associated implementations, and supporting software, are commonly referred to as NEON.

All Advanced SIMD instructions and VFP instructions are available in both ARM and Thumb states.

See the *ARM Architecture Reference Manual* for more information.

See the *Cortex-A7 MPCore Floating-Point Unit Technical Reference Manual* and *Cortex-A7 MPCore NEON Media Processing Engine Technical Reference Manual* for implementation-specific information.

3.5 Security Extensions architecture

The Security Extensions architecture facilitates the development of secure applications. This section describes the following:

- *System boot sequence*
- *Security Extensions write access disable*.

See the *ARM Architecture Reference Manual* for more information.

3.5.1 System boot sequence

Caution

Security Extensions computing enable a secure software environment. The technology does not protect the processor from hardware attacks, and you must make sure that the hardware containing the boot code is appropriately secure.

The processor always boots in the privileged Supervisor mode in the Secure state, with the NS bit set to 0. See *Secure Configuration Register* on page 4-63. This means that code that does not attempt to use the Security Extensions always runs in the Secure state. If the software uses both Secure and Non-secure states, the less trusted software, such as a complex operating system and application code running under that operating system, executes in Non-secure state, and the most trusted software executes in the Secure state.

The following sequence is expected to be typical use of the Security Extensions:

1. Exit from reset in Secure state.
2. Configure the security state of memory and peripherals. Some memory and peripherals are accessible only to the software running in Secure state.
3. Initialize the secure operating system. The required operations depend on the operating system, and include initialization of caches, MMU, exception vectors, and stacks.
4. Initialize Secure Monitor software to handle exceptions that switch execution between the Secure and Non-secure operating systems.
5. Optionally lock aspects of the secure state environment against further configuration.
6. Pass control through the Secure Monitor software to the non-secure OS with an `SMC` instruction.
7. Enable the Non-secure operating system to initialize. The required operations depend on the operating system, and typically include initialization of caches, MMU, exception vectors, and stacks.

The overall security of the software depends on the system design, and on the secure software itself.

3.5.2 Security Extensions write access disable

The processor pin **CP15SDISABLE** disables write access to certain registers in the CP15 System Control Coprocessor. Attempts to write to these registers when **CP15SDISABLE** is HIGH result in an Undefined instruction exception. Reads from the registers are still permitted.

A change to the **CP15SDISABLE** pin takes effect on the instructions decoded by the processor as quickly as practically possible. Software must perform an `ISB` instruction, after a change to this pin on the boundary of the macrocell, to ensure that its effect is recognized for following instructions. It it is expected that:

- control of the **CP15SDISABLE** pin remains within the SoC that embodies the macrocell
- the **CP15SDISABLE** pin is driven LOW by the SoC hardware at reset.

You can use the **CP15SDISABLE** pin to disable subsequent access to the system control processor registers after the Secure boot code runs. This protects the configuration set up by the Secure boot code.

3.6 Virtualization Extensions architecture

The Virtualization Extensions are a set of features that provide hardware support for virtualizing the Non-secure state of an ARM VMSAv7 implementation. This supports system use of a virtual machine monitor, known as the hypervisor, to switch guest operating systems.

The Virtualization Extensions require implementation of the Security Extensions and the Large *Physical Address Extension* (LPAE).

The Virtualization Extensions also require implementation of:

- the v7.1 Debug architecture, see Chapter 10 *Debug*
- the PMUv2 Performance Monitors, see Chapter 11 *Performance Monitoring Unit*.

See the *ARM Architecture Reference Manual* for more information.

3.7 Large Physical Address Extension architecture

The *Large Physical Address Extension* (LPAE) is an extension to the *Virtual Memory System Architecture* (VMSAv7) that provides an address translation system that supports physical addresses of up to 40 bits.

See the *ARM Architecture Reference Manual* for more information.

3.8 Multiprocessing Extensions

The Multiprocessing Extensions are an extension to the ARMv7-A profile, that provides a set of features that enhance multiprocessing functionality.

See the *ARM Architecture Reference Manual* for more information.

3.9 Modes of operation

The processor has the following instruction set operating states controlled by the T bit and J bit in the CPSR.

ARM state	The processor executes 32-bit, word-aligned ARM instructions.
Thumb state	The processor executes 16-bit and 32-bit, halfword-aligned Thumb instructions.
ThumbEE state	The processor executes a variant of the Thumb instruction set designed as a target for dynamically generated code. This is code compiled on the device either shortly before or during execution from a portable bytecode or other intermediate or native representation.

The J bit and the T bit determine the instruction set used by the processor. Table 3-5 shows the encoding of these bits.

Table 3-5 CPSR J and T bit encoding

J	T	Instruction set state
0	0	ARM
0	1	Thumb
1	1	ThumbEE

Note

- The processor does not support Jazelle state. This means there is no processor state where the J bit is 1 and T bit is 0.
- Transition between ARM and Thumb states does not affect the processor mode or the register contents. See the *ARM Architecture Reference Manual* for information on entering and exiting ThumbEE state.

3.10 Memory model

The Cortex-A7 MPCore processor views memory as a linear collection of bytes numbered in ascending order from zero. For example, bytes 0-3 hold the first stored word, and bytes 4-7 hold the second stored word.

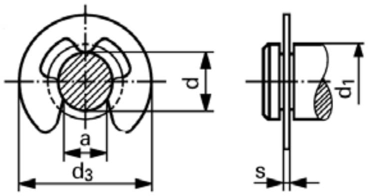
The processor can store words in memory as either:

- big-endian format
- little-endian format.

See the *ARM Architecture Reference Manual* for more information about big-endian and little-endian memory systems.

Note

Instructions are always treated as little-endian.



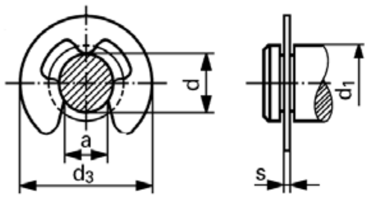
Circlips for shafts, spring steel

DIN 6799

d	d3	a	s	d1	black		zinc plated	
ø groove h11	fitted	± IT10	± 0,03	shaft				
quantity					100	1000+	100	1000+
1,2	3,25	1,01	0,3	1,4-2	4.50	1.95	5.70	2.40
1,5	4,25	1,28	0,4	2-2,5	4.50	1.45	5.70	1.95
1,9	4,8	1,61	0,5	2,5-3	4.50	1.55	5.70	1.95
2,3	6,6	1,94	0,6	3- 4	4.50	1.55	5.70	2.–
3,2	7,3	2,70	0,6	4- 5	4.50	1.60	5.70	2.–
4	9,3	3,34	0,7	5- 7	4.50	1.80	5.70	2.–
5	11,3	4,11	0,7	6- 8	4.50	1.95	5.70	2.–
6	12,3	5,26	0,7	7- 9	4.50	2.–	5.70	2.20
7	14,3	5,84	0,9	8-11	6.–	2.10	7.20	3.75
8	16,3	6,52	1	9-12	6.–	2.70	7.50	5.10
9	18,8	7,63	1,1	10-14	6.30	4.50	8.70	6.40
10	20,4	8,32	1,2	11-15	7.20	5.70	10.90	7.90
12	23,4	10,45	1,3	13-18	8.90	7.20	12.40	10.20
15	29,4	12,61	1,5	16-24	15.50	10.40	25.50	18.80
19	37,6	15,92	1,75	20-31	33.50	25.75	51.–	40.50
24	44,6	21,88	2	25-38	45.50	36.–	65.50	53.50
30	52,6	25,8	2,5	32-42	96.50	79.–		

Zinc plated
spring steel:

+ 12% discount for 10 000 pieces per item



Circlips for shafts, spring steel

DIN 6799 stacked

(O:2)

d	d3	a	s	d1	black		
ø groove h11	fitted	± IT10	± 0,03	shaft			
quantity					1000	10 000	
1,2	3,25	1,01	0,3	1,4-2	1.95	1.75	
1,5	4,25	1,28	0,4	2- 2,5	1.55	1.35	
1,9	4,8	1,61	0,5	2,5-3	1.55	1.35	
2,3	6,6	1,94	0,6	3- 4	1.50	1.15	
3,2	7,3	2,70	0,6	4- 5	1.60	1.25	
4	9,3	3,34	0,7	5- 7	2.–	1.75	
5	11,3	4,11	0,7	6- 8	2.10	1.85	
6	12,3	5,26	0,7	7- 9	2.30	2.–	
7	14,3	5,84	0,9	8-11	2.90	2.55	
8	16,3	6,52	1	9-12	4.30	3.80	
9	18,8	7,63	1,1	10-14	5.30	4.65	
10	20,4	8,32	1,2	11-15	6.90	6.10	
12	23,4	10,45	1,3	13-18	10.90	9.60	

Applicators: Beneri (O:Z) Price per piece

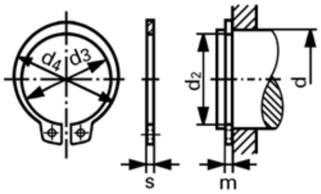
for	quantity	1	10
d =	1,2 / 1,5 / 1,9 / 2,3 / 3,2 / 4 / 5 / 6 / 7 / 8 / 9 10 / 12	39.– 114.–	30.– 91.–

Dispenser: Beneri (O:Z) Price per piece

for each size	310.–	

Retaining rings for shafts, spring steel standard type

DIN 471



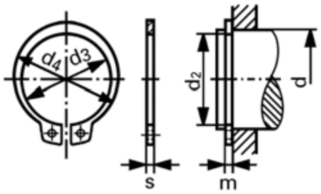
shaft		fitted		shaft groove		black		dacromet plated	
d	d3	d4	s	d2	m				
quantity			- 0,06	h11	H13	100	1000	100	1000
3	2,7	7	0,4	2,8	0,5	5.10	2.40		
4	3,7	8,6	0,4	3,8	0,5	4.–	1.50	6.–	3.45
5	4,7	10,3	0,6	4,8	0,7	4.–	1.50	6.–	3.45
6	5,6	11,7	0,7	5,7	0,8	4.–	1.75	6.–	3.45
7	6,5	13,5	0,8	6,7	0,9	4.20	1.95	6.–	3.45
8	7,4	14,7	0,8	7,6	0,9	4.20	2.10	6.–	3.45
9	8,4	16	1	8,6	1,1	4.60	2.60	6.10	3.45
10	9,3	17	1	9,6	1,1	4.70	3.–	6.20	3.45
11	10,2	18	1	10,5	1,1	5.–	3.40	6.80	3.80
12	11	19	1	11,5	1,1	4.90	2.95	6.70	3.70
13	11,9	20,2	1	12,4	1,1	5.40	3.10	7.80	4.80
14	12,9	21,4	1	13,4	1,1	5.50	3.20	8.10	5.20
15	13,8	22,6	1	14,3	1,1	5.90	3.25	8.70	5.50
16	14,7	23,8	1	15,2	1,1	6.30	3.75	9.20	6.–
17	15,7	25	1	16,2	1,1	6.60	4.40	10.–	6.80
18	16,5	26,2	1,2	17	1,3	7.40	5.30	12.70	8.60
19	17,5	27,2	1,2	18	1,3	8.–	6.–	14.20	9.60
20	18,5	28,4	1,2	19	1,3	8.40	6.20	15.10	10.20
21	19,5	29,6	1,2	20	1,3	9.20	6.30	17.20	11.50
22	20,5	30,8	1,2	21	1,3	9.50	6.70	18.–	12.20
(23)	21,5	32,6	1,2	22	1,3	10.20	7.60	19.90	13.40
24	22,2	33,2	1,2	22,9	1,3	10.20	7.60	19.90	13.40
25	23,2	34,2	1,2	23,9	1,3	10.70	8.–	21.25	14.50
26	24,2	35,5	1,2	24,9	1,3	12.–	9.20	23.50	15.80
(27)	24,9	37,2	1,2	25,6	1,3	12.90	9.90	27.75	18.70
28	25,9	37,9	1,5	26,6	1,6	16.10	12.40	29.75	21.–
29	26,9	39,1	1,5	27,6	1,6	17.10	13.10	32.–	22.50
30	27,9	40,5	1,5	28,6	1,6	17.20	13.20	32.50	22.75
(31)	28,6	41,5	1,5	29,3	1,6	23.50	18.–	39.25	27.50
32	29,6	43	1,5	30,3	1,6	20.25	15.50	38.60	26.70
(33)	30,5	44,4	1,5	31,3	1,6	24.75	19.–	41.50	29.25
34	31,5	45,4	1,5	32,3	1,6	25.–	19.20	41.50	29.25
35	32,2	46,8	1,5	33	1,6	26.50	20.50	42.50	30.–
36	33,2	47,8	1,75	34	1,85	33.25	27.–	52.50	36.50
(37)	34,2	49,4	1,75	35	1,85	34.25	28.–	57.–	40.–
38	35,2	50,2	1,75	36	1,85	34.25	28.–	57.–	40.–

shaft		fitted		shaft groove		black		dacromet plated	
d	d3	d4	s	d2	m				
quantity			- 0,10	h12	H13	100	1000	100	1000
(39)	36	51,3	1,75	37	1,85	34.25	28.–	59.–	41.–
40	36,5	52,6	1,75	37,5	1,85	35.50	29.25	59.50	42.–
(41)	37,5	54,4	1,75	38,5	1,85	38.–	30.50	69.–	48.50
42	38,5	55,7	1,75	39,5	1,85	36.75	29.50	69.–	48.50
(44)	40,5	58,5	1,75	41,5	1,85	42.–	33.75	74.50	54.–
45	41,5	59,1	1,75	42,5	1,85	42.–	33.75	74.50	54.–
(46)	42,5	60,5	1,75	43,5	1,85	47.50	37.75	78.50	57.50
(47)	43,5	61,6	1,75	44,5	1,85	47.50	37.75	79.50	58.–
48	44,5	62,5	1,75	45,5	1,85	42.50	34.–	79.50	58.–
50	45,8	64,5	2	47	2,15	42.50	34.–	86.50	63.–
52	47,8	66,7	2	49	2,15	54.–	43.–	94.–	68.50
(54)	49,8	69,5	2	51	2,15	57.–	46.–	99.–	72.50
55	50,8	70,2	2	52	2,15	56.–	44.50	97.–	71.–
56	51,8	71,6	2	53	2,15	62.50	52.50	108.–	79.–
(57)	52,8	72,6	2	54	2,15	67.–	55.50	115.–	84.–
58	53,8	73,6	2	55	2,15	67.–	55.50	115.–	84.–
60	55,8	75,6	2	57	2,15	62.–	49.50	109.–	79.50
62	57,8	77,8	2	59	2,15	68.50	59.50	121.–	89.–
63	58,8	79	2	60	2,15	70.50	61.–	132.–	96.50
65	60,8	81,4	2,5	62	2,65	77.–	64.–	147.–	112.–
(67)	62,5	83,7	2,5	64	2,65	94.50	79.–	174.–	133.–
68	63,5	84,8	2,5	65	2,65	94.50	79.–	174.–	133.–
70	65,5	87	2,5	67	2,65	91.–	76.–	168.–	128.–
72	67,5	89,2	2,5	69	2,65	96.–	80.–	177.–	135.–
75	70,5	92,7	2,5	72	2,65	96.–	80.–	184.–	140.–
(77)	72,5	95,3	2,5	74	2,65	111.–	92.–	206.–	157.–
78	73,5	96,1	2,5	75	2,65	111.–	93.–	206.–	157.–
80	74,5	98,1	2,5	76,5	2,65	101.–	84.–	203.–	155.–
82	76,5	100,3	2,5	78,5	2,65	112.–	93.50	216.–	162.–
85	79,5	103,3	3	81,5	3,15	130.–	108.–	256.–	171.–
(87)	81,5	105,8	3	83,5	3,15	158.–	132.–	294.–	178.–
88	82,5	106,5	3	84,5	3,15	159.–	133.–	294.–	178.–
90	84,5	108,5	3	86,5	3,15	151.–	131.–	300.–	181.–
(92)	86,5	111	3	88,5	3,15	171.–	143.–	326.–	200.–
95	89,5	114,8	3	91,5	3,15	158.–	138.–		
(97)	91,5	117,3	3	93,5	3,15	202.–	161.–		
(98)	92,5	118,3	3	94,5	3,15	202.–	161.–		
100	94,5	120,2	3	96,5	3,15	165.–	126.–		
(102)	95	120,8	4	98	4,15	318.–	260.–		
105	98	125,8	4	101	4,15	318.–	260.–		
(107)	100	128,3	4	103	4,15	318.–	260.–		
(108)	101	129,5	4	104	4,15	318.–			
110	103	131,2	4	106	4,15	302.–			
(112)	105	133,7	4	108	4,15	330.–			

shaft		fitted		shaft groove		black	
d	d3	d4	s	d2	m		
quantity			- 0,12	h13	H13	100	
115	108	137,3	4	111	4,15	330.–	
(117)	110	140,2	4	113	4,15	330.–	
(118)	111	141,4	4	114	4,15	330.–	
120	113	143,1	4	116	4,15	316.–	
122	115	145,6	4	118	4,15	376.–	
125	118	149	4	121	4,15	330.–	
(127)	120	151,5	4	123	4,15	420.–	
(128)	121	152,7	4	124	4,15	420.–	
130	123	154,4	4	126	4,15	360.–	
(132)	125	156,9	4	128	4,15	450.–	
135	128	159,8	4	131	4,15	390.–	
(137)	130	162,2	4	133	4,15	480.–	
(138)	131	163,4	4	134	4,15	480.–	
140	133	165,2	4	136	4,15	405.–	
(142)	135	167,6	4	138	4,15	510.–	
145	138	170,6	4	141	4,15	420.–	
(147)	140	173	4	143	4,15	540.–	
(148)	141	174,2	4	144	4,15	540.–	
150	142	177,3	4	145	4,15	435.–	
155	146	182,3	4	150	4,15	540.–	
160	151	188	4	155	4,15	570.–	
165	155,5	193,4	4	160	4,15	630.–	
170	160,5	198,4	4	165	4,15	870.–	
175	165,5	203,4	4	170	4,15	1020.–	
180	170,5	210	4	175	4,15	1050.–	
185	175,5	215	4	180	4,15	1080.–	
190	180,5	220	4	185	4,15	1110.–	
195	185,5	225	4	190	4,15	1120.–	
200	190,5	230	4	195	4,15	1140.–	
(205)	195	235	5	200	4,15	2450.–	
210	198	240	5	204	5,15	3080.–	
(215)	203	245	5	209	5,15	3490.–	
220	208	250	5	214	5,15	2970.–	
(225)	213	255	5	219	5,15	3190.–	
230	218	260	5	224	5,15	3420.–	
(235)	223	265	5	229	5,15	3450.–	
240	228	270	5	234	5,15	3470.–	
(245)	233	275	5	239	5,15	3580.–	
250	238	280	5	244	5,15	3690.–	
260	245	294	5	252	5,15	4000.–	
270	255	304	5	262	5,15	4180.–	
280	265	314	5	272	5,15	4310.–	
290	275	324	5	282	5,15	5060.–	
300	285	334	5	292	5,15	4650.–	

Retaining rings for shafts, spring steel

DIN 471 heavy type

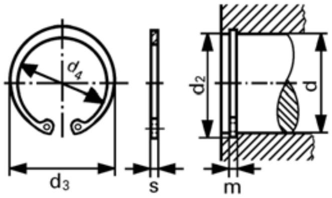


shaft		fitted		shaft groove		black		
d	d3	d4	s*	d2	m			
quantity			- 0,10	h11	H13	100	1000	
15	13,8	25,1	1,5	14,3	1,6	29.–	25.20	
16	14,7	26,5	1,5	15,2	1,6	24.–	20.25	
17	15,7	27,5	1,5	16,2	1,6	25.25	21.50	
18	16,5	28,7	1,5	17	1,6	28.75	24.50	
20	18,5	31,6	1,75	19	1,85	29.25	24.75	
22	20,5	34,6	1,75	21	1,85	35.50	31.10	
24	22,2	37,3	1,75	22,9	1,85	48.–	39.75	
25	23,2	38,5	2	23,9	2,15	50.50	42.–	
28	25,9	41,7	2	26,6	2,15	53.50	45.–	
30	27,9	43,7	2	28,6	2,15	55.50	47.50	
32	29,6	45,7	2	30,3	2,15	68.50		
34	31,5	47,9	2,5	32,3	2,65	79.10		
35	32,2	49,1	2,5	33	2,65	109.–		
40	36,5	54,7	2,5	37,5	2,65	114.–		
45	41,5	60,8	2,5	42,5	2,65	161.–		
50	45,8	66,8	3	47	3,15	270.–		
55	50,8	72,9	3	52	3,15	286.–		
60	55,8	78,9	3	57	3,15	358.–		
70	65,5	90	4	67	4,15	425.–		
80	74,5	100,6	4	76,5	4,15	760.–		
90	84,5	111,5	4	86,5	4,15	810.–		
100	94,5	122,1	4	96,5	4,15	865.–		

* for that size note thickness «s»

Retaining rings for bores, spring steel standard type

DIN 472



bore plated		fitted		bore groove		black		dacromet	
d	d3	d4	s	d2	m	100	1000	100	1000
8	8,7	3	0,8	8,4	0,9	4.20	2.60	5.70	3.45
9	9,8	3,7	0,8	9,4	0,9	4.20	2.60	5.70	3.45
10	10,8	3,3	1	10,4	1,1	4.50	2.60	5.70	3.45
11	11,8	4,1	1	11,4	1,1	5.20	3.65	5.80	3.90
12	13	4,9	1	12,5	1,1	5.20	3.65	5.90	3.90
13	14,1	5,4	1	13,6	1,1	5.20	3.65	6.10	4.–
14	15,1	6,2	1	14,6	1,1	5.20	3.65	6.60	4.10
15	16,2	7,2	1	15,7	1,1	5.20	3.65	6.70	4.30
16	17,3	8	1	16,8	1,1	5.20	3.65	7.10	4.70
17	18,3	8,8	1	17,8	1,1	5.40	3.70	7.50	5.20
18	19,5	9,4	1	19	1,1	5.80	3.85	8.10	5.60
19	20,5	10,4	1	20	1,1	6.20	4.40	9.20	6.10
20	21,5	11,2	1	21	1,1	6.20	4.40	9.50	6.40
21	22,5	12,2	1	22	1,1	6.90	4.60	11.30	7.70
22	23,5	13,2	1	23	1,1	7.40	5.–	11.90	8.–
(23)	24,6	13,6	1	24,1	1,3	8.80	6.60	15.30	10.30
24	25,9	14,8	1,2	25,2	1,3	8.80	6.60	16.10	10.90
25	26,9	15,5	1,2	26,2	1,3	9.–	7.–	17.–	11.40
26	27,9	16,1	1,2	27,2	1,3	10.–	7.70	18.50	12.50
(27)	29,1	16,6	1,2	28,4	1,3	10.50	8.10	19.70	13.30
28	30,1	17,9	1,2	29,4	1,3	11.90	9.20	20.50	13.80
(29)	31,1	18,4	1,2	30,4	1,3	13.–	10.–	22.50	15.20
30	32,1	19,9	1,2	31,4	1,3	13.30	10.50	23.50	15.90
31	33,4	20	1,2	32,7	1,3	14.50	11.10	24.50	16.60
32	34,4	20,6	1,2	33,7	1,3	15.10	11.60	25.–	17.–
(33)	35,5	21,6	1,2	34,7	1,3	16.30	12.60	30.75	21.50
34	36,5	22,6	1,5	35,7	1,6	18.30	14.–	33.25	23.25
35	37,8	23,6	1,5	37	1,6	19.20	14.80	34.50	24.25
36	38,8	24,6	1,5	38	1,6	20.–	15.40	36.50	25.50
37	39,8	25,4	1,5	39	1,6	22.50	17.40	38.75	27.25
38	40,8	26,4	1,5	40	1,6	23.50	18.–	40.50	28.50
(39)	42	27,4	1,5	41	1,6	25.75	19.90	45.–	31.75
40	43,5	27,8	1,75	42,5	1,85	28.50	23.–	48.50	34.25
(41)	44,5	28,4	1,75	43,5	1,85	33.75	27.–	54.–	37.75
42	45,5	29,9	1,75	44,5	1,85	34.40	27.80	55.–	38.75

bore plated		fitted		bore groove		black		dacromet	
d	d3	d4	s	d2	m	100	1000	100	1000
(44)	47,5	30,9	1,75	46,5	1,85	35.75	28.50	60.50	42.50
45	48,5	32	1,75	47,5	1,85	32.50	26.–	58.50	41.–
(46)	49,5	32,6	1,75	48,5	1,85	36.–	28.75	62.–	43.50
47	50,5	33,5	1,75	49,5	1,85	33.75	27.–	60.–	42.–
48	51,5	34,5	1,75	50,5	1,85	38.75	31.–	67.50	47.50
50	54,2	36,3	2	53	2,15	39.20	32.50	68.–	48.–
(51)	55,2	36,9	2	54	2,15	43.–	34.25	75.50	55.–
52	56,2	37,9	2	55	2,15	39.–	31.–	71.50	52.–
(53)	57,2	38,5	2	56	2,15	43.–	34.75	79.–	57.50
(54)	58,2	39,1	2	57	2,15	43.–	34.75	79.50	58.–
55	59,2	40,7	2	58	2,15	41.50	33.25	78.–	56.50
56	60,2	41,7	2	59	2,15	51.50	41.50	82.50	60.50
(57)	61,2	42,3	2	60	2,15	52.50	42.–	89.50	65.50
58	62,2	43,5	2	61	2,15	53.50	42.50	91.50	67.–
60	64,2	44,7	2	63	2,15	48.–	38.25	90.–	66.–
62	66,2	46,7	2	65	2,15	49.–	39.25	90.50	66.–
63	67,2	47,7	2	66	2,15	55.–	44.–	101.–	74.–
65	69,2	49	2,5	68	2,65	65.–	54.–	118.–	86.50
(67)	71,5	50,6	2,5	70	2,65	70.50	59.–	132.–	96.50
68	72,5	51,6	2,5	71	2,65	70.50	59.–	136.–	99.50
70	74,5	53,6	2,5	73	2,65	72.–	60.–	137.–	101.–
72	76,5	55,6	2,5	75	2,65	80.–	67.–	137.–	104.–
75	79,5	58,6	2,5	78	2,65	82.50	68.50	141.–	108.–
(77)	81,5	59,6	2,5	80	2,65	93.–	77.50	156.–	119.–
78	82,5	60,1	2,5	81	2,65	93.–	78.–	156.–	119.–
80	85,5	62,1	2,5	83,5	2,65	94.–	79.–	157.–	120.–
82	87,5	64,1	2,5	85,5	2,65	95.50	79.50	166.–	126.–
85	90,5	66,9	3	88,5	3,15	113.–	94.–	184.–	127.–
(87)	92,5	68,3	3	90,5	3,15	134.–	112.–	208.–	128.–
88	93,5	69,9	3	91,5	3,15	134.–	112.–	208.–	128.–
90	95,5	71,9	3	93,5	3,15	119.–	99.–	202.–	119.–
92	97,5	73,7	3	95,5	3,15	130.–	108.–	207.–	126.–
95	100,5	76,5	3	98,5	3,15	132.–	109.–	235.–	141.–
(97)	102,5	78	3	100,5	3,15	154.–	134.–	262.–	160.–
98	103,5	79	3	101,5	3,15	156.–	135.–		
100	105,5	80,6	3	103,5	3,15	133.–	116.–	252.–	158.–
102	108	82	4	106	4,15	239.–	208.–		
105	112	85	4	109	4,15	224.–	194.–		
(107)	114	86,5	4	111	4,15	260.–	227.–		
108	115	88	4	112	4,15	264.–			
110	117	88,2	4	114	4,15	240.–			
112	119	90	4	116	4,15	266.–			
115	122	93	4	119	4,15	246.–			
(117)	124	94,5	4	121	4,15	274.–			
(118)	125	95,2	4	122	4,15	276.–			
120	127	96,9	4	124	4,15	316.–			

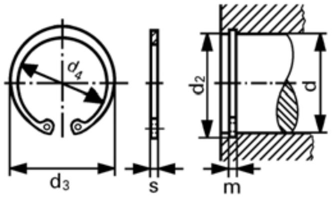
Return to chapter index | Return to main fastener catalog index | Return to Maryland Metrics home page

bore		fitted		bore groove		black	
d	d3	d4	s	d2	m		
quantity			- 0,12	H13	H13	100	
(122)	129	100,7	4	126	4,15	376.–	
125	132	101,9	4	129	4,15	330.–	
(127)	134	103,4	4	131	4,15	420.–	
(128)	135	104,1	4	133	4,15	420.–	
130	137	106,9	4	134	4,15	360.–	
(132)	139	108,4	4	136	4,15	450.–	
135	142	111,5	4	139	4,15	390.–	
(137)	144	113	4	141	4,15	480.–	
(138)	145	113,7	4	142	4,15	480.–	
140	147	116,5	4	144	4,15	405.–	
(142)	149	118	4	146	4,15	510.–	
145	152	121	4	149	4,15	420.–	
(147)	154	122,5	4	151	4,15	540.–	
(148)	155	123,2	4	152	4,15	540.–	
150	158	124,8	4	155	4,15	435.–	
155	164	129,8	4	160	4,15	540.–	
160	169	132,7	4	165	4,15	570.–	
165	174,5	137,7	4	170	4,15	630.–	
170	179,5	141,6	4	175	4,15	870.–	
175	184,5	146,6	4	180	4,15	1020.–	
180	189,5	150,2	4	185	4,15	1050.–	
185	194,5	155,2	4	190	4,15	1080.–	
190	199,5	160,2	4	195	4,15	1110.–	
195	204,5	165,2	4	200	4,15	1120.–	
200	209,5	170,2	4	205	4,15	1140.–	
210	222	180,2	5	216	5,15	3080.–	
(215)	227	185,2	5	221	5,15	3490.–	
220	232	190,2	5	226	5,15	2970.–	
(225)	237	195,2	5	231	5,15	3190.–	
230	242	200,2	5	236	5,15	3420.–	
(235)	247	205,2	5	241	5,15	3450.–	
240	252	210,2	5	246	5,15	3470.–	
(245)	257	215,2	5	251	5,15	3580.–	
250	262	220,2	5	256	5,15	3690.–	
260	275	226	5	268	5,15	4000.–	
265	280	231	5	273	5,15	4115.–	
270	285	236	5	278	5,15	4180.–	
275	290	241	5	283	5,15	4226.–	
280	295	246	5	288	5,15	4310.–	
290	305	256	5	298	5,15	5060.–	
300	315	266	5	308	5,15	4650.–	
310	327	273	6	320	6,20	15340.–	
320	337	278	6	330	6,20	15600.–	

Return to chapter index | Return to main fastener catalog index | Return to Maryland Metrics home page

Retaining rings for bores, spring steel

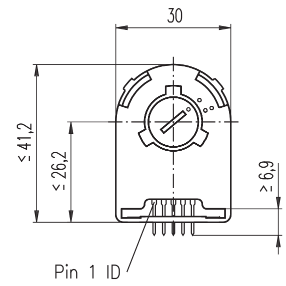
DIN 472 heavy type

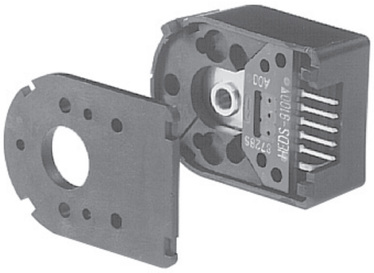


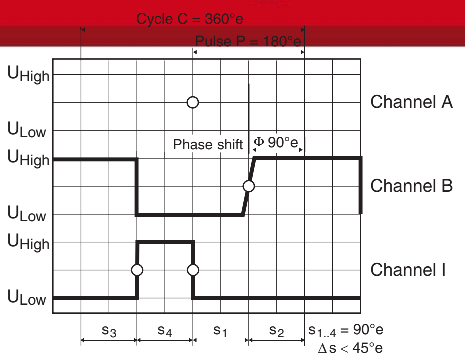
bore		fitted		bore groove		black		
d	d3	d4	s*	d2	m			
quantity			- 0,10	H12	H13	100	1000	
20	21,5	10,5	1,5	21	1,6	25.–	19.80	
22	23,5	12,1	1,5	23	1,6	27.60	21.05	
25	26,9	14,5	1,5	26,2	1,6	29.–	24.–	
26	27,9	15,3	1,5	27,2	1,6	42.–	35.50	
28	30,1	16,9	1,5	29,4	1,6	34.75	29.75	
30	32,1	18,4	1,5	31,4	1,6	38.25	32.50	
32	34,4	20	1,5	33,,7	1,6	50.–	41.50	
35	37,8	22,4	1,75	37	1,85	75.–	61.–	
40	43,5	26,3	2	42,5	2,15	100.–	80.–	
42	45,5	27,9	2	44,5	2,15	102.–		
45	48,5	30,3	2	47,5	2,15	110.–		
47	50,5	31,9	2	49,5	2,15	123.–		
50	54,2	34,2	2,5	53	2,65	136.–		
55	59,2	38,2	2,5	58	2,65	152.–		
60	64,2	42,1	3	63	3,15	182.–		
72	76,5	52,7	3	75	3,15	352.–		
75	79,5	55,5	3	78	3,15	380.–		

* for that size note Thickness «s»

Encoder HEDS 5540 500 Counts per turn, 3 Channels







Direction of rotation cw (definition cw p. 70)

🟥 Stock program
☐ Standard program
▨ Special program (on request)

Part Numbers

Type	110511	110513	110515
Counts per turn	500	500	500
Number of channels	3	3	3
Max. operating frequency (kHz)	100	100	100
Max. speed (rpm)	12 000	12 000	12 000
Shaft diameter (mm)	3	4	6

maxon Modular System

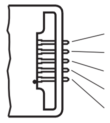
+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see Gearhead		
RE 25	99/101					75.3		
RE 25	99/101	GP 26, 0.75 - 2.0 Nm	257			●		
RE 25	99/101	GP 32, 0.75 - 6.0 Nm	259-263			●		
RE 25	99/101	KD 32, 1.0 - 4.5 Nm	268			●		
RE 25	99/101	GP 32 S	286-288			●		
RE 25, 20 W	101			AB 28	348	105.8		
RE 25, 20 W	101	GP 26, 0.75 - 2.0 Nm	257	AB 28	348	●		
RE 25, 20 W	101	GP 32, 0.75 - 6.0 Nm	259-263	AB 28	348	●		
RE 25, 20 W	101	KD 32, 1.0 - 4.5 Nm	268	AB 28	348	●		
RE 25, 20 W	101	GP 32 S	286-288	AB 28	348	●		
RE 30, 15 W	102					88.8		
RE 30, 15 W	102	GP 32, 0.75 - 4.5 Nm	261			●		
RE 30, 60 W	103					88.8		
RE 30, 60 W	103	GP 32, 0.75 - 6.0 Nm	259-265			●		
RE 30, 60 W	103	KD 32, 1.0 - 4.5 Nm	268			●		
RE 30, 60 W	103	GP 32 S	286-288			●		
RE 35, 90 W	104						91.7	
RE 35, 90 W	104	GP 32, 0.75 - 8.0 Nm	259-266				●	
RE 35, 90 W	104	GP 42, 3.0 - 15 Nm	270				●	
RE 35, 90 W	104	GP 32 S	286-288				●	
RE 35, 90 W	104			AB 28	348	124.3		
RE 35, 90 W	104	GP 32, 0.75 - 8.0 Nm	259-266	AB 28	348	●		
RE 35, 90 W	104	GP 42, 3.0 - 15 Nm	270	AB 28	348	●		
RE 35, 90 W	104	GP 32 S	286-288	AB 28	348	●		
RE 40, 150 W	105						91.7	
RE 40, 150 W	105	GP 42, 3.0 - 15 Nm	270				●	
RE 40, 150 W	105	GP 52, 4.0 - 30 Nm	273				●	
RE 40, 150 W	105			AB 28	348	124.3		
RE 40, 150 W	105	GP 42, 3.0 - 15 Nm	270	AB 28	348	●		
RE 40, 150 W	105	GP 52, 4.0 - 30 Nm	273	AB 28	348	●		

Technical Data

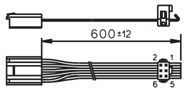
Supply voltage V_{CC}	5 V ± 10%
Output signal	TTL compatible
Phase shift Φ	90°e ± 45°e
Signal rise time (typically, at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	180 ns
Signal fall time (typically, at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	40 ns
Index pulse width (nominal)	90°e
Operating temperature range	-40…+100°C
Moment of inertia of code wheel	≤ 0.6 gcm^2
Max. angular acceleration	250 000 rad s^{-2}
Output current per channel	min. -1 mA, max. 5 mA

The index signal I is synchronised with channel A or B.

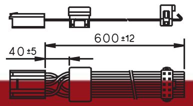
Pin Allocation



Encoder	Description	Pin no. from 3409.506
Pin 5	Channel B	1
Pin 4	V_{CC}	2
Pin 3	Channel A	3
Pin 2	Channel I	4
Pin 1	GND	5

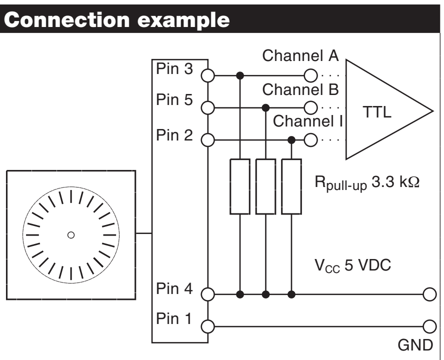


Cable with plug:
maxon Art. Nr. 3409.506
The plug (Harting 918.906.6803) can be fixed in the required position.



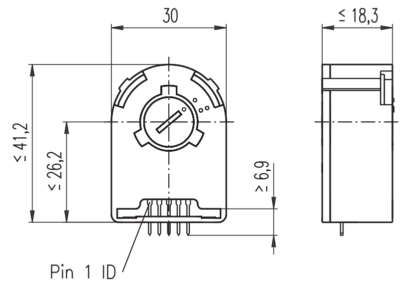
Cable with plug (compatible with encoder HEDS5010):
maxon Art. Nr. 3409.504
The plug (3M 89110-0101) can be fixed in the required position.

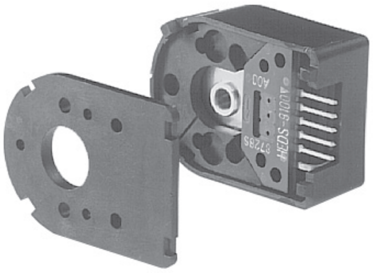
Connection example

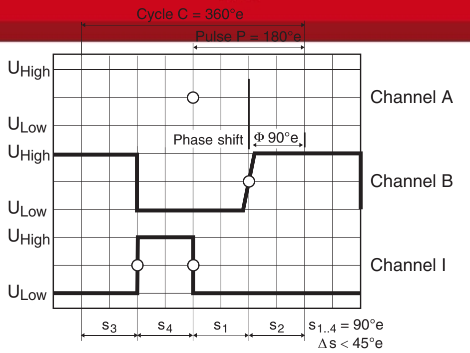


Ambient temperature range ϑ_U = 25°C

Encoder HEDS 5540 500 Counts per turn, 3 Channels





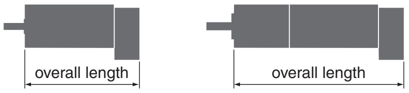


Direction of rotation cw (definition cw p. 70)

🟥 Stock program
☐ Standard program
▨ Special program (on request)

Part Numbers

Type	110511	110513	110515	110517
Counts per turn	500	500	500	500
Number of channels	3	3	3	3
Max. operating frequency (kHz)	100	100	100	100
Max. speed (rpm)	12 000	12 000	12 000	12 000
Shaft diameter (mm)	3	4	6	8



maxon Modular System

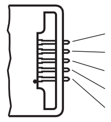
+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see Gearhead			
RE 25	100					63.8			
RE 25	100	GP 26, 0.75 - 2.0 Nm	257			●			
RE 25	100	GP 32, 0.75 - 4.5 Nm	259			●			
RE 25	100	GP 32, 0.75 - 6.0 Nm	260/263			●			
RE 25	100	KD 32, 1.0 - 4.5 Nm	268			●			
RE 25	100	GP 32 S	286-288			●			
RE 25, 20 W	100			AB 28	348	94.3			
RE 25, 20 W	100	GP 26, 0.75 - 2.0 Nm	257	AB 28	348	●			
RE 25, 20 W	100	GP 32, 0.75 - 4.5 Nm	259	AB 28	348	●			
RE 25, 20 W	100	GP 32, 0.75 - 6.0 Nm	260/263	AB 28	348	●			
RE 25, 20 W	100	KD 32, 1.0 - 4.5 Nm	268	AB 28	348	●			
RE 25, 20 W	100	GP 32 S	286-288	AB 28	348	●			
RE 50, 200 W	106								128.7
RE 50, 200 W	106	GP 52, 4 - 30 Nm	274						●
RE 50, 200 W	106	GP 62, 8 - 50 Nm	275						●
RE 65, 250 W	107								157.3
RE 65, 250 W	107	GP 81, 20 - 120 Nm	276						●
A-max 26	126-132					63.1			
A-max 26	126-132	GP 26, 0.75 - 4.5 Nm	257			●			
A-max 26	126-132	GS 30, 0.07 - 0.2 Nm	258			●			
A-max 26	126-132	GP 32, 0.75 - 4.5 Nm	259			●			
A-max 26	126-132	GP 32, 0.75 - 6.0 Nm	260/264			●			
A-max 26	126-132	GS 38, 0.1 - 0.6 Nm	269			●			
A-max 26	126-132	GP 32 S	286-288			●			
A-max 32	134/136						82.3		
A-max 32	134/136	GP 32, 0.75 - 6.0 Nm	259-265				●		
A-max 32	134/136	GS 38, 0.1 - 0.6 Nm	269				●		
A-max 32	134/136	GP 32 S	286-288				●		
EC 32, 80 W	180						78.4		
EC 32, 80 W	180	GP 32, 0.75 - 6.0 Nm	259-265				●		
EC 32, 80 W	180	GP 32 S	286-288				●		
EC 40, 170 W	181							103.4	
EC 40, 170 W	181	GP 42, 3.0 - 15 Nm	270					●	
EC 40, 170 W	181	GP 52, 4.0 - 30 Nm	273					●	

Technical Data

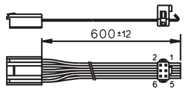
Supply voltage V_{CC}	5 V ± 10%
Output signal	TTL compatible
Phase shift Φ	90°e ± 45°e
Signal rise time (typically, at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	180 ns
Signal fall time (typically, at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	40 ns
Index pulse width	90°e
Operating temperature range	-40…+100°C
Moment of inertia of code wheel	≤ 0.6 gcm^2
Max. angular acceleration	250 000 rad s^{-2}
Output current per channel	min. -1 mA, max. 5 mA

The index signal I is synchronised with channel A or B.

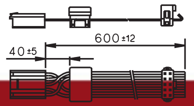
Pin Allocation



Encoder	Description	Pin no. from 3409.506
Pin 5	Channel B	1
Pin 4	V_{CC}	2
Pin 3	Channel A	3
Pin 2	Channel I	4
Pin 1	GND	5

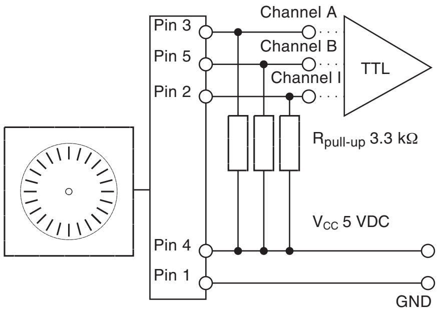


Cable with plug:
maxon Art. Nr. 3409.506
The plug (Harting 918.906.6803) can be fixed in the required position.



Cable with plug (compatible with encoder HEDS5010):
maxon Art. Nr. 3409.504
The plug (3M 89110-0101) can be fixed in the required position.

Connection example

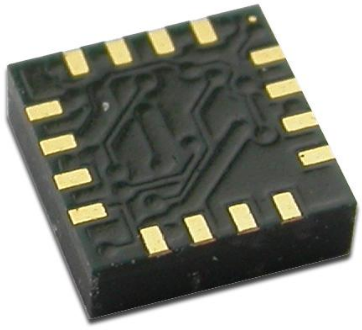


Ambient temperature range ϑ_U = 25°C

3-Axis Digital Compass IC
HMC5883L

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I^{2}C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magnetoresistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES	BENEFITS
3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package	Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly
12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ±8 Gauss Fields	Enables 1° to 2° Degree Compass Heading Accuracy
Built-In Self Test	Enables Low-Cost Functionality Test after Assembly in Production
Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 μA)	Compatible for Battery Powered Applications
Built-In Strap Drive Circuits	Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
I^{2}C Digital Interface	Popular Two-Wire Serial Data Interface for Consumer Electronics
Lead Free Package Construction	RoHS Compliance
Wide Magnetic Field Range (+/-8 Oe)	Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy
Software and Algorithm Support Available	Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available
Fast 160 Hz Maximum Output Rate	Enables Pedestrian Navigation and LBS Applications

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND VDDIO Referenced to DGND	2.16 1.71	2.5 1.8	3.6 VDD+0.1	Volts Volts
Average Current Draw	Idle Mode Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V (Dual Supply) VDD = VDDIO = 2.5V (Single Supply)	- -	2 100	- -	μA μA
Performance					
Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss
Digital Resolution	VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC	0.73		4.35	milli-gauss
Noise Floor (Field Resolution)	VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below)		2		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurment Mode Single Measurement Mode	0.75		75 160	Hz Hz
Measurement Period	From receiving command to data ready		6		ms
Turn-on Time	Ready for I2C commands Analog Circuit Ready for Measurements		200 50		μs ms
Gain Tolerance	All gain/dynamic range settings		±5		%
I^{2}C Address	8-bit read address 8-bit write address		0x3D 0x3C		hex hex
I^{2}C Rate	Controlled by I^{2}C Master			400	kHz
I^{2}C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		0.2*VDDIO 0.8*VDDIO		Volts Volts
Self Test	X & Y Axes Z Axis		±1.16 ±1.08		gauss
	X & Y & Z Axes (GN=5) Positive Bias X & Y & Z Axes (GN=5) Negative Bias	243 -575		575 -243	LSb
Sensitivity Tempco	T_A = -40 to 125°C, Uncompensated Output		-0.3		%/°C
General					
ESD Voltage	Human Body Model (all pins) Charged Device Model (all pins)			2000 750	Volts
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C

Characteristics	Conditions*	Min	Typ	Max	Units
Reflow Classification	MSL 3, 260 °C Peak Temperature				
Package Size	Length and Width	2.85	3.00	3.15	mm
Package Height		0.8	0.9	1.0	mm
Package Weight			18		mg

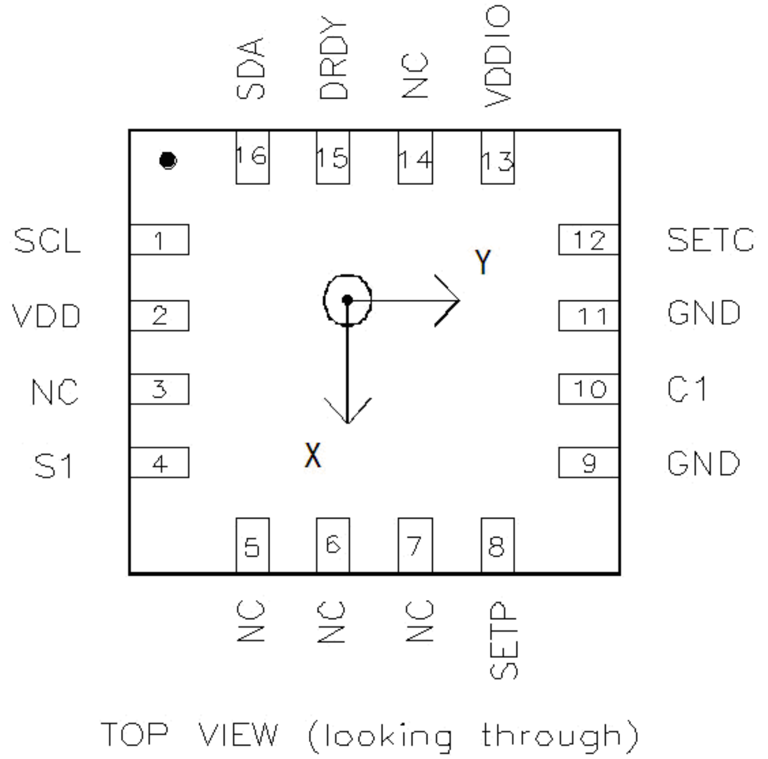
Absolute Maximum Ratings (* Tested at 25°C except stated otherwise.)

Characteristics	Min	Max	Units
Supply Voltage VDD	-0.3	4.8	Volts
Supply Voltage VDDIO	-0.3	4.8	Volts

PIN CONFIGURATIONS

Pin	Name	Description
1	SCL	Serial Clock – I^2C Master/Slave Clock
2	VDD	Power Supply (2.16V to 3.6V)
3	NC	Not to be Connected
4	S1	Tie to VDDIO
5	NC	Not to be Connected
6	NC	Not to be Connected
7	NC	Not to be Connected
8	SETP	Set/Reset Strap Positive – S/R Capacitor (C2) Connection
9	GND	Supply Ground
10	C1	Reservoir Capacitor (C1) Connection
11	GND	Supply Ground
12	SETC	S/R Capacitor (C2) Connection – Driver Side
13	VDDIO	IO Power Supply (1.71V to VDD)
14	NC	Not to be Connected
15	DRDY	Data Ready, Interrupt Pin. Internally pulled high. Optional connection. Low for 250 μsec when data is placed in the data output registers.
16	SDA	Serial Data – I^2C Master/Slave Data

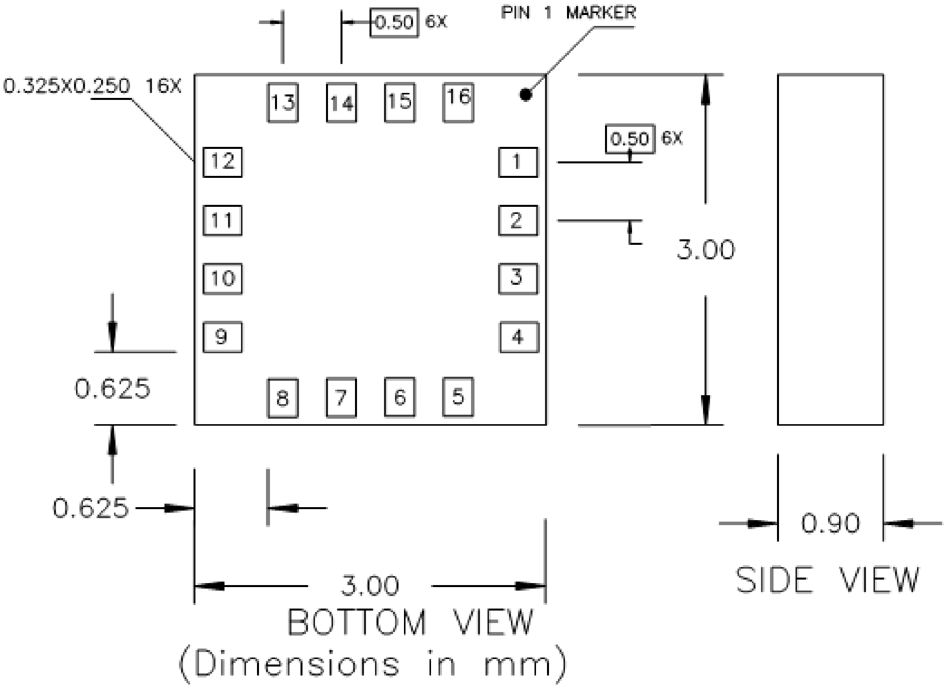
Table 1: Pin Configurations



Arrow indicates direction of magnetic field that generates a positive output reading in Normal Measurement configuration.

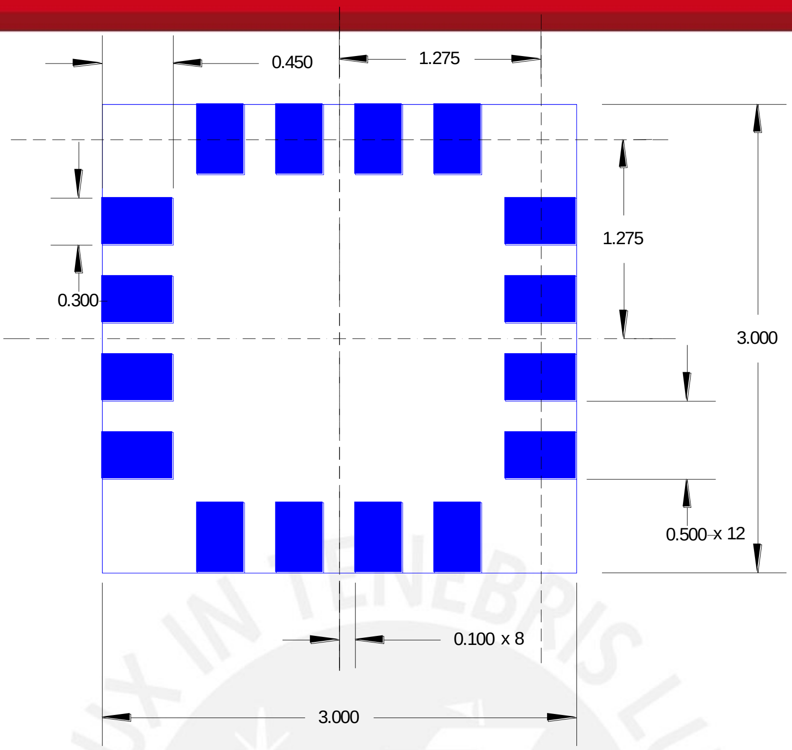
PACKAGE OUTLINES

PACKAGE DRAWING HMC5883L (16-PIN LPCC, dimensions in millimeters)



MOUNTING CONSIDERATIONS

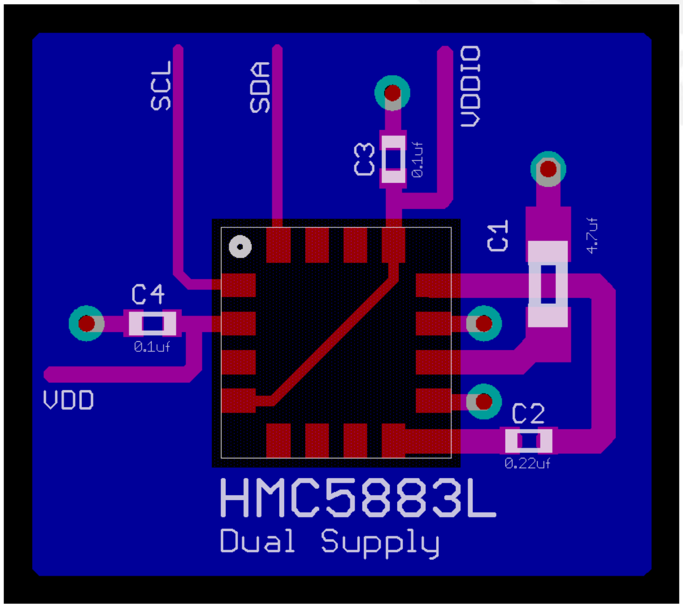
The following is the recommend printed circuit board (PCB) footprint for the HMC5883L.

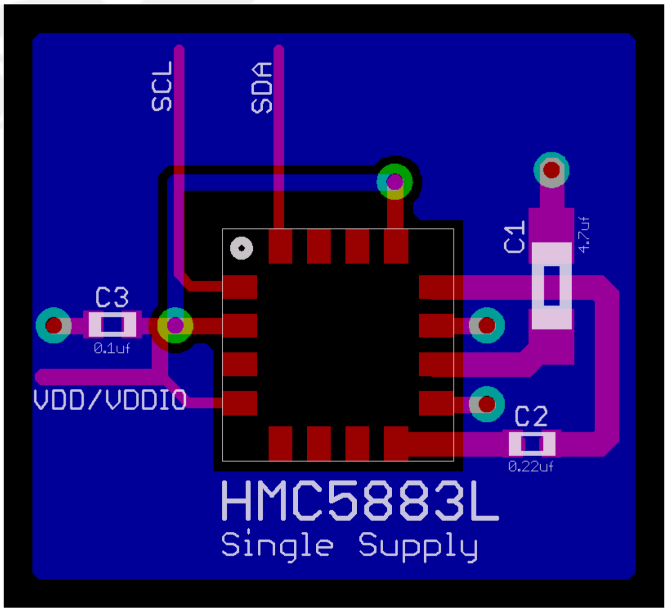


HMC5883 Land Pad Pattern
(All dimensions are in mm)

LAYOUT CONSIDERATIONS

Besides keeping all components that may contain ferrous materials (nickel, etc.) away from the sensor on both sides of the PCB, it is also recommended that there is no conducting copper under/near the sensor in any of the PCB layers. See recommended layout below. Notice that the one trace under the sensor in the dual supply mode is not expected to carry active current since it is for pin 4 pull-up to VDDIO. Power and ground planes are removed under the sensor to minimize possible source of magnetic noise. For best results, use non-ferrous materials for all exposed copper coding.





PCB Pad Definition and Traces

The HMC5883L is a fine pitch LCC package. Refer to previous figure for recommended PCB footprint for proper package centering. Size the traces between the HMC5883L and the external capacitors (C1 and C2) to handle the 1 ampere peak current pulses with low voltage drop on the traces.

Stencil Design and Solder Paste

A 4 mil stencil and 100% paste coverage is recommended for the electrical contact pads.

Reflow Assembly

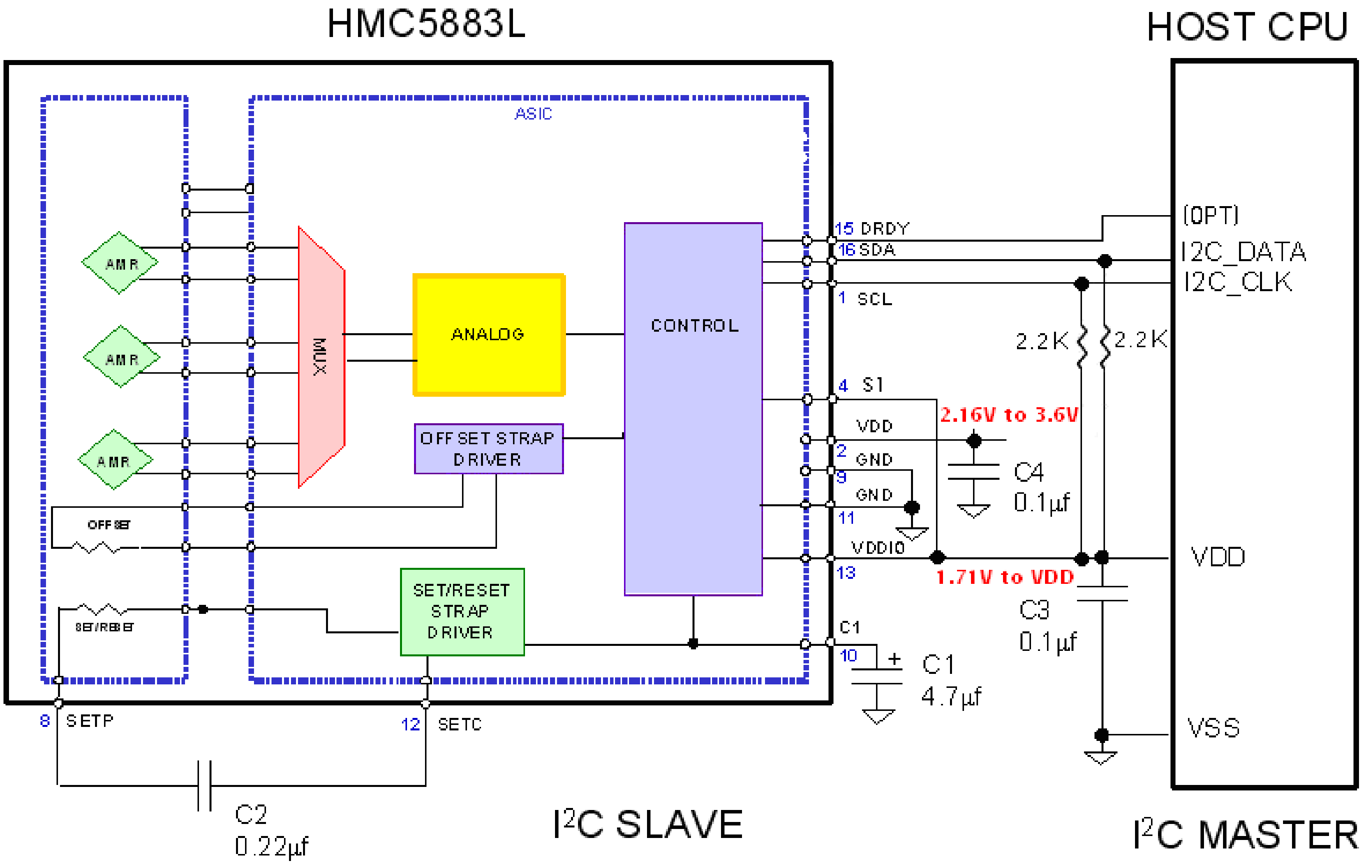
This device is classified as MSL 3 with 260°C peak reflow temperature. A baking process (125°C, 24 hrs) is required if device is not kept continuously in a dry (< 10% RH) environment before assembly. No special reflow profile is required for HMC5883L, which is compatible with lead eutectic and lead-free solder paste reflow profiles. Honeywell recommends adherence to solder paste manufacturer's guidelines. Hand soldering is not recommended. Built-in self test can be used to verify device functionalities after assembly.

External Capacitors

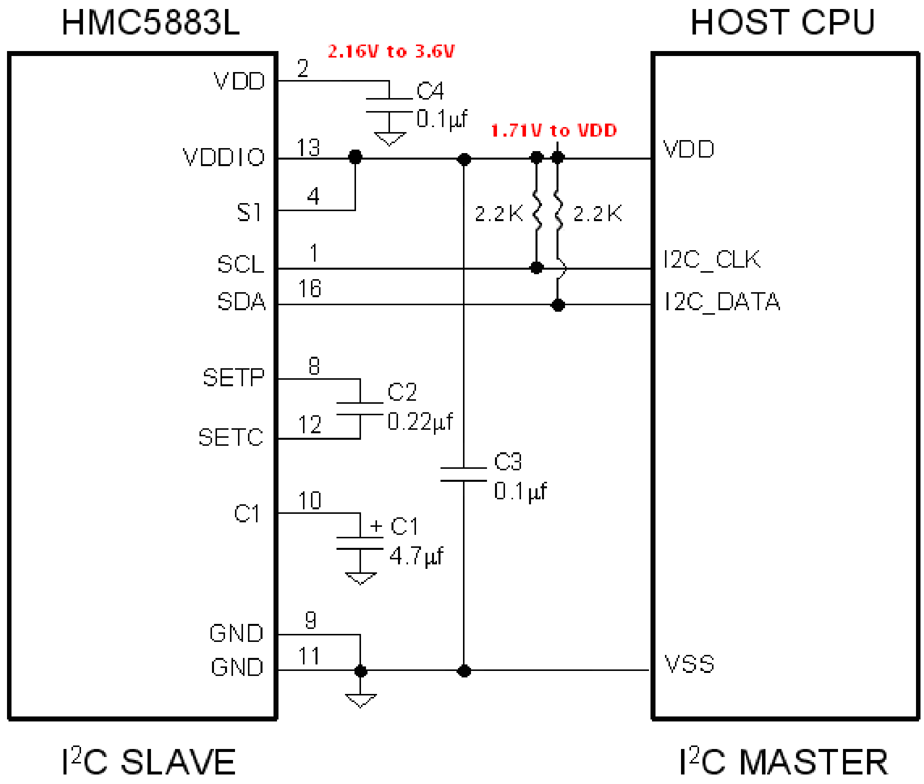
The two external capacitors should be ceramic type construction with low ESR characteristics. The exact ESR values are not critical but values less than 200 milli-ohms are recommended. Reservoir capacitor C1 is nominally 4.7 µF in capacitance, with the set/reset capacitor C2 nominally 0.22 µF in capacitance. Low ESR characteristics may not be in many small SMT ceramic capacitors (0402), so be prepared to up-size the capacitors to gain Low ESR characteristics.

INTERNAL SCHEMATIC DIAGRAM

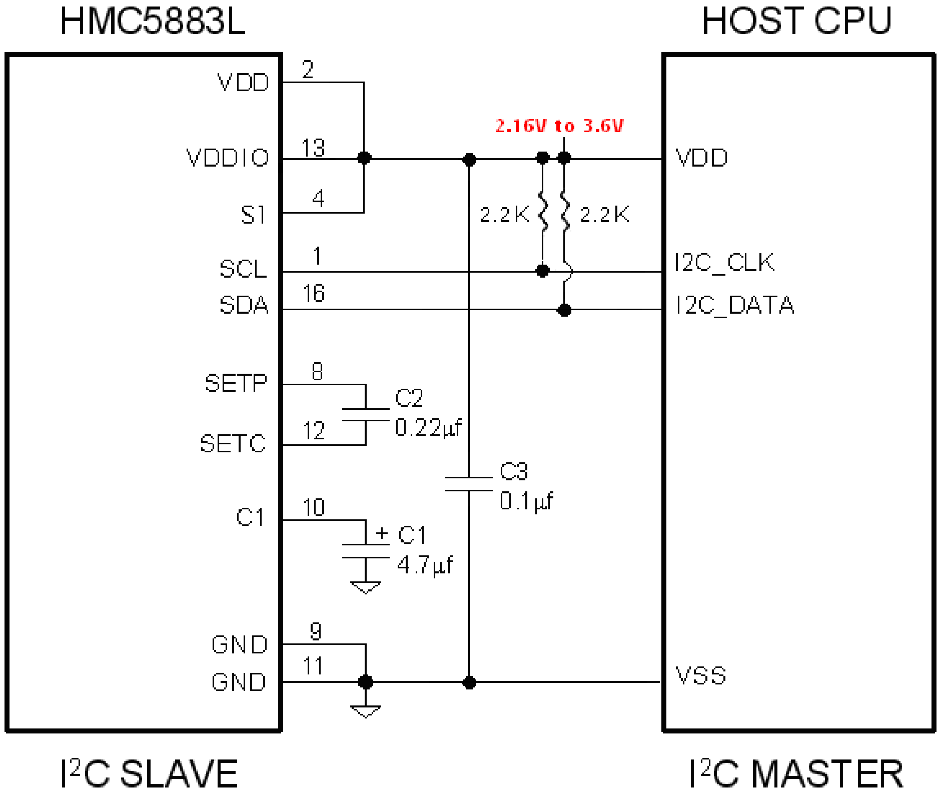
HMC5883L



DUAL SUPPLY REFERENCE DESIGN



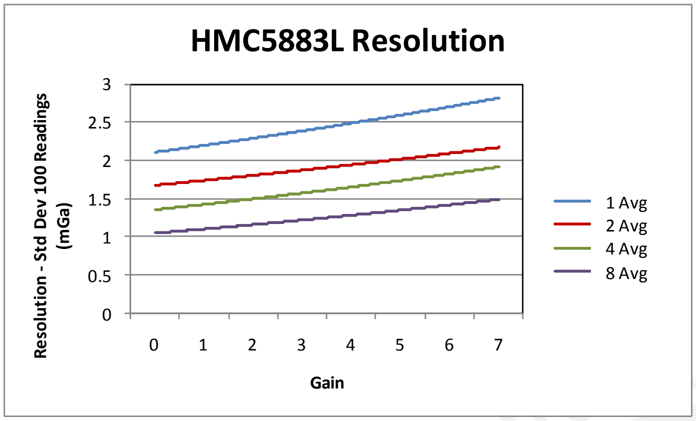
SINGLE SUPPLY REFERENCE DESIGN

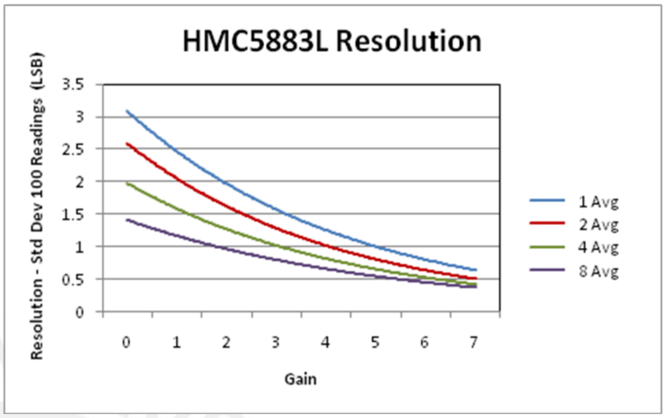


PERFORMANCE

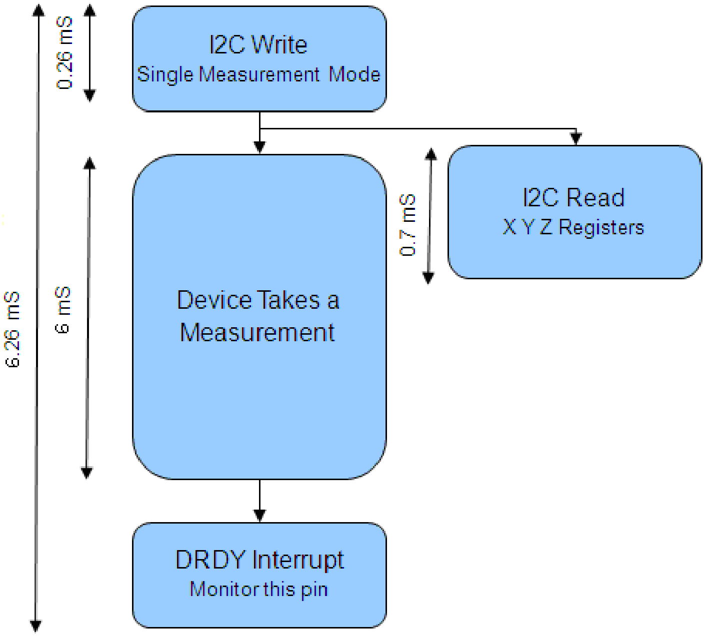
The following graph(s) highlight HMC5883L's performance.

Typical Noise Floor (Field Resolution)





Typical Measurement Period in Single-Measurement Mode



** Monitoring of the DRDY Interrupt pin is only required if maximum output rate is desired.*

BASIC DEVICE OPERATION

Anisotropic Magneto-Resistive Sensors

The Honeywell HMC5883L magnetoresistive sensor circuit is a trio of sensors and application specific support circuits to measure magnetic fields. With power supply applied, the sensor converts any incident magnetic field in the sensitive axis directions to a differential voltage output. The magnetoresistive sensors are made of a nickel-iron (Permalloy) thin-film and patterned as a resistive strip element. In the presence of a magnetic field, a change in the bridge resistive elements causes a corresponding change in voltage across the bridge outputs.

These resistive elements are aligned together to have a common sensitive axis (indicated by arrows in the pinout diagram) that will provide positive voltage change with magnetic fields increasing in the sensitive direction. Because the output is only proportional to the magnetic field component along its axis, additional sensor bridges are placed at orthogonal directions to permit accurate measurement of magnetic field in any orientation.

Self Test

To check the HMC5883L for proper operation, a self test feature in incorporated in which the sensor is internally excited with a nominal magnetic field (in either positive or negative bias configuration). This field is then measured and reported. This function is enabled and the polarity is set by bits MS[n] in the configuration register A. An internal current source generates DC current (about 10 mA) from the VDD supply. This DC current is applied to the offset straps of the magneto-resistive sensor, which creates an artificial magnetic field bias on the sensor. The difference of this measurement and the measurement of the ambient field will be put in the data output register for each of the three axes. By using this built-in function, the manufacturer can quickly verify the sensor's full functionality after the assembly without additional test setup. The self test results can also be used to estimate/compensate the sensor's sensitivity drift due to temperature.

For each "self test measurement", the ASIC:

1. Sends a "Set" pulse
2. Takes one measurement (M1)
3. Sends the (~10 mA) offset current to generate the (~1.1 Gauss) offset field and takes another measurement (M2)
4. Puts the difference of the two measurements in sensor's data output register:

   Output = [M2 – M1] (i.e. output = offset field only)

See SELF TEST OPERATION section later in this datasheet for additional details.

Power Management

This device has two different domains of power supply. The first one is VDD that is the power supply for internal operations and the second one is VDDIO that is dedicated to IO interface. It is possible to work with VDDIO equal to VDD; Single Supply mode, or with VDDIO lower than VDD allowing HMC5883L to be compatible with other devices on board.

I^{2}C Interface

Control of this device is carried out via the I^{2}C bus. This device will be connected to this bus as a slave device under the control of a master device, such as the processor.

This device is compliant with *I^{2}C-Bus Specification*, document number: 9398 393 40011. As an I^{2}C compatible device, this device has a 7-bit serial address and supports I^{2}C protocols. This device supports standard and fast modes, 100kHz and 400kHz, respectively, but does not support the high speed mode (Hs). External pull-up resistors are required to support these standard and fast speed modes.

Activities required by the master (register read and write) have priority over internal activities, such as the measurement. The purpose of this priority is to not keep the master waiting and the I^{2}C bus engaged for longer than necessary.

Internal Clock

The device has an internal clock for internal digital logic functions and timing management. This clock is not available to external usage.

H-Bridge for Set/Reset Strap Drive

The ASIC contains large switching FETs capable of delivering a large but brief pulse to the Set/Reset strap of the sensor. This strap is largely a resistive load. There is no need for an external Set/Reset circuit. The controlling of the Set/Reset function is done automatically by the ASIC for each measurement. One half of the difference from the measurements taken after a set pulse and after a reset pulse will be put in the data output register for each of the three axes. By doing so, the sensor's internal offset and its temperature dependence is removed/cancelled for all measurements. The set/reset pulses also effectively remove the past magnetic history (magnetism) in the sensor, if any.

For each "measurement", the ASIC:

1. Sends a "Set" pulse
2. Takes one measurement (Mset)
3. Sends a "Reset" pulse
4. Takes another measurement (Mreset)
5. Puts the following result in sensor's data output register:

Output = [Mset – Mreset] / 2

Charge Current Limit

The current that reservoir capacitor (C1) can draw when charging is limited for both single supply and dual supply configurations. This prevents drawing down the supply voltage (VDD).

MODES OF OPERATION

This device has several operating modes whose primary purpose is power management and is controlled by the Mode Register. This section describes these modes.

Continuous-Measurement Mode

During continuous-measurement mode, the device continuously makes measurements, at user selectable rate, and places measured data in data output registers. Data can be re-read from the data output registers if necessary; however, if the master does not ensure that the data register is accessed before the completion of the next measurement, the data output registers are updated with the new measurement. To conserve current between measurements, the device is placed in a state similar to idle mode, but the Mode Register is not changed to Idle Mode. That is, MD[n] bits are unchanged. Settings in the Configuration Register A affect the data output rate (bits DO[n]), the measurement configuration (bits MS[n]), when in continuous-measurement mode. All registers maintain values while in continuous-measurement mode. The I^2C bus is enabled for use by other devices on the network in while continuous-measurement mode.

Single-Measurement Mode

This is the default power-up mode. During single-measurement mode, the device makes a single measurement and places the measured data in data output registers. After the measurement is complete and output data registers are updated, the device is placed in idle mode, and the Mode Register *is* changed to idle mode by setting MD[n] bits. Settings in the configuration register affect the measurement configuration (bits MS[n])when in single-measurement mode. All registers maintain values while in single-measurement mode. The I^2C bus is enabled for use by other devices on the network while in single-measurement mode.

Idle Mode

During this mode the device is accessible through the I^2C bus, but major sources of power consumption are disabled, such as, but not limited to, the ADC, the amplifier, and the sensor bias current. All registers maintain values while in idle mode. The I^2C bus is enabled for use by other devices on the network while in idle mode.

REGISTERS

This device is controlled and configured via a number of on-chip registers, which are described in this section. In the following descriptions, *set* implies a logic 1, and *reset* or *clear* implies a logic 0, unless stated otherwise.

Register List

The table below lists the registers and their access. All address locations are 8 bits.

Address Location	Name	Access
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

Table2: Register List

Register Access

This section describes the process of reading from and writing to this device. The devices uses an address pointer to indicate which register location is to be read from or written to. These pointer locations are sent from the master to this slave device and succeed the 7-bit address (0x1E) plus 1 bit read/write identifier, i.e. 0x3D for read and 0x3C for write.

To minimize the communication between the master and this device, the address pointer updated automatically without master intervention. The register pointer will be incremented by 1 automatically after the current register has been read successfully.

The address pointer value itself cannot be read via the I^{2}C bus.
Any attempt to read an invalid address location returns 0's, and any write to an invalid address location or an undefined bit within a valid address location is ignored by this device.

To move the address pointer to a random register location, first issue a "write" to that register location with no data byte following the commend. For example, to move the address pointer to register 10, send 0x3C 0x0A.

Configuration Register A

The configuration register is used to configure the device for setting the data output rate and measurement configuration. CRA0 through CRA7 indicate bit locations, with *CRA* denoting the bits that are in the configuration register. CRA7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit.CRA default is 0x10.

CRA7	**CRA6**	**CRA5**	**CRA4**	**CRA3**	**CRA2**	**CRA1**	**CRA0**
(0)	MA1(0)	MA0(0)	DO2 (1)	DO1 (0)	DO0 (0)	MS1 (0)	MS0 (0)

Table 3: Configuration Register A

Location	**Name**	**Description**
CRA7	CRA7	Bit CRA7 is reserved for future function. Set to 0 when configuring CRA.
CRA6 to CRA5	MA1 to MA0	Select number of samples averaged (1 to 8) per measurement output. 00 = 1(Default); 01 = 2; 10 = 4; 11 = 8
CRA4 to CRA2	DO2 to DO0	Data Output Rate Bits. These bits set the rate at which data is written to all three data output registers.
CRA1 to CRA0	MS1 to MS0	Measurement Configuration Bits. These bits define the measurement flow of the device, specifically whether or not to incorporate an applied bias into the measurement.

Table 4: Configuration Register A Bit Designations

The Table below shows all selectable output rates in continuous measurement mode. All three channels shall be measured within a given output rate. Other output rates with maximum rate of 160 Hz can be achieved by monitoring DRDY interrupt pin in single measurement mode.

DO2	**DO1**	**DO0**	**Typical Data Output Rate (Hz)**
0	0	0	0.75
0	0	1	1.5
0	1	0	3
0	1	1	7.5
1	0	0	15 (Default)
1	0	1	30
1	1	0	75
1	1	1	Reserved

Table 5: Data Output Rates

MS1	**MS0**	**Measurement Mode**
0	0	Normal measurement configuration (Default). In normal measurement configuration the device follows normal measurement flow. The positive and negative pins of the resistive load are left floating and high impedance.
0	1	Positive bias configuration for X, Y, and Z axes. In this configuration, a positive current is forced across the resistive load for all three axes.
1	0	Negative bias configuration for X, Y and Z axes. In this configuration, a negative current is forced across the resistive load for all three axes..
1	1	This configuration is reserved.

Table 6: Measurement Modes

Configuration Register B

The configuration register B for setting the device gain. CRB0 through CRB7 indicate bit locations, with *CRB* denoting the bits that are in the configuration register. CRB7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. CRB default is 0x20.

CRB7	CRB6	CRB5	CRB4	CRB3	CRB2	CRB1	CRB0
GN2 (0)	GN1 (0)	GN0 (1)	(0)	(0)	(0)	(0)	(0)

Table 7: Configuration B Register

Location	Name	Description
CRB7 to CRB5	GN2 to GN0	Gain Configuration Bits. These bits configure the gain for the device. The gain configuration is common for all channels.
CRB4 to CRB0	0	These bits must be cleared for correct operation.

Table 8: Configuration Register B Bit Designations

The table below shows nominal gain settings. Use the “Gain” column to convert counts to Gauss. The “Digital Resolution” column is the theoretical value in term of milli-Gauss per count (LSb) which is the inverse of the values in the “Gain” column. The effective resolution of the usable signal also depends on the noise floor of the system, i.e.

Effective Resolution = Max (Digital Resolution, Noise Floor)

Choose a lower gain value (higher GN#) when total field strength causes overflow in one of the data output registers (saturation). Note that the very first measurement after a gain change maintains the same gain as the previous setting. **The new gain setting is effective from the second measurement and on.**

GN2	GN1	GN0	Recommended Sensor Field Range	Gain (LSb/ Gauss)	Digital Resolution (mG/LSb)	Output Range
0	0	0	± 0.88 Ga	1370	0.73	0xF800–0x07FF (-2048–2047)
0	0	1	± 1.3 Ga	1090 (default)	0.92	0xF800–0x07FF (-2048–2047)
0	1	0	± 1.9 Ga	820	1.22	0xF800–0x07FF (-2048–2047)
0	1	1	± 2.5 Ga	660	1.52	0xF800–0x07FF (-2048–2047)
1	0	0	± 4.0 Ga	440	2.27	0xF800–0x07FF (-2048–2047)
1	0	1	± 4.7 Ga	390	2.56	0xF800–0x07FF (-2048–2047)
1	1	0	± 5.6 Ga	330	3.03	0xF800–0x07FF (-2048–2047)
1	1	1	± 8.1 Ga	230	4.35	0xF800–0x07FF (-2048–2047)

Table 9: Gain Settings

Mode Register

The mode register is an 8-bit register from which data can be read or to which data can be written. This register is used to select the operating mode of the device. MR0 through MR7 indicate bit locations, with *MR* denoting the bits that are in the mode register. MR7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. Mode register default is 0x01.

MR7	**MR6**	**MR5**	**MR4**	**MR3**	**MR2**	**MR1**	**MR0**
HS(0)	(0)	(0)	(0)	(0)	(0)	MD1 (0)	MD0 (1)

Table 10: Mode Register

Location	**Name**	**Description**
MR7 to MR2	HS	Set this pin to enable High Speed I2C, 3400kHz.
MR1 to MR0	MD1 to MD0	Mode Select Bits. These bits select the operation mode of this device.

Table 11: Mode Register Bit Designations

MD1	**MD0**	**Operating Mode**
0	0	Continuous-Measurement Mode. In continuous-measurement mode, the device continuously performs measurements and places the result in the data register. RDY goes high when new data is placed in all three registers. After a power-on or a write to the mode or configuration register, the first measurement set is available from all three data output registers after a period of $2/f_{DO}$ and subsequent measurements are available at a frequency of f_{DO}, where f_{DO} is the frequency of data output.
0	1	Single-Measurement Mode (Default). When single-measurement mode is selected, device performs a single measurement, sets RDY high and returned to idle mode. Mode register returns to idle mode bit values. The measurement remains in the data output register and RDY remains high until the data output register is read or another measurement is performed.
1	0	Idle Mode. Device is placed in idle mode.
1	1	Idle Mode. Device is placed in idle mode.

Table 12: Operating Modes

Data Output X Registers A and B

The data output X registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel X. Data output X register A contains the MSB from the measurement result, and data output X register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DXRA0 through DXRA7 and DXRB0 through DXRB7 indicate bit locations, with *DXRA* and *DXRB* denoting the bits that are in the data output X registers. DXRA7 and DXRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DXRA7	**DXRA6**	**DXRA5**	**DXRA4**	**DXRA3**	**DXRA2**	**DXRA1**	**DXRA0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DXRB7	**DXRB6**	**DXRB5**	**DXRB4**	**DXRB3**	**DXRB2**	**DXRB1**	**DXRB0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 13: Data Output X Registers A and B

Data Output Y Registers A and B

The data output Y registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel Y. Data output Y register A contains the MSB from the measurement result, and data output Y register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DYRA0 through DYRA7 and DYRB0 through DYRB7 indicate bit locations, with *DYRA* and *DYRB* denoting the bits that are in the data output Y registers. DYRA7 and DYRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DYRA7	**DYRA6**	**DYRA5**	**DYRA4**	**DYRA3**	**DYRA2**	**DYRA1**	**DYRA0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DYRB7	**DYRB6**	**DYRB5**	**DYRB4**	**DYRB3**	**DYRB2**	**DYRB1**	**DYRB0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 14: Data Output Y Registers A and B

Data Output Z Registers A and B

The data output Z registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel Z. Data output Z register A contains the MSB from the measurement result, and data output Z register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DZRA0 through DZRA7 and DZRB0 through DZRB7 indicate bit locations, with *DZRA* and *DZRB* denoting the bits that are in the data output Z registers. DZRA7 and DZRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DZRA7	**DZRA6**	**DZRA5**	**DZRA4**	**DZRA3**	**DZRA2**	**DZRA1**	**DZRA0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DZRB7	**DZRB6**	**DZRB5**	**DZRB4**	**DZRB3**	**DZRB2**	**DZRB1**	**DZRB0**
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 15: Data Output Z Registers A and B

Data Output Register Operation

When one or more of the output registers are read, new data cannot be placed in any of the output data registers until all six data output registers are read. This requirement also impacts DRDY and RDY, which cannot be cleared until new data is placed in all the output registers.

Status Register

The status register is an 8-bit read-only register. This register is used to indicate device status. SR0 through SR7 indicate bit locations, with *SR* denoting the bits that are in the status register. SR7 denotes the first bit of the data stream.

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
(0)	(0)	(0)	(0)	(0)	(0)	LOCK (0)	RDY(0)

Table 16: Status Register

Location	Name	Description
SR7 to SR2	0	These bits are reserved.
SR1	LOCK	Data output register lock. This bit is set when: 1.some but not all for of the six data output registers have been read, 2. Mode register has been read. When this bit is set, the six data output registers are locked and any new data will not be placed in these register until one of these conditions are met: 1.all six bytes have been read, 2. the mode register is changed, 3. the measurement configuration (CRA) is changed, 4. power is reset.
SR0	RDY	Ready Bit. Set when data is written to all six data registers. Cleared when device initiates a write to the data output registers and after one or more of the data output registers are written to. When RDY bit is clear it shall remain cleared for a 250 μs. DRDY pin can be used as an alternative to the status register for monitoring the device for measurement data.

Table 17: Status Register Bit Designations

Identification Register A

The identification register A is used to identify the device. IRA0 through IRA7 indicate bit locations, with *IRA* denoting the bits that are in the identification register A. IRA7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

The identification value for this device is stored in this register. This is a read-only register.
Register values. ASCII value *H*

IRA7	IRA6	IRA5	IRA4	IRA3	IRA2	IRA1	IRA0
0	1	0	0	1	0	0	0

Table 18: Identification Register A Default Values

Identification Register B

The identification register B is used to identify the device. IRB0 through IRB7 indicate bit locations, with *IRB* denoting the bits that are in the identification register A. IRB7 denotes the first bit of the data stream.

Register values. ASCII value *4*

IRB7	IRB6	IRB5	IRB4	IRB3	IRB2	IRB1	IRB0
0	0	1	1	0	1	0	0

Table 19: Identification Register B Default Values

Identification Register C

The identification register C is used to identify the device. IRC0 through IRC7 indicate bit locations, with *IRC* denoting the bits that are in the identification register A. IRC7 denotes the first bit of the data stream.

Register values. ASCII value *3*

IRC7	IRC6	IRC5	IRC4	IRC3	IRC2	IRC1	IRC0
0	0	1	1	0	0	1	1

Table 20: Identification Register C Default Values

I^{2}C COMMUNICATION PROTOCOL

The HMC5883L communicates via a two-wire I^{2}C bus system as a slave device. The HMC5883L uses a simple protocol with the interface protocol defined by the I^{2}C bus specification, and by this document. The data rate is at the standard-mode 100kbps or 400kbps rates as defined in the I^{2}C Bus Specifications. The bus bit format is an 8-bit Data/Address send and a 1-bit acknowledge bit. The format of the data bytes (payload) shall be case sensitive ASCII characters or binary data to the HMC5883L slave, and binary data returned. Negative binary values will be in two's complement form. The default (factory) HMC5883L 8-bit slave address is 0x3C for write operations, or 0x3D for read operations.

The HMC5883L Serial Clock (SCL) and Serial Data (SDA) lines require resistive pull-ups (Rp) between the master device (usually a host microprocessor) and the HMC5883L. Pull-up resistance values of about 2.2K to 10K ohms are recommended with a nominal VDDIO voltage. Other resistor values may be used as defined in the I^{2}C Bus Specifications that can be tied to VDDIO.

The SCL and SDA lines in this bus specification may be connected to multiple devices. The bus can be a single master to multiple slaves, or it can be a multiple master configuration. All data transfers are initiated by the master device, which is responsible for generating the clock signal, and the data transfers are 8 bit long. All devices are addressed by I^{2}C's unique 7-bit address. After each 8-bit transfer, the master device generates a 9th clock pulse, and releases the SDA line. The receiving device (addressed slave) will pull the SDA line low to acknowledge (ACK) the successful transfer or leave the SDA high to negative acknowledge (NACK).

Per the I^{2}C spec, all transitions in the SDA line must occur when SCL is low. This requirement leads to two unique conditions on the bus associated with the SDA transitions when SCL is high. Master device pulling the SDA line low while the SCL line is high indicates the Start (S) condition, and the Stop (P) condition is when the SDA line is pulled high while the SCL line is high. The I^{2}C protocol also allows for the Restart condition in which the master device issues a second start condition without issuing a stop.

All bus transactions begin with the master device issuing the start sequence followed by the slave address byte. The address byte contains the slave address; the upper 7 bits (bits7-1), and the Least Significant bit (LSb). The LSb of the address byte designates if the operation is a read (LSb=1) or a write (LSb=0). At the 9th clock pulse, the receiving slave device will issue the ACK (or NACK). Following these bus events, the master will send data bytes for a write operation, or the slave will clock out data with a read operation. All bus transactions are terminated with the master issuing a stop sequence.

I^{2}C bus control can be implemented with either hardware logic or in software. Typical hardware designs will release the SDA and SCL lines as appropriate to allow the slave device to manipulate these lines. In a software implementation, care must be taken to perform these tasks in code.

OPERATIONAL EXAMPLES

The HMC5883L has a fairly quick stabilization time from no voltage to stable and ready for data retrieval. The nominal 56 milli-seconds with the factory default single measurement mode means that the six bytes of magnetic data registers (DXRA, DXRB, DZRA, DZRB, DYRA, and DYRB) are filled with a valid first measurement.

To change the measurement mode to continuous measurement mode, after the power-up time send the three bytes:

0x3C 0x02 0x00

This writes the 00 into the second register or mode register to switch from single to continuous measurement mode setting. With the data rate at the factory default of 15Hz updates, a 67 milli-second typical delay should be allowed by the I^{2}C master before querying the HMC5883L data registers for new measurements. To clock out the new data, send:

0x3D, and clock out DXRA, DXRB, DZRA, DZRB, DYRA, and DYRB located in registers 3 through 8. The HMC5883L will automatically re-point back to register 3 for the next 0x3D query. All six data registers must be read properly before new data can be placed in any of these data registers.

Below is an example of a (power-on) initialization process for “continuous-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default, normal measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
3. Write Mode (02) – send **0x3C 0x02 0x00** (Continuous-measurement mode)
4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
5. Loop
    - Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
    - Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.
    - Send **0x3C 0x03** (point to first data register 03)
    - Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin

   End_loop

Below is an example of a (power-on) initialization process for “single-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default or any other rate, normal measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
3. For each measurement query:
    - Write Mode (02) – send **0x3C 0x02 0x01** (Single-measurement mode)
    - Wait 6 ms or monitor status register or DRDY hardware interrupt pin
    - Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
    - Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.

SELF TEST OPERATION

To check the HMC5883L for proper operation, a self test feature in incorporated in which the sensor offset straps are excited to create a nominal field strength (bias field) to be measured. To implement self test, the least significant bits (MS1 and MS0) of configuration register A are changed from 00 to 01 (positive bias) or 10 (negetive bias).

Then, by placing the mode register into single or continuous-measurement mode, two data acquisition cycles will be made on each magnetic vector. The first acquisition will be a set pulse followed shortly by measurement data of the external field. The second acquisition will have the offset strap excited (about 10 mA) in the positive bias mode for X, Y, and Z axes to create about a 1.1 gauss self test field plus the external field. The first acquisition values will be subtracted from the second acquisition, and the net measurement will be placed into the data output registers.

Since self test adds ~1.1 Gauss additional field to the existing field strength, using a reduced gain setting prevents sensor from being saturated and data registers overflowed. For example, if the configuration register B is set to 0xA0 (Gain=5), values around +452 LSb (1.16 Ga * 390 LSb/Ga) will be placed in the X and Y data output registers and around +421 (1.08 Ga * 390 LSb/Ga) will be placed in Z data output register. To leave the self test mode, change MS1 and MS0 bit of the configuration register A back to 00 (Normal Measurement Mode). Acceptable limits of the self test values depend on the gain setting. Limits for Gain=5 is provided in the specification table.

Below is an example of a “positive self test” process using continuous-measurement mode:

```
1. Write CRA (00) – send 0x3C 0x00 0x71 (8-average, 15 Hz default, positive self test measurement)
2. Write CRB (01) – send 0x3C 0x01 0xA0 (Gain=5)
3. Write Mode (02) – send 0x3C 0x02 0x00 (Continuous-measurement mode)
4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
5. Loop
           Send 0x3D 0x06 (Read all 6 bytes. If gain is changed then this data set is using previous gain)
           Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.
           Send 0x3C 0x03 (point to first data register 03)
           Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin
   End_loop
6. Check limits –
   If all 3 axes (X, Y, and Z) are within reasonable limits (243 to 575 for Gain=5, adjust these limits basing on the
   gain setting used. See an example below.) Then
           All 3 axes pass positive self test
           Write CRA (00) – send 0x3C 0x00 0x70 (Exit self test mode and this procedure)
   Else
           If Gain<7
                   Write CRB (01) – send 0x3C 0x01 0x_0 (Increase gain setting and retry, skip the next data set)
           Else
                   At least one axis did not pass positive self test
                   Write CRA (00) – send 0x3C 0x00 0x70 (Exit self test mode and this procedure)
   End If
```

Below is an example of how to adjust the “positive self” test limits basing on the gain setting:

1. If Gain = 6, self test limits are:
   Low Limit = 243 * 330/390 = 206
   High Limit = 575 * 330/390 = 487

2. If Gain = 7, self test limits are:
   Low Limit = 243 * 230/390 = 143
   High Limit = 575 * 230/390 = 339

SCALE FACTOR TEMPERATURE COMPENSATION

The built-in self test can also be used to periodically compensate the scaling errors due to temperature variations. A compensation factor can be found by comparing the self test outputs with the ones obtained at a known temperature. For example, if the self test output is 400 at room temperature and 300 at the current temperature then a compensation factor of (400/300) should be applied to all current magnetic readings. A temperature sensor is not required using this method.

Below is an example of a temperature compensation process using positive self test method:

1. If self test measurement at a temperature “when the last magnetic calibration was done”:
    X_STP = 400
    Y_STP = 410
    Z_STP = 420
2. If self test measurement at a different tmperature:
    X_STP = 300 (Lower than before)
    Y_STP = 310 (Lower than before)
    Z_STP = 320 (Lower than before)

    Then
    X_TempComp = 400/300
    Y_TempComp = 410/310
    Z_TempComp = 420/320
3. Applying to all new measurements:
    X = X * X_TempComp
    Y = Y * Y_TempComp
    Z = Z * Z_TempComp

    Now all 3 axes are temperature compensated, i.e. sensitivity is same as “when the last magnetic calibration was done”; therefore, the calibration coefficients can be applied without modification.
4. Repeat this process periodically or,for every Δt degrees of temperature change measured, if available.

ORDERING INFORMATION

Ordering Number	Product
HMC5883L-T HMC5883L-TR	Cut Tape Tape and Reel 4k pieces/reel

Caution
This part is sensitive to damage by electrostatic discharge. Use ESD precautionary procedures when touching, removing or inserting.

CAUTION: ESDS CAT. 1B

FIND OUT MORE

For more information on Honeywell’s Magnetic Sensors visit us online at www.magneticsensors.com or contact us at 1-800-323-8295 (763-954-2474 internationally).

The application circuits herein constitute typical usage and interface of Honeywell product. Honeywell does not warranty or assume liability of customer-designed circuits derived from this description or depiction.

Honeywell reserves the right to make changes to improve reliability, function or design. Honeywell does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

U.S. Patents 4,441,072, 4,533,872, 4,569,742, 4,681,812, 4,847,584 and 6,529,114 apply to the technology described

Honeywell
12001 Highway 55
Plymouth, MN 55441
Tel: 800-323-8295
www.magneticsensors.com

Form # 900405 Rev E
February 2013

PD -95225

International
IOR Rectifier

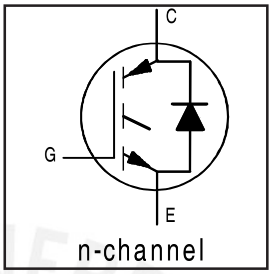
IRG4PC50FDPbF

INSULATED GATE BIPOLAR TRANSISTOR WITH ULTRAFAST SOFT RECOVERY DIODE

Fast CoPack IGBT

Features

- Fast: Optimized for medium operating frequencies (1-5 kHz in hard switching, >20 kHz in resonant mode).
- Generation 4 IGBT design provides tighter parameter distribution and higher efficiency than Generation 3
- IGBT co-packaged with HEXFRED^{TM} ultrafast, ultra-soft-recovery anti-parallel diodes for use in bridge configurations
- Industry standard TO-247AC package
- Lead-Free



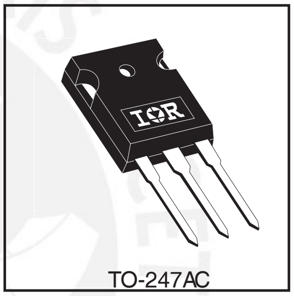
V_{CES} = 600V

$V_{CE(on)\ typ.}$ = 1.45V

@V_{GE} = 15V, I_C = 39A

Benefits

- Generation -4 IGBT's offer highest efficiencies available
- IGBT's optimized for specific application conditions
- HEXFRED diodes optimized for performance with IGBT's . Minimized recovery characteristics require less/no snubbing
- Designed to be a "drop-in" replacement for equivalent industry-standard Generation 3 IR IGBT's



TO-247AC

Absolute Maximum Ratings

	Parameter	Max.	Units
V_{CES}	Collector-to-Emitter Voltage	600	V
I_C @ T_C = 25°C	Continuous Collector Current	70	A
I_C @ T_C = 100°C	Continuous Collector Current	39	
I_{CM}	Pulsed Collector Current ①	280	
I_{LM}	Clamped Inductive Load Current ②	280	
I_F @ T_C = 100°C	Diode Continuous Forward Current	25	
I_{FM}	Diode Maximum Forward Current	280	
V_{GE}	Gate-to-Emitter Voltage	± 20	V
P_D @ T_C = 25°C	Maximum Power Dissipation	200	W
P_D @ T_C = 100°C	Maximum Power Dissipation	78	
T_J T_{STG}	Operating Junction and Storage Temperature Range	-55 to +150	°C
	Soldering Temperature, for 10 sec.	300 (0.063 in. (1.6mm) from case)	
	Mounting Torque, 6-32 or M3 Screw.	10 lbf•in (1.1 N•m)	

Thermal Resistance

	Parameter	Min.	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case - IGBT	------	------	0.64	°C/W
$R_{\theta JC}$	Junction-to-Case - Diode	------	------	0.83	
$R_{\theta CS}$	Case-to-Sink, flat, greased surface	------	0.24	------	
$R_{\theta JA}$	Junction-to-Ambient, typical socket mount	-----	-----	40	
Wt	Weight	------	6 (0.21)	------	g (oz)

04/29/04

Electrical Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions	
$V_{(BR)CES}$	Collector-to-Emitter Breakdown Voltage③	600	----	----	V	V_{GE} = 0V, I_C = 250μA	
$\Delta V_{(BR)CES}/\Delta T_J$	Temperature Coeff. of Breakdown Voltage	----	0.62	----	V/°C	V_{GE} = 0V, I_C = 1.0mA	
$V_{CE(on)}$	Collector-to-Emitter Saturation Voltage	----	1.45	1.6	V	I_C = 39A	V_{GE} = 15V
		----	1.79	----		I_C = 70A	See Fig. 2, 5
		----	1.53	----		I_C = 39A, T_J = 150°C	
$V_{GE(th)}$	Gate Threshold Voltage	3.0	----	6.0		V_{CE} = V_{GE}, I_C = 250μA	
$\Delta V_{GE(th)}/\Delta T_J$	Temperature Coeff. of Threshold Voltage	----	-14	----	mV/°C	V_{CE} = V_{GE}, I_C = 250μA	
g_{fe}	Forward Transconductance ④	21	30	----	S	V_{CE} = 100V, I_C = 39A	
I_{CES}	Zero Gate Voltage Collector Current	----	----	250	μA	V_{GE} = 0V, V_{CE} = 600V	
		----	----	6500		V_{GE} = 0V, V_{CE} = 600V, T_J = 150°C	
V_{FM}	Diode Forward Voltage Drop	----	1.3	1.7	V	I_C = 25A	See Fig. 13
		----	1.2	1.5		I_C = 25A, T_J = 150°C	
I_{GES}	Gate-to-Emitter Leakage Current	----	----	±100	nA	V_{GE} = ±20V	

Switching Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions		
Q_g	Total Gate Charge (turn-on)	----	190	290		I_C = 39A		
Qge	Gate - Emitter Charge (turn-on)	----	28	42	nC	V_{CC} = 400V	See Fig. 8	
Q_{gc}	Gate - Collector Charge (turn-on)	----	65	97		V_{GE} = 15V		
$t_{d(on)}$	Turn-On Delay Time	----	55	----		T_J = 25°C		
t_r	Rise Time	----	25	----	ns	I_C = 39A, V_{CC} = 480V		
$t_{d(off)}$	Turn-Off Delay Time	----	240	360		V_{GE} = 15V, R_G = 5.0Ω		
t_f	Fall Time	----	140	210		Energy losses include "tail" and		
E_{on}	Turn-On Switching Loss	----	1.5	----		diode reverse recovery.		
E_{off}	Turn-Off Switching Loss	----	2.4	----	mJ	See Fig. 9, 10, 11, 18		
E_{ts}	Total Switching Loss	----	3.9	5.0				
$t_{d(on)}$	Turn-On Delay Time	----	59	----		T_J = 150°C,	See Fig. 9, 10, 11, 18	
t_r	Rise Time	----	27	----	ns	I_C = 39A, V_{CC} = 480V		
$t_{d(off)}$	Turn-Off Delay Time	----	400	----		V_{GE} = 15V, R_G = 5.0Ω		
t_f	Fall Time	----	260	----		Energy losses include "tail" and		
E_{ts}	Total Switching Loss	----	6.5	----	mJ	diode reverse recovery.		
L_E	Internal Emitter Inductance	----	13	----	nH	Measured 5mm from package		
C_{ies}	Input Capacitance	----	4100	----		V_{GE} = 0V		
C_{oes}	Output Capacitance	----	250	----	pF	V_{CC} = 30V	See Fig. 7	
C_{res}	Reverse Transfer Capacitance	----	49	----		ƒ = 1.0MHz		
t_{rr}	Diode Reverse Recovery Time	----	50	75	ns	T_J = 25°C	See Fig.	
		----	105	160		T_J = 125°C	14	I_F = 25A
I_{rr}	Diode Peak Reverse Recovery Current	----	4.5	10	A	T_J = 25°C	See Fig.	
		----	8.0	15		T_J = 125°C	15	V_R = 200V
Q_{rr}	Diode Reverse Recovery Charge	----	112	375	nC	T_J = 25°C	See Fig.	
		----	420	1200		T_J = 125°C	16	di/dt 200A/μs
$di_{(rec)M}/dt$	Diode Peak Rate of Fall of Recovery	----	250	----	A/μs	T_J = 25°C	See Fig.	
	During t_b	----	160	----		T_J = 125°C	17	

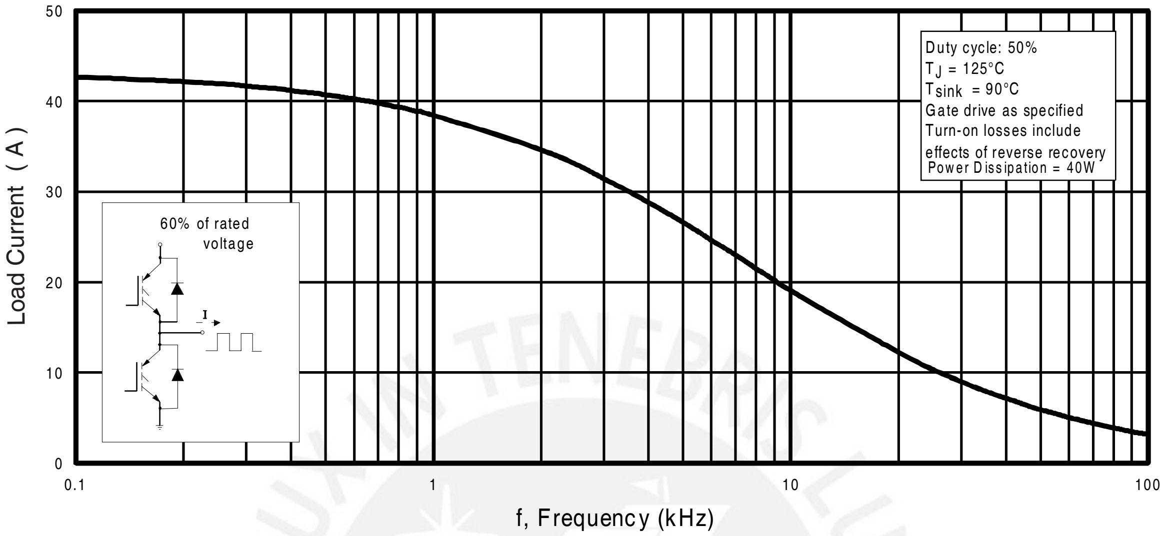
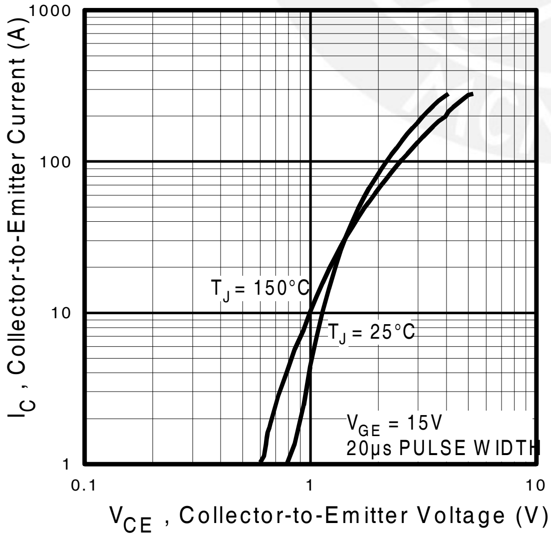


Fig. 1 - Typical Load Current vs. Frequency
(Load Current = I_{RMS} of fundamental)



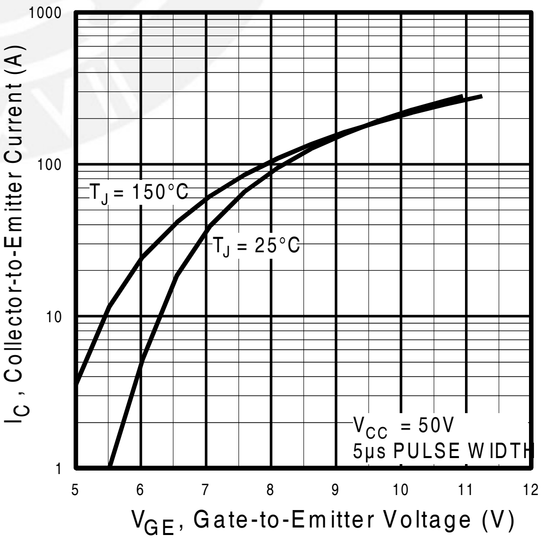
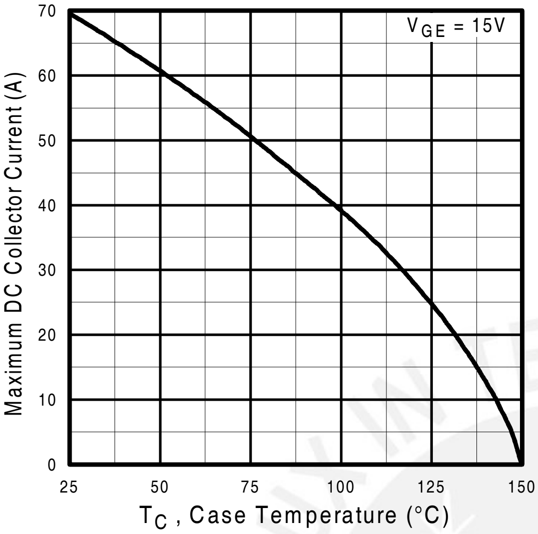


Fig. 2 - Typical Output Characteristics

Fig. 3 - Typical Transfer Characteristics



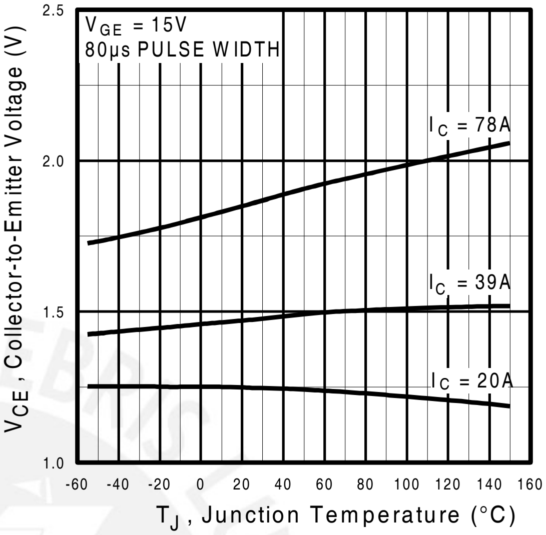


Fig. 4 - Maximum Collector Current vs. Case Temperature

Fig. 5 - Typical Collector-to-Emitter Voltage vs. Junction Temperature

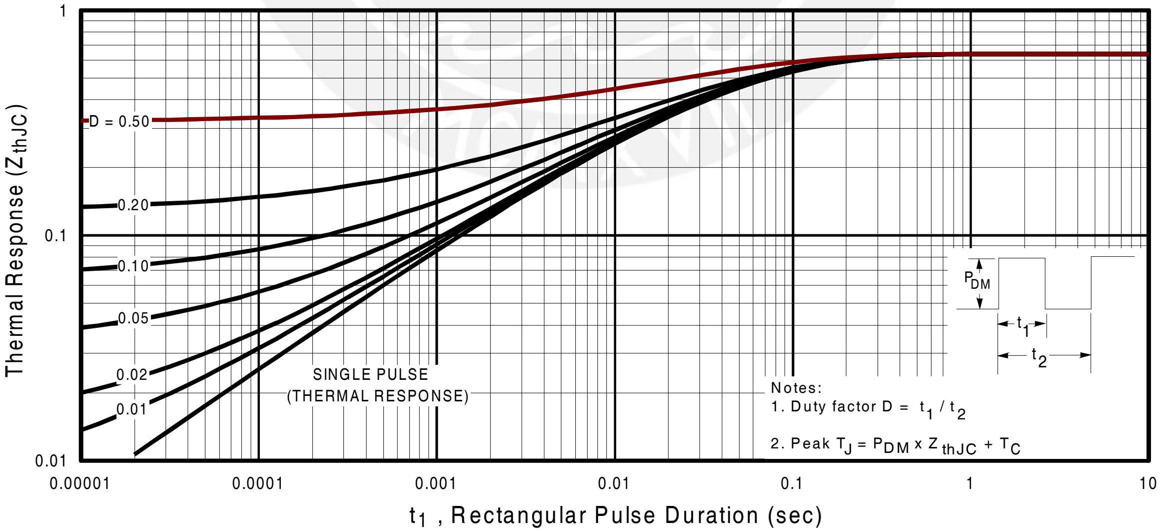


Fig. 6 - Maximum IGBT Effective Transient Thermal Impedance, Junction-to-Case

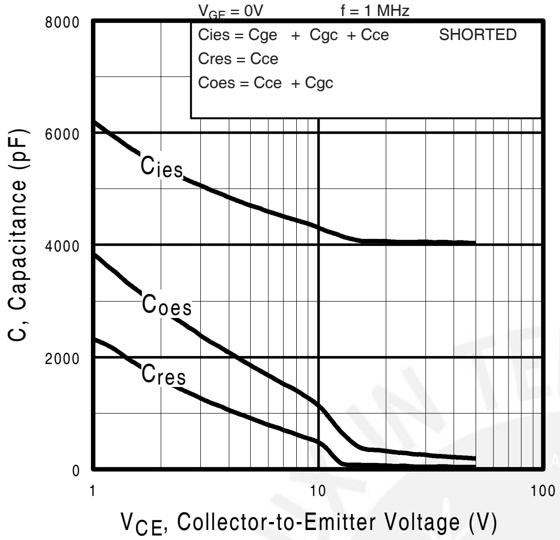


Fig. 7 - Typical Capacitance vs. Collector-to-Emitter Voltage

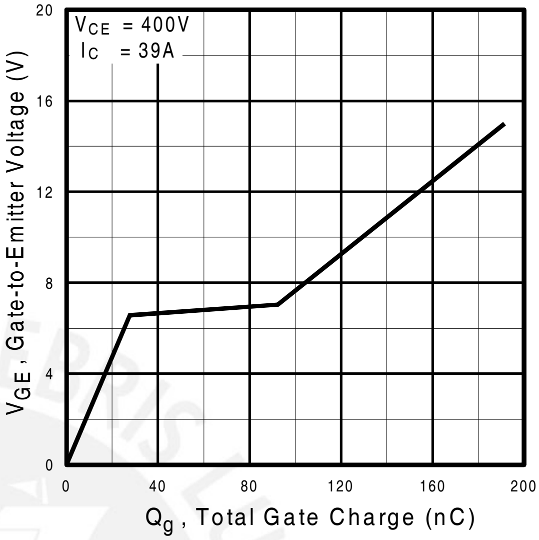


Fig. 8 - Typical Gate Charge vs. Gate-to-Emitter Voltage

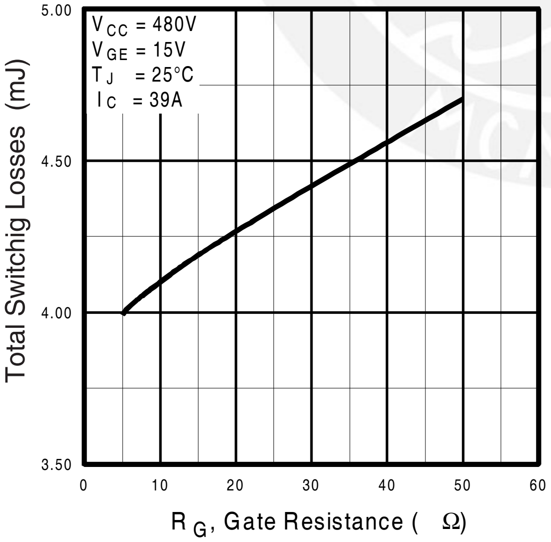


Fig. 9 - Typical Switching Losses vs. Gate Resistance

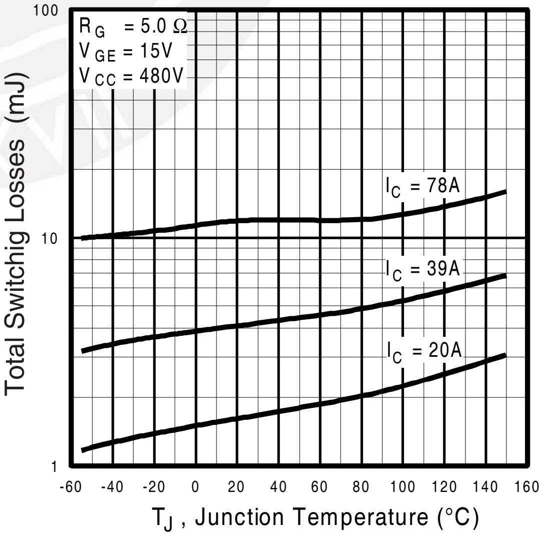


Fig. 10 - Typical Switching Losses vs. Junction Temperature

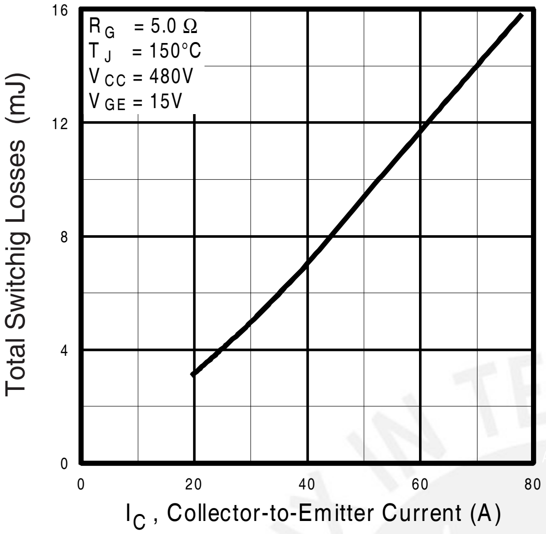


Fig. 11 - Typical Switching Losses vs. Collector-to-Emitter Current

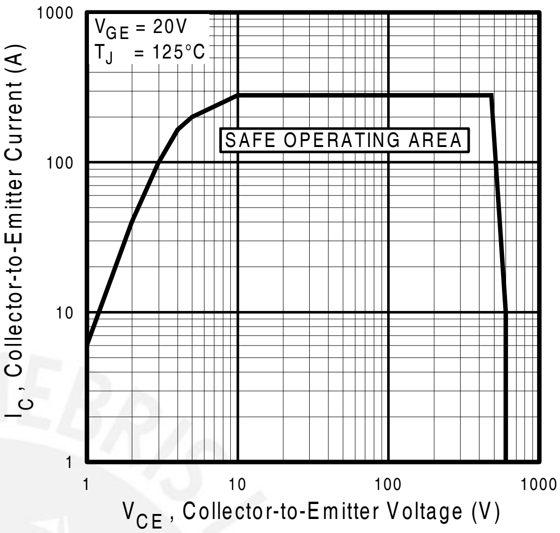


Fig. 12 - Turn-Off SOA

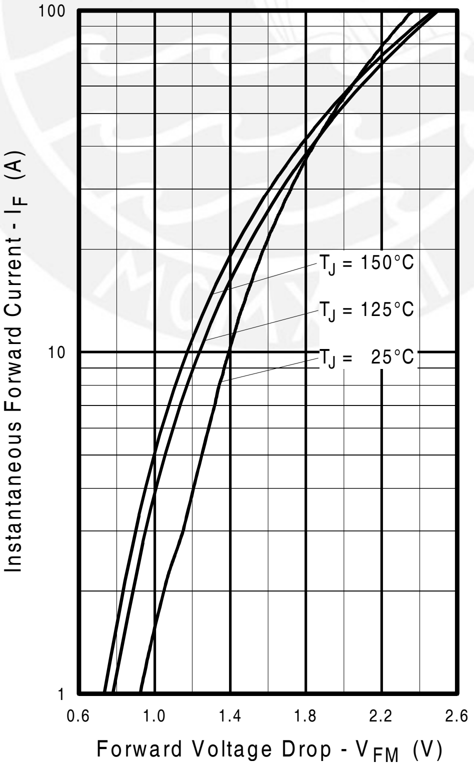


Fig. 13 - Maximum Forward Voltage Drop vs. Instantaneous Forward Current

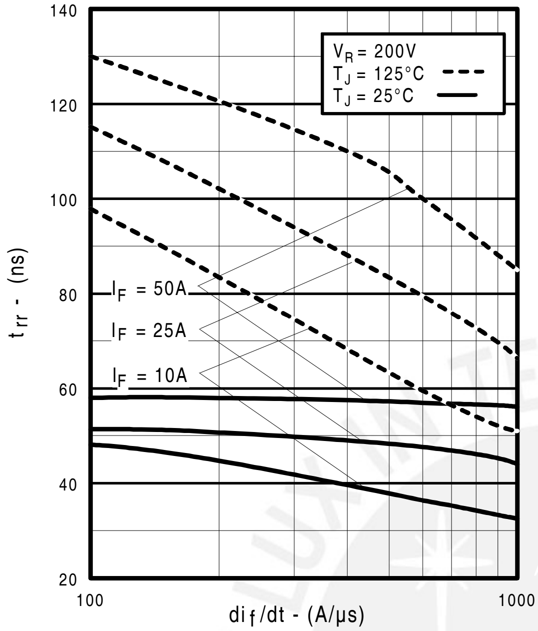


Fig. 14 - Typical Reverse Recovery vs. di_f/dt

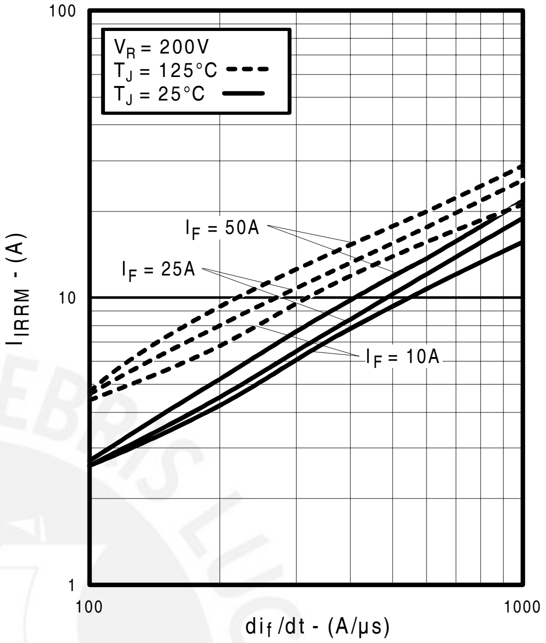


Fig. 15 - Typical Recovery Current vs. di_f/dt

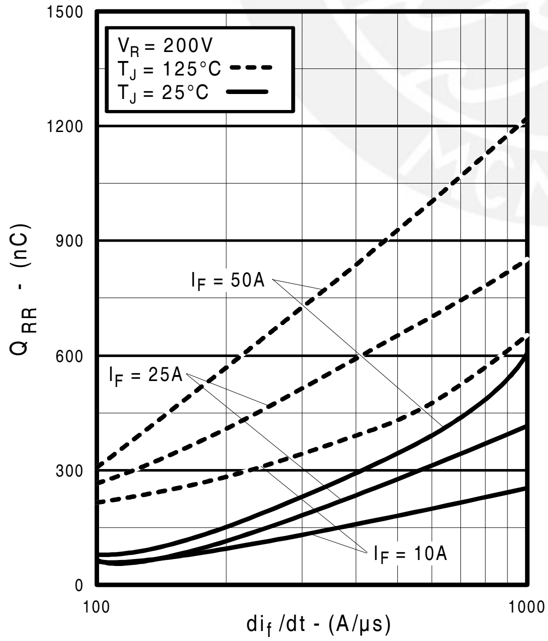


Fig. 16 - Typical Stored Charge vs. di_f/dt

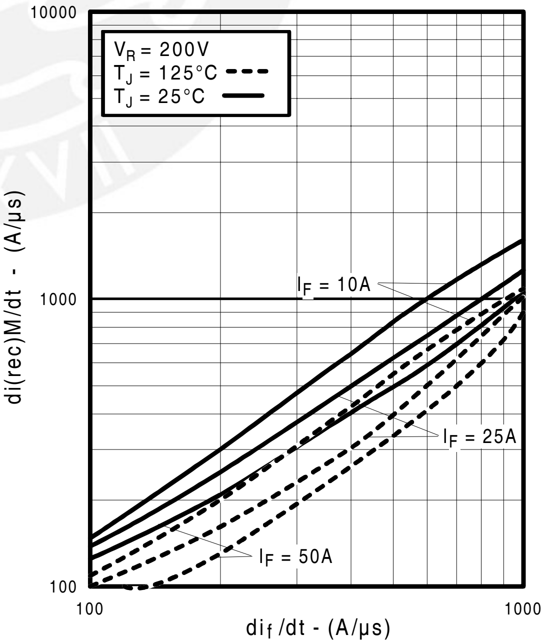


Fig. 17 - Typical $di_{(rec)M}/dt$ vs. di_f/dt

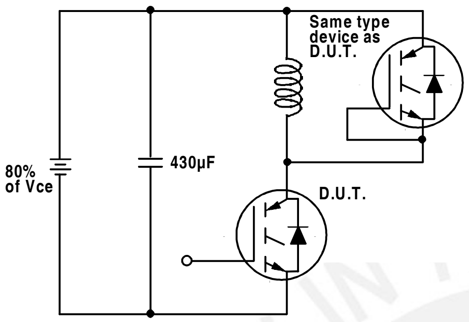


Fig. 18a - Test Circuit for Measurement of I_{LM}, E_{on}, $E_{off(diode)}$, t_{rr}, Q_{rr}, I_{rr}, $t_{d(on)}$, t_r, $t_{d(off)}$, t_f

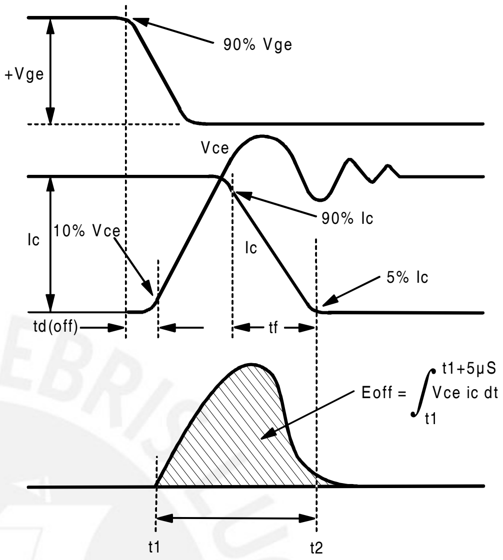
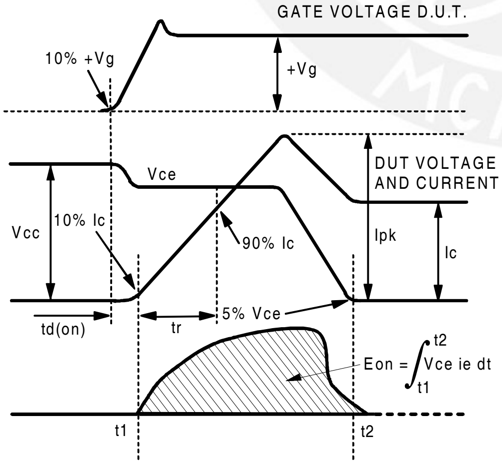


Fig. 18b - Test Waveforms for Circuit of Fig. 18a, Defining E_{off}, $t_{d(off)}$, t_f



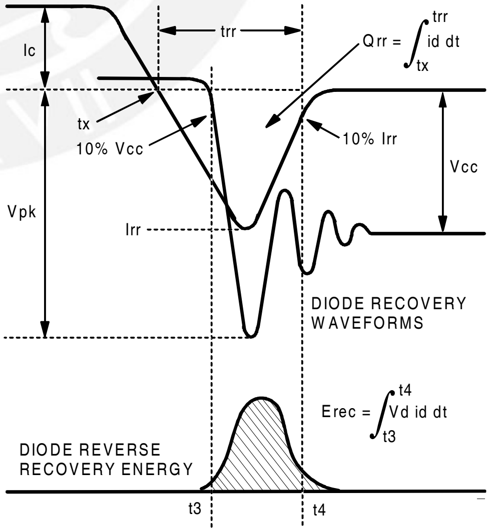


Fig. 18c - Test Waveforms for Circuit of Fig. 18a, Defining E_{on}, $t_{d(on)}$, t_r

Fig. 18d - Test Waveforms for Circuit of Fig. 18a, Defining E_{rec}, t_{rr}, Q_{rr}, I_{rr}

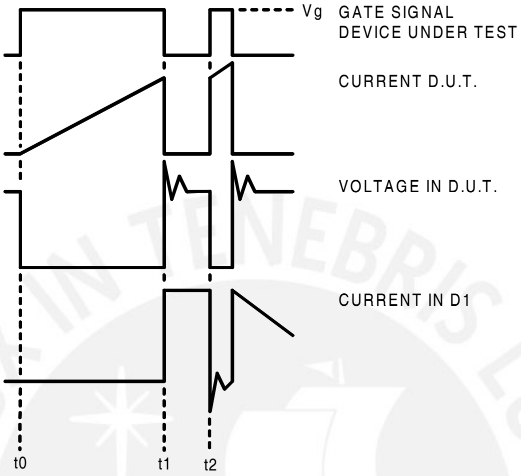
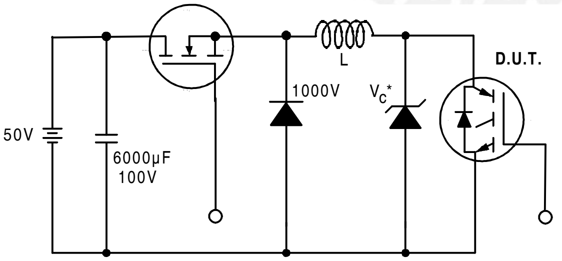


Figure 18e. Macro Waveforms for Figure 18a's Test Circuit



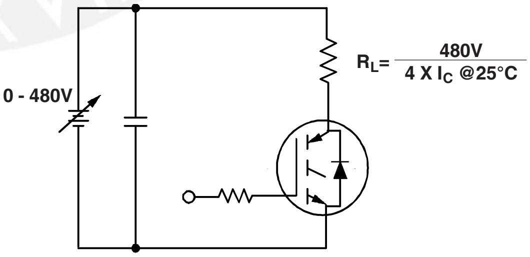


Figure 19. Clamped Inductive Load Test Circuit

Figure 20. Pulsed Collector Current Test Circuit

Notes:

① Repetitive rating: V_{GE}=20V; pulse width limited by maximum junction temperature (figure 20)

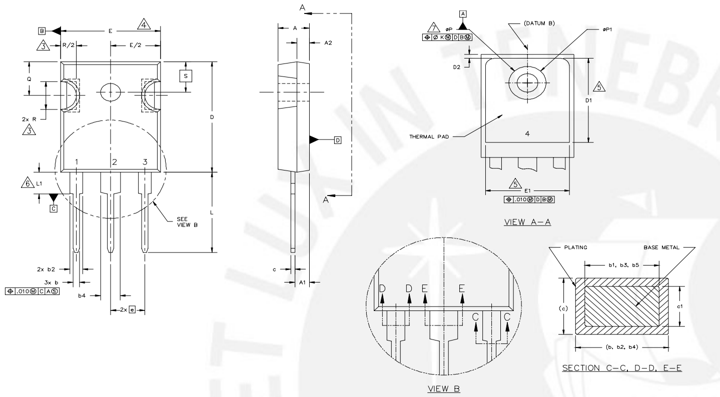
② V_{CC}=80%(V_{CES}), V_{GE}=20V, L=10µH, R_G = 5.0Ω (figure 19)

③ Pulse width ≤ 80µs; duty factor ≤ 0.1%.

④ Pulse width 5.0µs, single shot.

TO-247AC Package Outline

Dimensions are shown in millimeters (inches)



NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M 1994.
2. DIMENSIONS ARE SHOWN IN INCHES [MILLIMETERS]
3. CONTOUR OF SLOT OPTIONAL.
4. DIMENSION D & E DO NOT INCLUDE MOLD FLASH. MOLD FLASH SHALL NOT EXCEED .005" (0.127) PER SIDE. THESE DIMENSIONS ARE MEASURED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY.
5. THERMAL PAD CONTOUR OPTIONAL WITHIN DIMENISONS D1 & E1.
6. LEAD FINISH UNCONTROLLED IN L1.
7. øP TO HAVE A MAXIMUM DRAFT ANGLE OF 1.5 ° TO THE TOP OF THE PART WITH A MAXIMUM HOLE DIAMETER OF .154" [3.91].
8. OUTLINE CONFORMS TO JEDEC OUTLINE TO-247 WITH THE EXCEPTION OF DIMENSION c.

SYMBOL	DIMENSIONS				NOTES
	INCHES		MILLIMETERS		
	MIN.	MAX.	MIN.	MAX.	
A	.183	.209	4.65	5.31	
A1	.087	.102	2.21	2.59	
A2	.059	.098	1.50	2.49	
b	.039	.055	0.99	1.40	
b1	.039	.053	0.99	1.35	
b2	.065	.094	1.65	2.39	
b3	.065	.092	1.65	2.37	
b4	.102	.135	2.59	3.43	
b5	.102	.133	2.59	3.38	
c	.015	.034	0.38	0.86	
c1	.015	.030	0.38	0.76	
D	.776	.815	19.71	20.70	4
D1	.515	–	13.08	–	5
D2	.020	.030	0.51	0.76	
E	.602	.625	15.29	15.87	4
E1	.540	–	15.72	–	
e	.215 BSC		5.46 BSC		
Øk	.010		2.54		
L	.559	.634	14.20	16.10	
L1	.146	.169	3.71	4.29	
N	3		7.62 BSC		
ØP	.140	.144	3.56	3.66	
ØP1	–	.275	–	6.98	
Q	.209	.224	5.31	5.69	
R	.178	.216	4.52	5.49	
S	.217 BSC		5.51 BSC		

LEAD ASSIGNMENTS

HEXFET

1.– GATE
2.– DRAIN
3.– SOURCE
4.– DRAIN

IGBTs, CoPACK

1.– GATE
2.– COLLECTOR
3.– EMITTER
4.– COLLECTOR

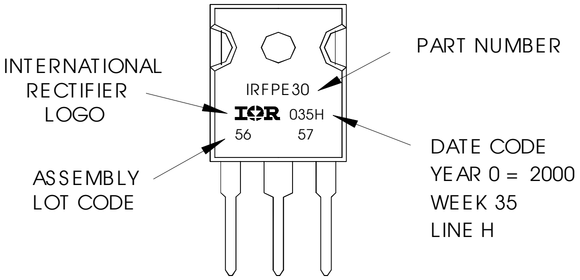
DIODES

1.– ANODE/OPEN
2.– CATHODE
3.– ANODE

TO-247AC Part Marking Information

EXAMPLE: THIS IS AN IRFPE30
WITH ASSEMBLY
LOT CODE 5657
ASSEMBLED ON WW 35, 2000
IN THE ASSEMBLY LINE "H"

Note: "P" in assembly line position indicates "Lead-Free"



Data and specifications subject to change without notice.

International
IOR Rectifier

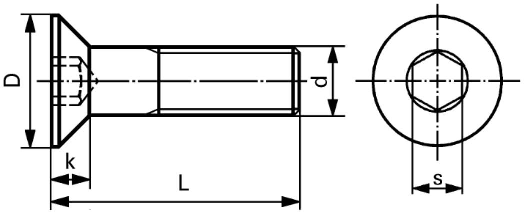
IR WORLD HEADQUARTERS: 233 Kansas St., El Segundo, California 90245, USA Tel: (310) 252-7105
TAC Fax: (310) 252-7903
Visit us at www.irf.com for sales contact information. 04/04

Note: For the most current drawings please refer to the IR website at:
http://www.irf.com/package/

10.9

Socket flat head screws, steel

DIN 7991 / ISO 10642



d	M 3		M 4		M 5		M 6	
D	6		8		10		12	
s	2		2,5		3		4	
k max.	1,7		2,3		2,8		3,3	
k ISO max.	1,86		2,48		3,1		3,72	
L quantity	100	1000+	100	1000+	100	1000+	100	1000+
5	18.40	13.70						
6	17.20	12.80	16.10	12.–	15.–	12.50		
8	15.90	11.90	14.50	10.80	12.90	9.70	13.90	10.40
10	15.20	11.30	13.–	9.70	12.70	9.50	14.50	10.80
12	15.20	11.30	13.60	10.20	12.70	9.50	12.80	9.60
(14)	18.10	13.50	18.90	13.–	14.70	10.80	15.50	11.50
16	16.–	12.–	15.20	11.30	13.10	9.80	12.70	9.50
(18)	19.–	14.20	18.10	12.60	14.20	10.50	14.80	10.80
20	16.50	12.40	15.20	11.30	13.60	10.20	13.10	9.80
(22)	19.40	14.70	17.90	13.30	14.20	10.50	15.30	11.30
25	17.90	13.30	16.80	12.60	15.80	11.90	14.80	10.80
28			23.50	17.80			23.–	16.10
30	19.30	14.50	17.90	13.30	16.10	12.50	16.20	12.20
35	21.75	16.20	21.–	15.60	21.–	15.70	18.50	13.80
40	22.50	16.80	21.–	15.60	21.–	15.70	18.70	13.90
45			24.–	17.80	21.–	15.70	22.–	16.40
50			30.–	23.50	24.–	18.–	26.–	19.40
55					43.50	34.–	39.25	31.–
60					44.50	34.50	40.50	32.–
65					67.50	51.50	57.50	44.50
70					70.–	57.–	60.50	46.50
(75)					83.–	66.50	77.50	58.–
80					103.–	88.50	78.50	58.50
90							123.–	89.50
100							127.–	98.50
120							150.–	121.–

Reduced tightening and loads: refer to page T 25

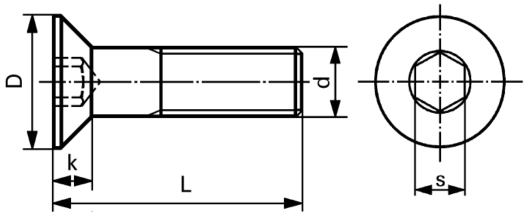
\+ 10% discount for 10 000 pieces per item

----- Below dash-line: fully or partially threaded at our option

10.9

Socket flat head screws, steel

DIN 7991 / ISO 10642



d	M 8		M 10		M 12	(M14)	M 16	M 20	M 24
D	16		20		24	27	30	36	39
s	5		6		8	10	10	12	14
k max.	4,4		5,5		6,5	7	7,5	8,5	14
k ISO max.	4,96		6,2		7,44	8,4	8,8	10,16	-
L quantity	100	1000+	100	1000	100	100	100	100	100
10	32.–	24.75							
12	29.25	22.75	89.–	74.50					
(14)	32.50	25.25	75.–	63.–					
16	17.60	13.10	57.–	44.50	77.50				
(18)	20.75	15.10	73.–	59.–					
20	16.50	12.40	35.–	27.25	54.–		198.–		
(22)	24.–	17.10			60.–				
25	17.10	12.80	36.–	28.–	50.50	172.–	200.–		
30	18.80	14.–	37.25	29.–	52.50	129.–	129.–		760.–
35	20.75	15.50	42.50	32.75	54.–	134.–	123.–	340.–	
40	22.25	16.60	42.50	32.75	57.–	135.–	124.–	292.–	770.–
45	25.75	19.20	61.50	48.–	66.50	148.–	136.–	304.–	615.–
50	29.50	22.75	61.50	48.–	69.50	148.–	136.–	304.–	715.–
55	36.–	28.–	64.50	49.50	74.50	171.–	149.–	344.–	725.–
60	42.–	32.75	65.–	50.–	96.–	171.–	149.–	344.–	730.–
65	59.50	46.–	78.–	61.–	123.–	194.–	224.–	392.–	810.–
70	66.–	45.50	71.–	57.50	116.–	201.–	201.–	394.–	785.–
(75)	71.–	56.–	94.–	72.–	129.–	256.–	254.–	470.–	845.–
80	73.–	59.–	80.50	65.50	136.–	231.–	224.–	510.–	815.–
90	81.–	65.50	98.–	79.–	153.–	318.–	286.–	555.–	865.–
100	89.–	74.–	121.–	98.–	191.–	380.–	342.–	545.–	885.–
110	120.–	98.–	153.–	129.–	239.–		392.–	605.–	950.–
120	129.–	109.–	166.–	140.–	272.–		405.–	635.–	1040.–
130			175.–	150.–	294.–		450.–	755.–	1200.–
140					314.–		505.–	920.–	1250.–
150					348.–		565.–	775.–	1260.–
160							640.–	865.–	1440.–
180								930.–	1560.–
200								1000.–	

Reduced tightening and loads: refer to page T 25

+ 10% discount for 10 000 pieces per item

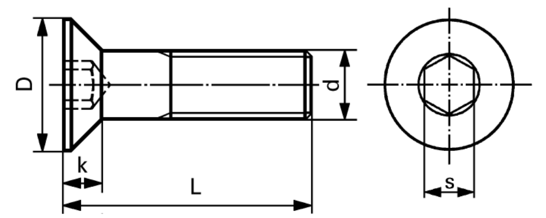
----- Below dash-line: fully or partially threaded at our option

Return to chapter index | Return to main fastener catalog index | Return to Maryland Metrics home page

10.9

Socket flat head screws, steel

DIN 7991 / ISO 10642



Zinc plated, baked (hydrogen embrittlement relieved)

d	M 3		M 4		M 5		M 6	
D	6		8		10		12	
s	2		2,5		3		4	
k max.	1,7		2,3		2,8		3,3	
k ISO max.	1,86		2,48		3,1		3,72	
L quantity	100	1000+	100	1000+	100	1000+	100	1000+
5	20.75	15.20						
6	19.20	14.10	18.20	13.40	17.50	13.50		
8	17.80	13.–	16.60	12.20	15.30	11.10	15.90	11.60
10	17.10	12.50	15.10	11.–	15.10	11.–	17.40	12.70
12	17.10	12.50	16.10	11.90	15.30	11.10	16.30	12.–
(14)	20.25	14.90	25.–	18.30	17.10	12.50	19.10	14.–
16	17.90	13.10	17.80	13.–	16.30	12.–	16.50	12.10
(18)	21.25	15.60	21.25	15.60	17.10	12.50	18.70	13.70
20	18.70	13.70	18.20	13.40	17.10	12.50	17.80	13.–
(22)			21.75	15.90	21.–	15.30	21.–	15.40
25	22.50	18.10	21.–	15.40	19.80	14.50	20.25	14.80
30	25.80	21.50	23.–	17.–	20.25	15.10	23.–	17.–
35	27.90	23.85	26.75	19.70	27.25	21.60	24.25	17.60
40	30.–	26.10	27.50	20.25	26.75	20.50	28.–	20.50
45			30.25	23.–	28.–	21.25	28.50	21.50
50			47.–	39.50	31.50	24.–	29.75	22.25
55					54.50	41.50	51.–	48.–
60					56.–	42.50	53.50	49.50
65							71.–	56.50
70							72.–	58.50
(75)							95.–	81.–
80					135.–	105.–	105.–	91.–
100							161.–	130.–

Reduced tightening and loads: refer to page T 25

Zinc plated high-strength steel: refer to page T 79

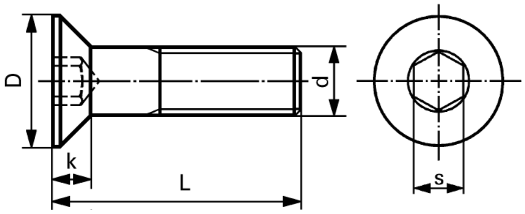
+ 10% discount for 10 000 pieces per item
----- Below dash-line: fully or partially threaded at our option

Return to chapter index | Return to main fastener catalog index | Return to Maryland Metrics home page

10.9

Socket flat head screws, steel

DIN 7991 / ISO 10642



Zinc plated, baked (hydrogen embrittlement relieved)

d	M 8		M 10		M 12	(M 14)	M 16	M 20
D	16		20		24	27	30	36
s	5		6		8	10	10	12
k max.	4,4		5,5		6,5	7	7,5	8,5
k ISO max.	4,96		6,2		7,44	8,4	8,8	10,16
L quantity	100	1000+	100	1000+	100	1000	100	1000
10	37.50	28.50						
12	34.75	26.50	89.–	73.80				
(14)	40.–	30.50	79.50	63.50				
16	23.25	17.10	66.50	50.50				
(18)	26.25	19.20	82.50	65.50				
20	23.75	17.40	46.50	35.25	64.–		250.–	
(22)	33.–	24.50			75.–			
25	25.50	18.70	49.50	37.75	66.50		230.–	
30	27.25	21.–	52.50	40.–	68.–	154.–	162.–	
35	30.50	23.25	59.–	45.–	71.–	161.–	165.–	
40	33.–	25.25	61.–	46.50	74.50	167.–	167.–	356.–
45	38.50	29.25	79.50	63.–	88.–	183.–	183.–	374.–
50	45.–	34.50	81.50	64.50	92.50	187.–	188.–	378.–
55	54.–	41.–	83.–	66.–	103.–	214.–	204.–	430.–
60	60.–	45.50	85.50	67.50	126.–		209.–	435.–
65			88.50	69.50			290.–	
70	83.–	67.50	113.–	90.–	146.–		268.–	495.–
(75)			125.–	102.–			290.–	
80	98.–	76.–	129.–	106.–	166.–		302.–	565.–
90	104.–	82.50	146.–	121.–	190.–		346.–	588.–
100	109.–	86.50	162.–	134.–	243.–		388.–	
110	130.–	105.–	179.–	151.–			460.–	
120			206.–	178.–	285.–			
130					320.–			
140							560.–	

Reduced tightening and loads: refer to page T 25

Zinc plated high-strength steel: refer to page T 79

+ 10% discount for 10 000 pieces per item

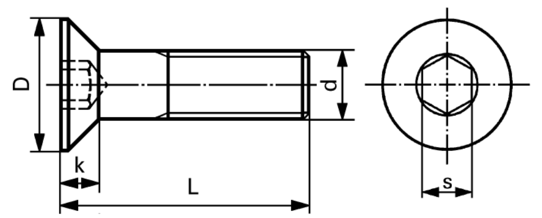
----- Below dash-line: fully or partially threaded at our option

Return to chapter index | Return to main fastener catalog index | Return to Maryland Metrics home page

10.9

Socket flat head screws, steel

DIN 7991 / ISO 10642



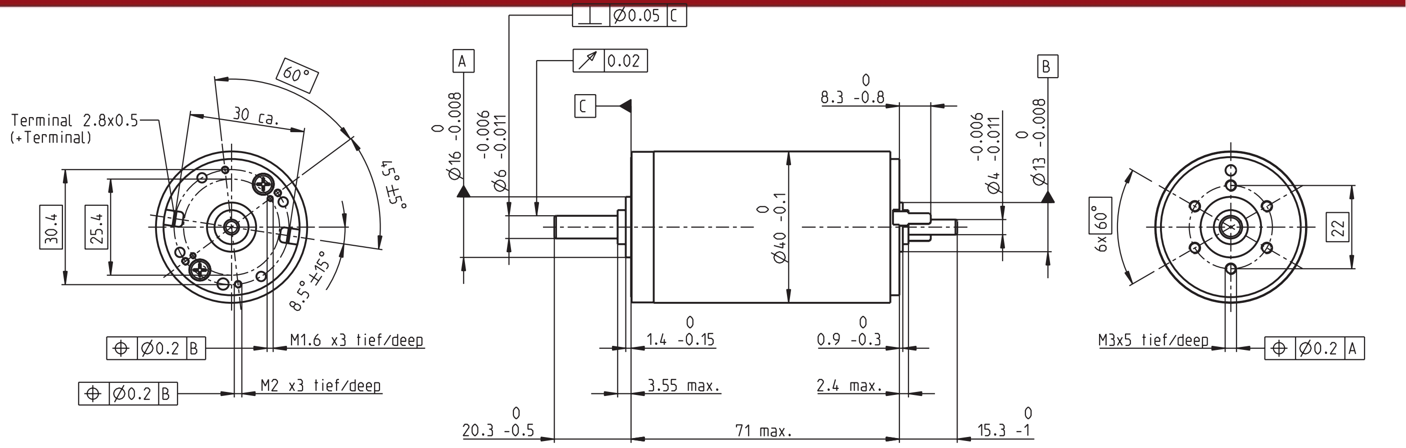
Dacromet plated

d	M 5		M 6		M 8		M 10		M 12
D	10		12		16		20		24
s	3		4		5		6		8
k max.	2,8		3,3		4,4		5,5		6,5
k ISO max.	3,1		3,72		4,96		6,2		7,44
L quantity	100	1000	100	1000	100	1000	100	1000	100
8	14.40	11.30							
10	15.80	12.50	17.–	13.50					
12	16.–	12.70	17.–	13.50					
(14)	21.25	16.90	17.60	14.10					
16	17.70	14.40	18.60	15.20	29.75	22.50			
(18)			20.50	17.45	31.50	24.60			
20	18.–	14.70	19.60	16.50	33.75	26.15	51.60	41.30	85.–
25	19.–	15.70	20.70	17.70	35.60	28.90	51.60	41.30	85.–
30	19.50	16.20	22.50	19.50	39.15	32.60	57.60	46.10	87.–
35	28.–	25.50	24.25	21.60	39.50	32.95	68.40	54.75	
40	48.–	42.60			40.20	33.60	75.60	60.50	91.–
45			35.–	30.60	42.–	35.25			
50			37.–	32.20	46.50	39.80	115.–	92.–	125.–
60			40.–	35.20	62.–	55.40	129.–	103.–	
65							140.–	112.–	
70			63.50	58.50			135.–	108.–	
80			102.–	88.60	130.–	104.–	145.–	116.–	

Reduced tightening and loads: refer to page T 25

----- Below dash-line: fully or partially threaded at our option

RE 40 Ø40 mm, Graphite Brushes, 150 Watt



M 1:2

🟥 Stock program
☐ Standard program
▨ Special program (on request)

Part Numbers

Motor Data

			148866	148867	148877	218008	218009	218010	218011	218012	218013	218014
	Values at nominal voltage											
1	Nominal voltage	V	12	24	48	48	48	48	48	48	48	48
2	No load speed	rpm	6920	7580	7590	6420	5560	3330	2690	2130	1720	1420
3	No load current	mA	241	137	68.6	53.6	43.7	21.9	16.6	12.5	9.66	7.76
4	Nominal speed	rpm	6380	6940	7000	5810	4930	2710	2060	1510	1080	781
5	Nominal torque (max. continuous torque)	mNm	94.9	177	187	186	180	189	190	192	192	190
6	Nominal current (max. continuous current)	A	6	6	3.17	2.66	2.23	1.4	1.13	0.909	0.73	0.6
7	Stall torque	mNm	1720	2420	2560	2040	1620	1020	814	655	523	424
8	Starting current	A	105	80.2	42.4	28.6	19.7	7.43	4.79	3.06	1.97	1.32
9	Max. efficiency	%	87	91	92	91	91	89	89	88	87	85
	Characteristics											
10	Terminal resistance	Ω	0.115	0.299	1.13	1.68	2.44	6.46	10	15.7	24.4	36.3
11	Terminal inductance	mH	0.0245	0.0823	0.329	0.46	0.612	1.7	2.62	4.14	6.4	9.31
12	Torque constant	mNm/A	16.4	30.2	60.3	71.3	82.2	137	170	214	266	321
13	Speed constant	rpm/V	581	317	158	134	116	69.7	56.2	44.7	35.9	29.8
14	Speed / torque gradient	rpm/mNm	4.05	3.14	2.97	3.16	3.45	3.29	3.31	3.27	3.29	3.37
15	Mechanical time constant	ms	5.89	4.67	4.28	4.2	4.19	4.16	4.15	4.15	4.15	4.16
16	Rotor inertia	gcm^2	139	142	137	127	116	121	120	121	120	118

Specifications

Thermal data

17	Thermal resistance housing-ambient	4.7 K/W
18	Thermal resistance winding-housing	1.9 K/W
19	Thermal time constant winding	41.5 s
20	Thermal time constant motor	736 s
21	Ambient temperature	-30…+100°C
22	Max. permissible winding temperature	+155°C

Mechanical data (ball bearings)

23	Max. permissible speed	12000 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	5.6 N
27	Max. force for press fits (static)	110 N
	(static, shaft supported)	1200 N
28	Max. radial loading, 5 mm from flange	28 N

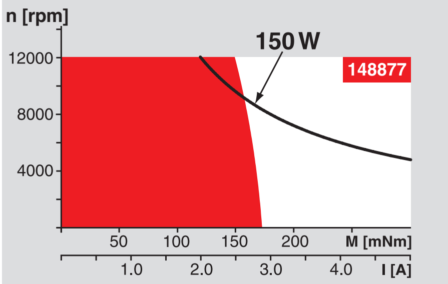
Other specifications

29	Number of pole pairs	1
30	Number of commutator segments	13
31	Weight of motor	480 g

Values listed in the table are nominal.
Explanation of the figures on page 71.

Option
Preloaded ball bearings

Operating Range



Comments

🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

☐ **Short term operation**
The motor may be briefly overloaded (recurring).

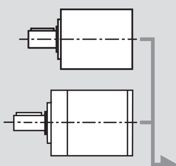
— **Assigned power rating**

maxon Modular System

Overview on page 20 - 25

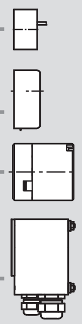
Planetary Gearhead
Ø42 mm
3 - 15 Nm
Page 270

Planetary Gearhead
Ø52 mm
4 - 30 Nm
Page 273



Recommended Electronics:

ESCON 50/5	Page 321
ESCON Module 50/5	321
ESCON 70/10	321
EPOS2 24/5	331
EPOS2 50/5	331
EPOS2 70/10	331
EPOS2 P 24/5	334
EPOS3 70/10 EtherCAT	337
Notes	**22**



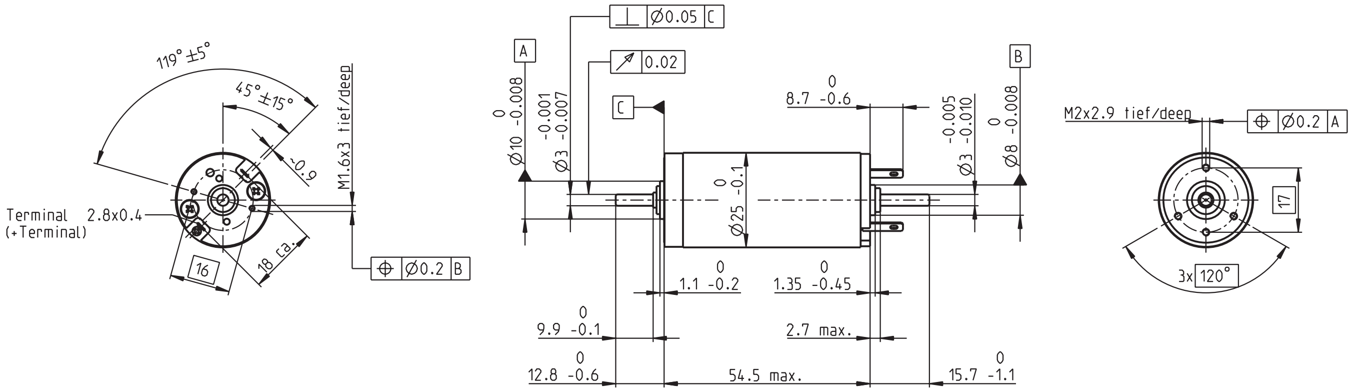
Encoder MR
256 - 1024 CPT,
3 channels
Page 303

Encoder HED_ 5540
500 CPT,
3 channels
Page 305/307

Brake AB 28
24 VDC
0.4 Nm
Page 348

Industrial Version
Encoder HEDL 9140
Page 310
Brake AB 28
Page 349
End cap
Page 353

RE 25 ∅25 mm, Precious Metal Brushes CLL, 10 Watt



M 1:2

🟥 Stock program
☐ Standard program
▨ Special program (on request)

Part Numbers

Motor Data

			118740	118741	118742	118743	118744	118745	118746	118747	118748
	Values at nominal voltage										
1	Nominal voltage	V	4.5	8	9	12	15	18	24	32	48
2	No load speed	rpm	5360	5320	5230	4850	4980	4790	5190	5510	5070
3	No load current	mA	79.7	44.4	38.7	26.3	21.8	9.88	14.4	11.7	6.96
4	Nominal speed	rpm	4980	4520	4220	3800	3920	3710	4130	4450	4000
5	Nominal torque (max. continuous torque)	mNm	11.4	20.9	23.9	28.6	28.2	28.7	28	27.9	27.9
6	Nominal current (max. continuous current)	A	1.5	1.5	1.5	1.24	1.01	0.811	0.652	0.516	0.317
7	Stall torque	mNm	131	132	119	129	131	126	136	144	132
8	Starting current	A	16.5	9.23	7.31	5.5	4.57	3.52	3.1	2.61	1.47
9	Max. efficiency	%	87	87	86	87	87	90	87	87	87
	Characteristics										
10	Terminal resistance	Ω	0.273	0.867	1.23	2.18	3.28	5.11	7.73	12.3	32.6
11	Terminal inductance	mH	0.0275	0.0882	0.115	0.238	0.353	0.551	0.832	1.31	3.48
12	Torque constant	mNm/A	7.99	14.3	16.3	23.5	28.6	35.8	43.9	55.2	89.9
13	Speed constant	rpm/V	1200	668	584	406	334	267	217	173	106
14	Speed / torque gradient	rpm/mNm	40.9	40.5	44	37.7	38.3	38.2	38.3	38.5	38.6
15	Mechanical time constant	ms	4.99	4.4	4.37	4.25	4.23	4.22	4.22	4.22	4.23
16	Rotor inertia	gcm^2	11.7	10.4	9.49	10.8	10.6	10.6	10.5	10.5	10.5

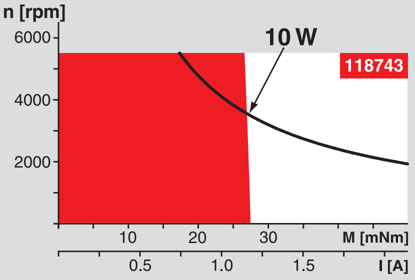
Specifications

	Thermal data	
17	Thermal resistance housing-ambient	14 K/W
18	Thermal resistance winding-housing	3.1 K/W
19	Thermal time constant winding s	12.5 s
20	Thermal time constant motor	612 s
21	Ambient temperature	-20…+85°C
22	Max. permissible winding temperature	+100°C
	Mechanical data (ball bearings)	
23	Max. permissible speed	5500 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	3.2 N
27	Max. force for press fits (static)	64 N
	(static, shaft supported)	800 N
28	Max. radial load, 5 mm from flange	16 N
	Other specifications	
29	Number of pole pairs	1
30	Number of commutator segments	11
31	Weight of motor	130 g
	CLL = Capacitor Long Life	

Values listed in the table are nominal.
Explanation of the figures on page 79.

Option
Preloaded ball bearings

Operating Range



Comments

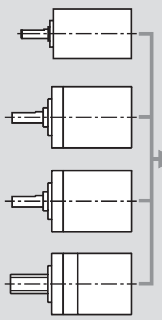
🟥 **Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

☐ **Short term operation**
The motor may be briefly overloaded (recurring).

— **Assigned power rating**

maxon Modular System

Overview on page 20–25

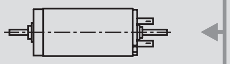


Planetary Gearhead
∅26 mm
0.75 - 4.5 Nm
Page 270

Planetary Gearhead
∅32 mm
0.75 - 6.0 Nm
Page 272/273/276

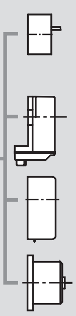
Koaxdrive
∅32 mm
1.0 - 4.5 Nm
Page 281

Spindle Drive
∅32 mm
Page 301–303



Recommended Electronics:

ESCON 36/2 DC	Page 342
ESCON Module 50/5	343
ESCON 50/5	344
ESCON 70/10	344
EPOS2 24/2	350
EPOS2 Module 36/2	350
EPOS2 24/5, EPOS2 50/5	351
EPOS2 P 24/5	354
EPOS3 70/10 EtherCAT	357
MAXPOS 50/5	360
Notes	**22**



Encoder MR
128 - 1000 CPT,
3 channels
Page 319

Encoder Enc
22 mm
100 CPT, 2 channels
Page 324

Encoder HED_ 5540
500 CPT,
3 channels
Page 325/327

DC-Tacho DCT
∅22 mm
0.52 V
Page 336

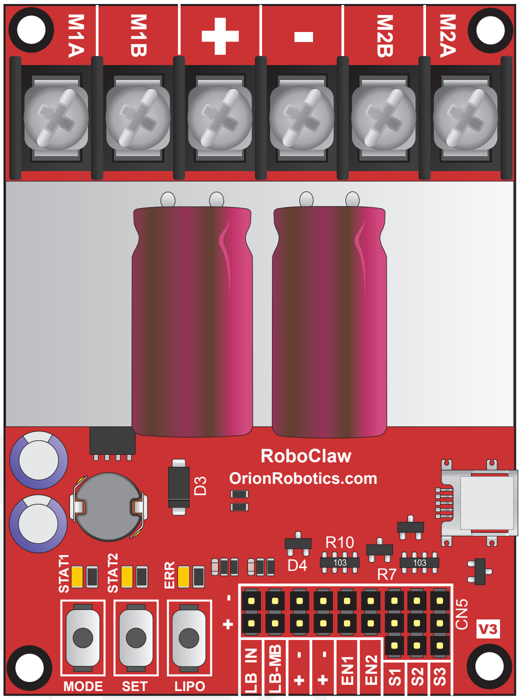
ORION

ROBOTICS

RoboClaw 2x30A Motor Controller
Data Sheet
Firmware Version 3.1.3+

Feature Overview:

- 2 Channels at 30Amp each, Peak 60Amp
- 3.3V Compliant Outputs
- 5V Tolerant Inputs
- Battery Elimination Circuit (BEC)
- Switching Mode BEC
- Hobby RC Radio Compatible
- Serial Modes
- TTL Input
- Analog Mode
- 2 Channel Quadrature Decoding
- Thermal Protection
- Lithium Cut Off
- Packet Serial with Error Detection
- High Speed Direction Switching
- Flip Over Switch
- Over Current Protection
- Regenerative Braking
- USB Capable(Optional)



Basic Description

The RoboClaw 2X30 Amp is an extremely efficient, versatile, dual channel synchronous regenerative motor controller. It supports dual quadrature encoders and can supply two brushed DC motors with 30 amps per channel continuous and 60 amp peak.

With support for dual quadrature decoding you get greater control over speed and velocity. Automatically maintain a speed even if load increases. RoboClaw uses PID calculations with feed forward in combination with external quadrature encoders to make an accurate control solution.

RoboClaw is easy to control with several built in modes. It can be controlled from a standard RC receiver/transmitter, serial device, microcontroller or an analog source, such as a potentiometer based joystick. RoboClaw is equipped with screw terminal for fast connect and disconnect. All modes are set by the onboard mode buttons making setup a snap!

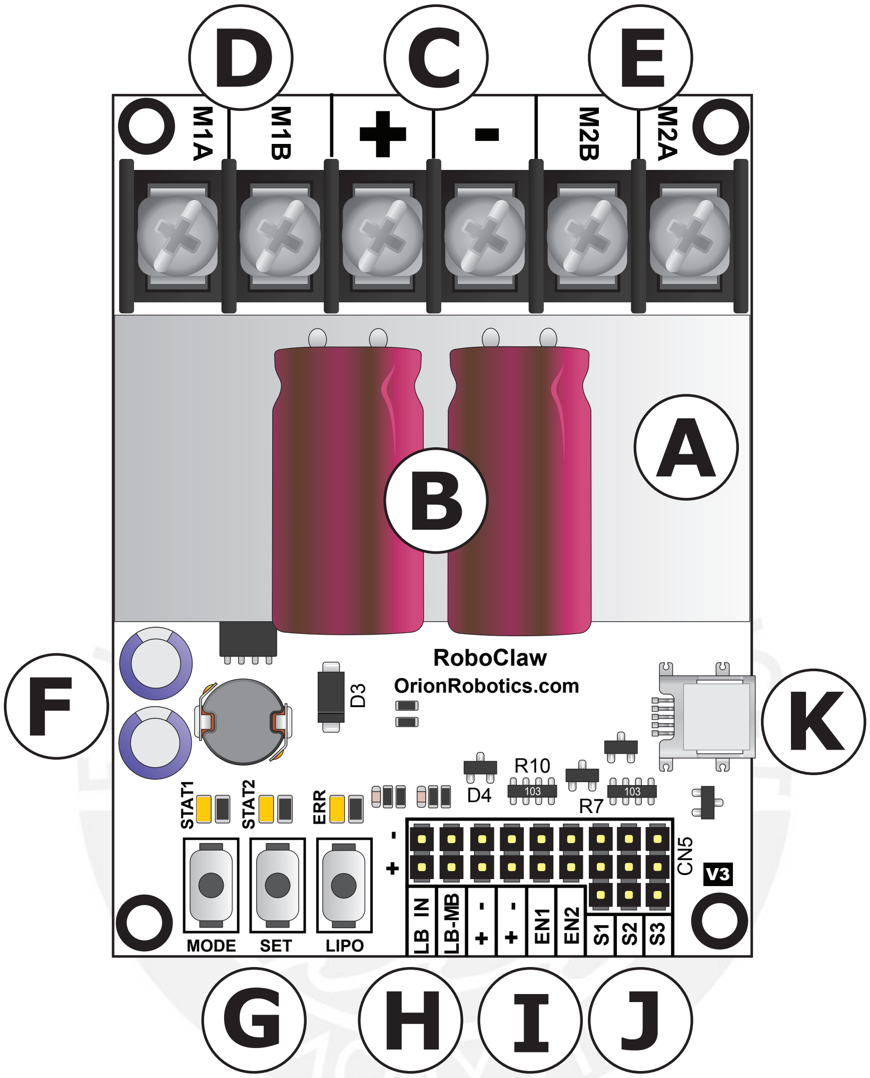
Optical Encoders

RoboClaw features dual channel quadrature decoding. RoboClaw gives you the ability to create a closed loop motion system. Now you can know a motors speed and direction giving you greater control over DC motors systems.

Power System

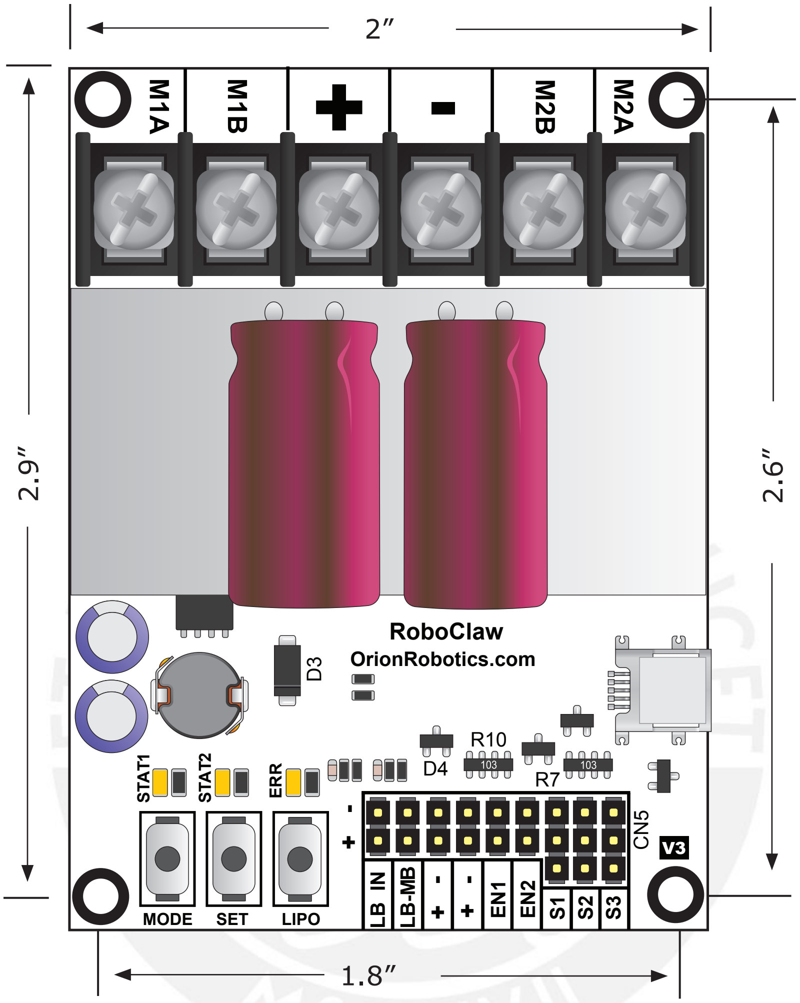
The RoboClaw is equipped with synchronous regenerative motor drivers. This means your battery is recharged when slowing down, braking or reversing. In addition a switching mode BEC is included. It can supply a useful current of up to 3Amps at 5v. The BEC is meant to provide power to a microcontroller or RC receiver.

Hardware Overview:



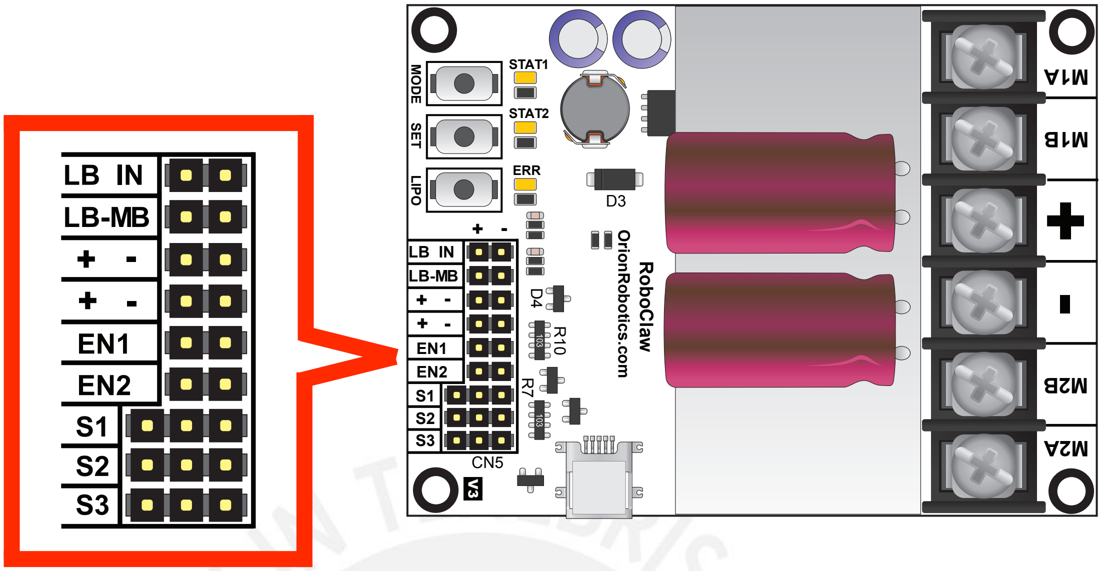
A: Heat Sink
B: Power Stabilizers
C: Main Battery Input
D: Motor Channel 1
E: Motor Channel 2
F: BEC 3A Circuit
G: Setup Buttons
H: Logic Voltage Source Selection Header
I: Encoder Inputs
J: Controller Inputs
K: USB Connector - MiniB (Optional)

Dimensions:



Board Edge: 2″W X 2.9″L
Hole Pattern: 0.125D, 1.8″W x 2.6″H

Header Overview



Logic Battery (LB IN)
The logic circuits can be powered from the main battery or a secondary battery wired LB IN. The positive (+) terminal is located at the board edge and ground (-) is the inside pin near the heatsink. Remove LB-MB jumper if power is applied to LB IN.

BEC Source (LB-MB)
RoboClaw logic requires 5VDC which is provided from the on board BEC circuit. The BEC source input is set with the LB-MB jumper. Install a jumper on the 2 pins labeled LB-MB to use the main battery as the BEC power source. Remove this jumper if using a separate logic battery.

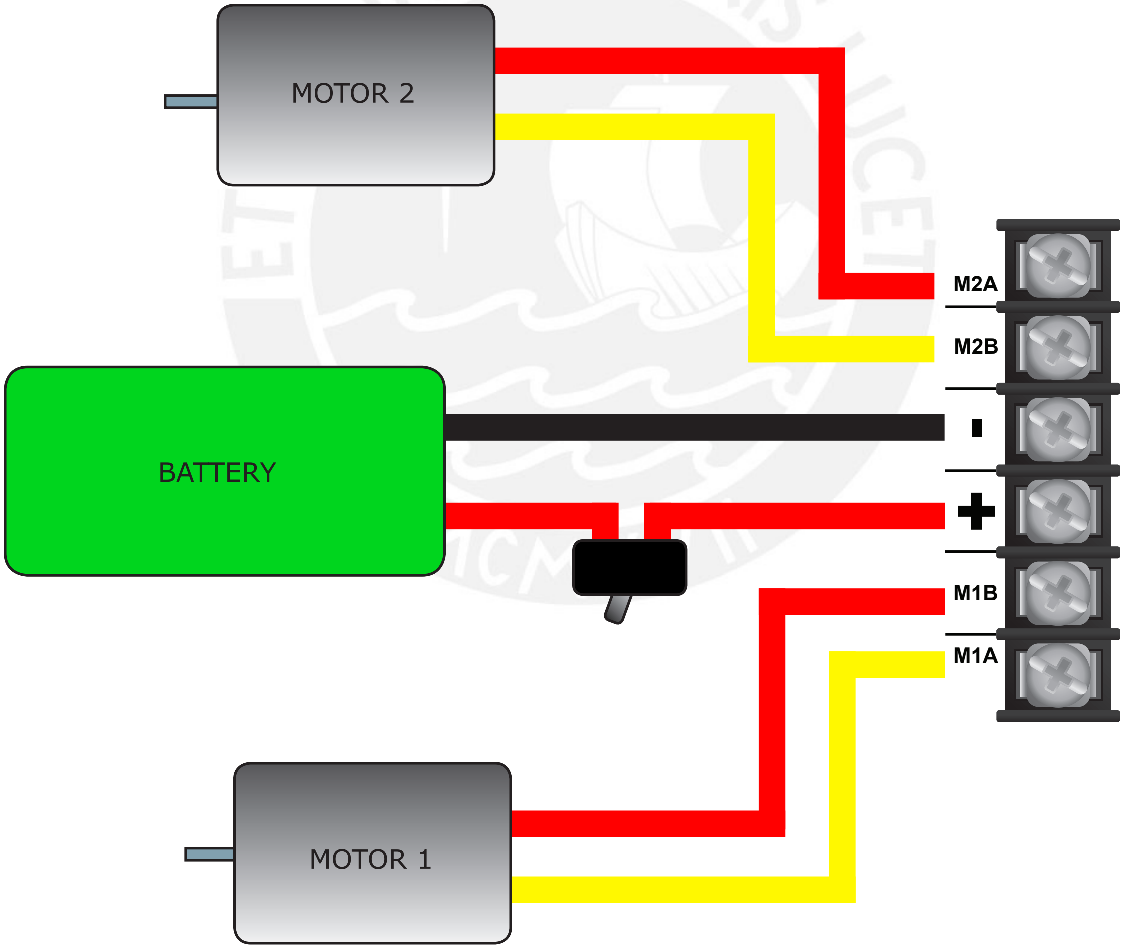
Encoder Power (+ -)
The pins labeled + and - are the source power pins for encoders. The positive (+) is located at the board edge and supplies +5VDC. The ground (-) pin is near the heatsink.

Encoder Inputs (EN1 / EN2)
EN1 and EN2 are the inputs from the encoders. Channel A of both EN1 and EN2 are located at the board edge. Channel B pins are located near the heatsink. When connecting the encoder make sure the leading channel for the direction of rotation is connected to A. If one encoder is backwards to the other you will have one internal counter counting up and the other counting down. Which can affect how RoboClaw operates. Refer to the data sheet of the encoder you are using for channel direction.

Control Inputs (S1 / S2 / S3)
S1, S2 and S3 are setup for standard servo style headers I/O, +5V and GND. S1 and S2 are the control inputs for serial, analog and RC modes. S3 can be used as a flip switch input when in RC or Analog modes. In serial mode S3 becomes an emergency stop. S3 is active when pulled low. It is internally pull up so it will not accidentally trip when left floating. The pins closest to the board edge are the I/0s, center pin is the +5V and the inside pins are ground. Some RC receivers have their own supply and will conflict with the RoboClaw's logic supply. It may be necessary to remove the +5V pin from the RC receivers cable in those cases.

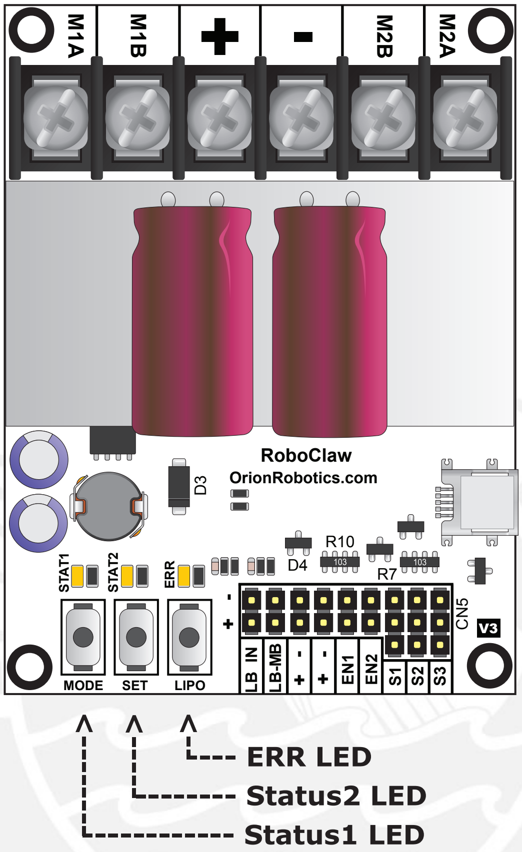
Main Battery Screw Terminals
RoboClaws main power input can be from 6VDC to 34VDC. The connections are marked + and - on the main screw terminal. + is the positive side typically marked with a red wire. The - is the negative side typically marked with a black wire. When connecting the main battery it is a good practice to use a switch to turn the main power on and off. The switch must be rated to handle the maximum current and voltage from the battery. This will vary depending on the type of motors and or power source you are using.

Motor Screw Terminals
The motor screw terminals are marked with M1A / M1B for channel 1 and M2A / M2B for channel 2. There is no specific polarities for the motors. However if you want both motors turning in the same direction on a 4 wheeled robot you need to reverse one of the motors polarities.



Status and Error LEDs

The RoboClaw has three LEDs. Two Status LEDs and one Error LED. When RoboClaw is first powered up all 3 LEDs should blink briefly to indicate all 3 LEDs are functional. The status LEDs will indicate a status based on what mode RoboClaw is set to.



ERR LED

Status2 LED

Status1 LED

`

Analog Mode

Status 1 LED = On continuous.

Status 2 LED = On when motor(s) active.

RC Mode

Status 1 LED = On continuous, blink when pulse received.

Status 2 LED = On when motor(s) active.

Serial Modes

Status 1 LED = On continuous, blink on serial receive.

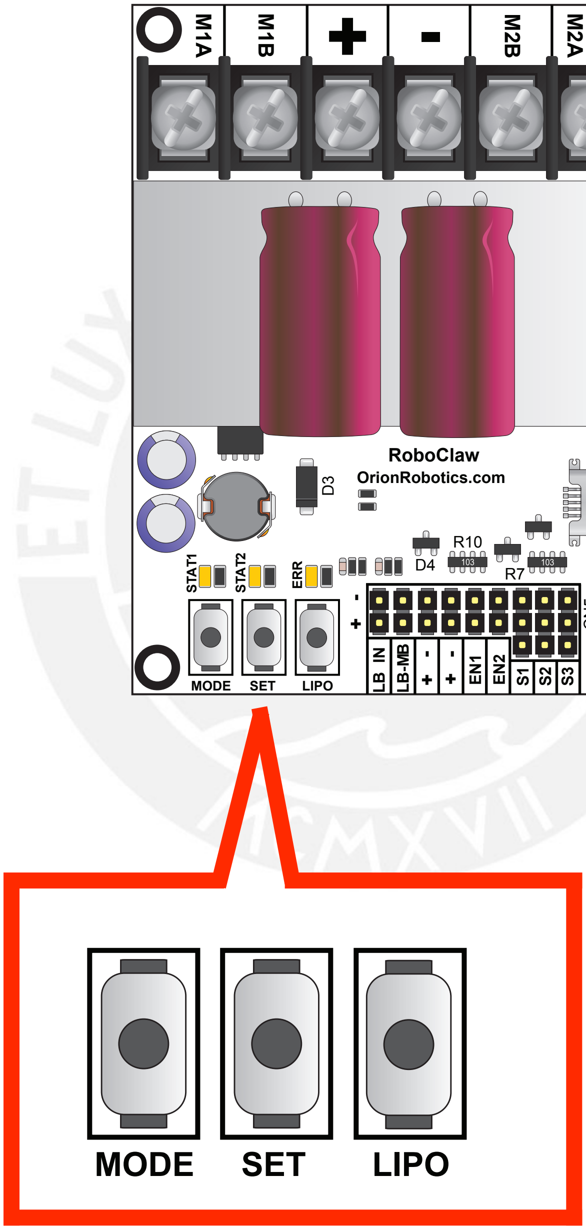
Status 2 LED = On when motor(s) active.

Errors

Over Current = Error LED on solid. Status 1 or 2 indicates which motor.
Over Heat = Error LED blinking once with a long pause. Status 1 & 2 off
Driver Error = Error LED blinking once with a long pause. Status 1 or 2 on
Main Batt Low = Error LED blinking twice with a long pause.
Main Batt High = Error LED on/flicker until condition is cleared.
Logic Batt Low = Error LED blinking three times with a long pause.
Logic Batt High = Error LED blinking four times with a long pause.

RoboClaw Setup

There are 3 buttons on RoboClaw which are used to set modes and configuration options. The MODE button sets the interface method such as Serial or RC modes. The SET button is used to configure the options for a given mode. The LIPO button doubles as a save button and configuring the low battery voltage cut out function of RoboClaw. See the following tables to navigate RoboClaw setup.



Interface Overview

There are 4 main modes with variations totaling 14 or 15 modes in all. Each mode enables RoboClaw to be controlled in a very specific way. The following list explains each mode and the ideal application.

RC Mode 1 & 2

With RC mode RoboClaw can be controlled from any hobby RC radio system. RC input mode also allows low powered microcontroller such as a Basic Stamp or Nano to control RoboClaw. RoboClaw expects servo pulse inputs to control the direction and speed. Very similar to how a regular servo is controlled. RC mode can not use encoders.

Analog Mode 3 & 4

Analog mode uses an analog signal from 0V to 5V to control the speed and direction of each motor. RoboClaw can be controlled using a potentiometer or filtered PWM from a microcontroller. Analog mode is ideal for interfacing RoboClaw joystick positioning systems or other non microcontroller interfacing hardware. Analog mode can not use encoders.

Simple Serial Mode 5 & 6

In simple serial mode RoboClaw expects TTL level RS-232 serial data to control direction and speed of each motor. Simple serial is typically used to control RoboClaw from a microcontroller or PC. If using a PC a MAX232 type circuit must be used since RoboClaw only works with TTL level input. Simple serial includes a slave select mode which allows multiple RoboClaws to be controlled from a signal RS-232 port (PC or microcontroller). Simple serial is a one way format, RoboClaw only receives data.

Packet Serial Mode 7 through 14

In packet serial mode RoboClaw expects TTL level RS-232 serial data to control direction and speed of each motor. Packet serial is typically used to control RoboClaw from a microcontroller or PC. If using a PC a MAX232 type circuit must be used since RoboClaw only works with TTL level input. In packet serial mode each RoboClaw is assigned an address using the dip switches. There are 8 addresses available. This means up to 8 RoboClaws can be on the same serial port. When using the quadrature decoding feature of RoboClaw packet serial is required since it is a two way communications format. This allows RoboClaw to transmit information about the encoders position and speed.

USB Mode 15(USB Roboclaw only)

In USB mode the RoboClaw's USB port acts as a CDC Virtual Comport in Packet Serial mode with packet address 128. Packet serial mode functionality is available in USB mode as well as baud rates up to 1mbit. There are two ways to activate the USB mode. Power up a USB RoboClaw while it is attached to an active USB cable, or set it to mode 15. If a PC is used to drive RoboClaw mode 15 should be set.

Configuring RoboClaw Modes

The buttons built into RoboClaw are used to set the different configuration options. To set the desired mode follow the steps below:

1. Press and release the MODE button to enter mode setup. The STAT2 LED will begin to blink out the current mode. Each blink is a half second with a long pause at the end of the count. Five blinks with a long pause equals mode 5 and so on.

2. Press SET to increment to the next mode. Press MODE to decrement to the previous mode.

3. Press and release the LIPO button to save this mode to memory.

Modes

Mode	Description
1	RC mode
2	RC mode with mixing
3	Analog mode
4	Analog mode with mixing
5	Simple Serial
6	Simple Serial with slave pin
7	Packet Serial Mode - Address 0x80
8	Packet Serial Mode - Address 0x81
9	Packet Serial Mode - Address 0x82
10	Packet Serial Mode - Address 0x83
11	Packet Serial Mode - Address 0x84
12	Packet Serial Mode - Address 0x85
13	Packet Serial Mode - Address 0x86
14	Packet Serial Mode - Address 0x87
15	USB Mode Packet Serial - Address 0x80

Mode Options

After the desired mode is set and saved press and release the SET button for options setup. The STAT2 LED will begin to blink out the current option. Press SET to increment to the next option. Press MODE to decrement to the previous option. Once the desired option is selected press and release the LIPO button to save the option to memory.

RC and Analog Mode Options

Option	Description
1	TTL Flip Switch
2	TTL Flip and Exponential Enabled
3	TTL Flip and MCU Enabled
4	TTL Flip and Exp and MCU Enabled
5	RC Flip Switch
6	RC Flip and Exponential Enabled
7	RC Flip and MCU Enabled
8	RC Flip and Exponential and MCU Enabled

Simple and Packet Serial Mode Options

Option	Description
1	2400bps
2	9600bps
3	19200bps
4	38400bps

Battery Cut Off Settings

The battery settings can be set by pressing and releasing the LIPO button. The STAT2 LED will begin to blink out the current setting. Press SET to increment to the next setting. Press MODE to decrement to the previous setting. Once the desired setting is selected press and release the LIPO button to save this setting to memory.

Battery Options

Option	Description
1	Normal
2	Lead Acid - Auto
3	2 Cell(6v Cutoff)
4	3 Cell(9v Cutoff)
5	4 Cell(12v Cutoff)
6	5 Cell(15v Cutoff)
7	6 Cell(18v Cutoff)
8	7 Cell(21v Cutoff)

Packet Serial

Packet Serial Mode

Packet serial is a buffered bidirectional serial mode. More sophisticated instructions can be sent to RoboClaw. The basic command structures consists of an address byte, command byte, data bytes and a checksum. The amount of data each command will send or receive can vary.

Address

Packet serial requires a unique address. With up to 8 addresses available you can have up to 8 RoboClaws bussed on the same RS232 port. There are 8 packet modes 7 to 14. Each mode has a unique address. The address is selected by setting the desired packet mode using the MODE button.

Packet Modes

Mode	Description
7	Packet Serial Mode - Address 0x80 (128)
8	Packet Serial Mode - Address 0x81 (129)
9	Packet Serial Mode - Address 0x82 (130)
10	Packet Serial Mode - Address 0x83 (131)
11	Packet Serial Mode - Address 0x84 (132)
12	Packet Serial Mode - Address 0x85 (133)
13	Packet Serial Mode - Address 0x86 (134)
14	Packet Serial Mode - Address 0x87 (135)

Packet Serial Baud Rate

When in serial mode or packet serial mode the baud rate can be changed to one of four different settings in the table below. These are set using the SET button as covered in Mode Options.

Serial Mode Options

Option	Description
1	2400
2	9600
3	19200
4	38400

Checksum Calculation

All packet serial commands use a 7 bit checksum to prevent corrupt commands from being executed. Since the RoboClaw expects a 7bit value the 8th bit is masked. The checksum is calculated as follows:

Checksum = (Address + Command + Data bytes) & 0x7F

When calculating the checksum all data bytes sent or received must be added together. The hexadecimal value 0X7F is used to mask the 8th bit.

Commands 0 - 7 Standard Commands

The following commands are the standard set of commands used with packet mode. The command syntax is the same for commands 0 to 7:

Address, Command, ByteValue, Checksum

0 - Drive Forward M1

Drive motor 1 forward. Valid data range is 0 - 127. A value of 127 = full speed forward, 64 = about half speed forward and 0 = full stop. Example with RoboClaw address set to 128:

```
Send: 128, 0, 127, ((128+0+127) & 0X7F)
```

1 - Drive Backwards M1

Drive motor 1 backwards. Valid data range is 0 - 127. A value of 127 full speed backwards, 64 = about half speed backward and 0 = full stop. Example with RoboClaw address set to 128:

```
Send: 128, 1, 127, ((128+0+127) & 0X7F)
```

2 - Set Minimum Main Voltage

Sets main battery (B- / B+) minimum voltage level. If the battery voltages drops below the set voltage level RoboClaw will shut down. The value is cleared at start up and must set after each power up. The voltage is set in .2 volt increments. A value of 0 sets the minimum value allowed which is 6V. The valid data range is 0 - 120 (6V - 30V). The formula for calculating the voltage is: (Desired Volts - 6) x 5 = Value. Examples of valid values are 6V = 0, 8V = 10 and 11V = 25. Example with RoboClaw address set to 128:

```
Send: 128, 2, 25, ((128+2+25) & 0X7F)
```

3 - Set Maximum Main Voltage

Sets main battery (B- / B+) maximum voltage level. The valid data range is 0 - 154 (0V - 30V). If you are using a battery of any type you can ignore this setting. During regenerative breaking a back voltage is applied to charge the battery. When using an ATX type power supply if it senses anything over 16V it will shut down. By setting the maximum voltage level, RoboClaw before exceeding it will go into hard breaking mode until the voltage drops below the maximum value set. The formula for calculating the voltage is: Desired Volts x 5.12 = Value. Examples of valid values are 12V = 62, 16V = 82 and 24V = 123. Example with RoboClaw address set to 128:

```
Send: 128, 3, 82, ((128+3+82) & 0X7F)
```

4 - Drive Forward M2

Drive motor 2 forward. Valid data range is 0 - 127. A value of 127 full speed forward, 64 = about half speed forward and 0 = full stop. Example with RoboClaw address set to 128:

```
Send: 128, 4, 127, ((128+4+127) & 0X7F)]
```

5 - Drive Backwards M2

Drive motor 2 backwards. Valid data range is 0 - 127. A value of 127 full speed backwards, 64 = about half speed backward and 0 = full stop. Example with RoboClaw address set to 128:

```
Send: 128, 5, 127, ((128+5+127) & 0X7F)
```

6 - Drive M1 (7 Bit)

Drive motor 1 forward and reverse. Valid data range is 0 - 127. A value of 0 = full speed reverse, 64 = stop and 127 = full speed forward. Example with RoboClaw address set to 128:

```
Send: 128, 6, 96, ((128+6+96) & 0X7F)
```

7 - Drive M2 (7 Bit)

Drive motor 2 forward and reverse. Valid data range is 0 - 127. A value of 0 = full speed reverse, 64 = stop and 127 = full speed forward. Example with RoboClaw address set to 128:

```
Send: 128, 7, 32, ((128+7+32) & 0X7F)
```

Commands 8 - 13 Mix Mode Commands

The following commands are mix mode commands and used to control speed and turn. Before a command is executed valid drive and turn data is required. You only need to send both data packets once. After receiving both valid drive and turn data RoboClaw will begin to operate. At this point you only need to update turn or drive data.

8 - Drive Forward

Drive forward in mix mode. Valid data range is 0 - 127. A value of 0 = full stop and 127 = full forward. Example with RoboClaw address set to 128:

```
Send: 128, 8, 127, ((128+8+127) & 0x7F)
```

9 - Drive Backwards

Drive backwards in mix mode. Valid data range is 0 - 127. A value of 0 = full stop and 127 = full reverse. Example with RoboClaw address set to 128:

```
Send: 128, 9, 127, ((128+9+127) & 0x7F)
```

10 - Turn right

Turn right in mix mode. Valid data range is 0 - 127. A value of 0 = stop turn and 127 = full speed turn. Example with RoboClaw address set to 128:

```
Send: 128, 10, 127, ((128+10+127) & 0x7F1)
```

11 - Turn left

Turn left in mix mode. Valid data range is 0 - 127. A value of 0 = stop turn and 127 = full speed turn. Example with RoboClaw address set to 128:

```
Send: 128, 11, 127, ((128+11+127) & 0x7F)
```

12 - Drive Forward or Backward (7 Bit)

Drive forward or backwards. Valid data range is 0 - 127. A value of 0 = full backward, 64 = stop and 127 = full forward. Example with RoboClaw address set to 128:

```
Send: 128, 12, 96, ((128+12=96) & 0x7F)
```

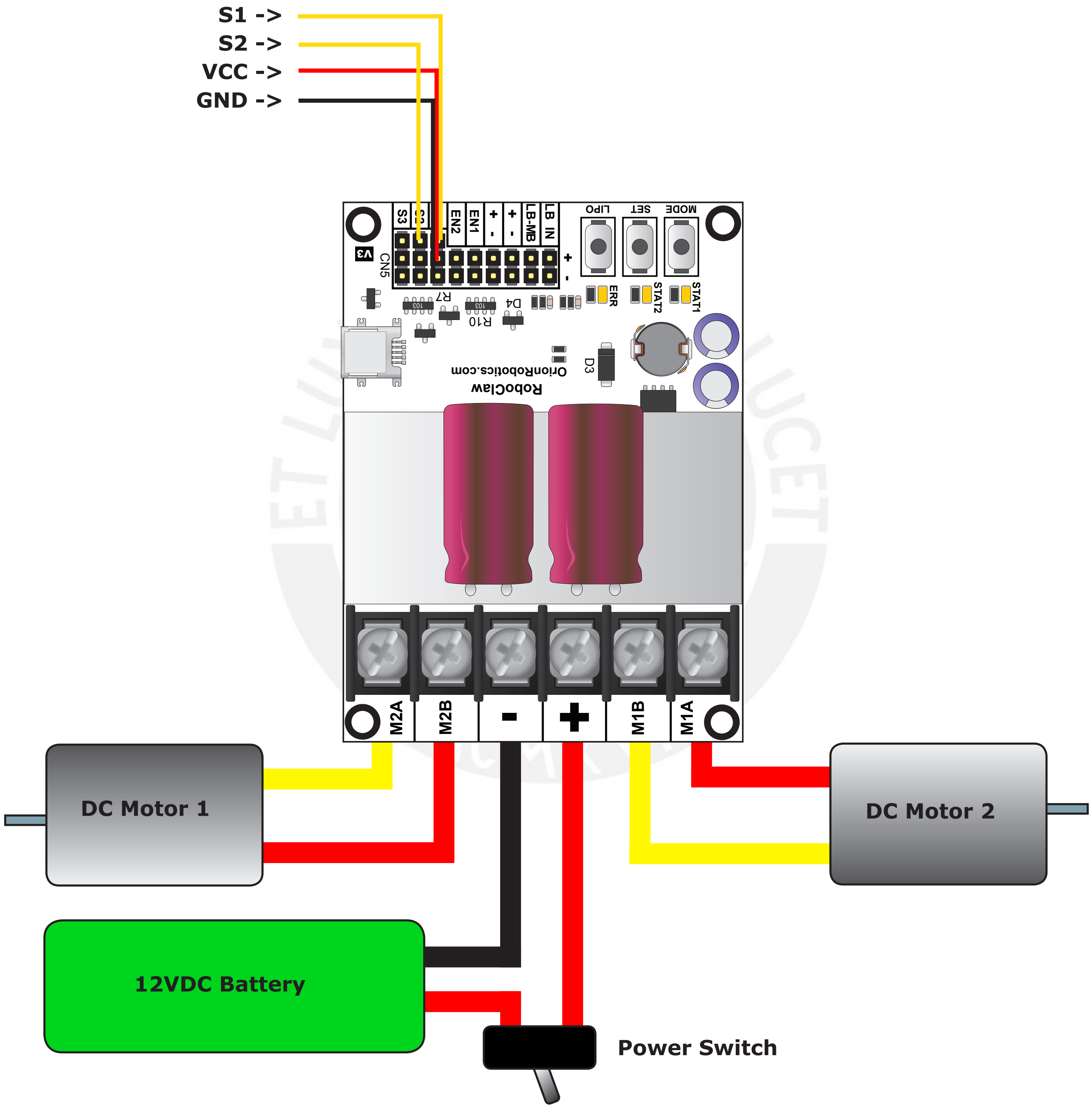
13 - Turn Left or Right (7 Bit)

Turn left or right. Valid data range is 0 - 127. A value of 0 = full left, 0 = stop turn and 127 = full right. Example with RoboClaw address set to 128:

```
Send: 128, 13, 0, ((128+13=0) & 0x7F)
```

Packet Serial Wiring

In packet mode the RoboClaw can transmit and receive serial data. A microcontroller with a UART is recommended. The UART will buffer the data received from RoboClaw. When a request for data is made to RoboClaw the return data will always have at least a 1ms delay after the command is received. This will allow slower processors and processors without UARTs to communicate with RoboClaw.



Packet Serial - Arduino Example

The example will start the motor channels independently. Then start turns with mix mode commands. The program was written and tested with a Arduno Uno and P5 connected to S1. Set mode 7 and option 3.

```
//Basic Micro RoboClaw Packet Serial Test Commands 0 to 13.
//Switch settings: SW3=ON and SW5=ON.

#include “BMSerial.h”
#include “RoboClaw.h”

#define address 0x80

RoboClaw roboclaw(5,6);

void setup() {
  roboclaw.begin(19200);
}

void loop() {
  roboclaw.ForwardM1(address,64);  //Cmd 0
  roboclaw.BackwardM2(address,64); //Cmd 5
  delay(2000);
  roboclaw.BackwardM1(address,64); //Cmd 1
  roboclaw.ForwardM2(address,64);  //Cmd 6
  delay(2000);
  roboclaw.ForwardBackwardM1(address,96);      //Cmd 6
  roboclaw.ForwardBackwardM2(address,32);      //Cmd 7
  delay(2000);
  roboclaw.ForwardBackwardM1(address,32);      //Cmd 6
  roboclaw.ForwardBackwardM2(address,96);      //Cmd 7
  delay(2000);

  //stop motors
  roboclaw.ForwardBackwardM1(address,0);
  roboclaw.ForwardBackwardM2(address,0);

  delay(10000);

  roboclaw.ForwardMixed(address, 64);    //Cmd 8
  delay(2000);
  roboclaw.BackwardMixed(address, 64);   //Cmd 9
  delay(2000);
  roboclaw.TurnRightMixed(address, 64);  //Cmd 10
  delay(2000);
  roboclaw.TurnLeftMixed(address, 64);         //Cmd 11
  delay(2000);
  roboclaw.ForwardBackwardMixed(address, 32);  //Cmd 12
  delay(2000);
  roboclaw.ForwardBackwardMixed(address, 96);  //Cmd 12
  delay(2000);
  roboclaw.LeftRightMixed(address, 32);  //Cmd 13
  delay(2000);
  roboclaw.LeftRightMixed(address, 96);  //Cmd 13
  delay(2000);

  //stop motors
  roboclaw.ForwardMixed(address, 0);

  delay(10000);
}
```

Advanced Packet Serial

Version, Status, and Settings Commands

The following commands are used to read board status, version information and set configuration values.

Command	Description
21	Read Firmware Version
24	Read Main Battery Voltage
25	Read Logic Battery Voltage
26	Set Minimum Logic Voltage Level
27	Set Maximum Logic Voltage Level
49	Read Motor Currents
55	Read Motor 1 Velocity PID Constants
56	Read Motor 2 Velocity PID Constants
57	Set Main Battery Voltages
58	Set Logic Battery Voltages
59	Read Main Battery Voltage Settings
60	Read Logic Battery Voltage Settings
63	Read Motor 1 Position PID Constants
64	Read Motor 2 Position PID Constants
82	Read Temperature
90	Read Error Status
91	Read Encoder Mode
92	Set Motor 1 Encoder Mode
93	Set Motor 1 Encoder Mode
94	Write Settings to EEPROM

21 - Read Firmware Version

Read RoboClaw firmware version. Returns up to 32 bytes and is terminated by a null character. Command syntax:

```
Send: [Address, 21]
Receive: ["RoboClaw 10.2A v1.3.9, Checksum]
```

The command will return up to 32 bytes. The return string includes the product name and firmware version. The return string is terminated with a null (0) character.

24 - Read Main Battery Voltage Level

Read the main battery voltage level connected to B+ and B- terminals. The voltage is returned in 10ths of a volt. Command syntax:

```
Send: [Address, 24]
Receive: [Value.Byte1, Value.Byte0, Checksum]
```

The command will return 3 bytes. Byte 1 and 2 make up a word variable which is received MSB first and is 10th of a volt. A returned value of 300 would equal 30V. Byte 3 is the checksum. It is calculated the same way as sending a command and can be used to validate the data.

25 - Read Logic Battery Voltage Level

Read a logic battery voltage level connected to LB+ and LB- terminals. The voltage is returned in 10ths of a volt. Command syntax:

```
Send: [Address, 25]
Receive: [Value.Byte1, Value.Byte0, Checksum]
```

The command will return 3 bytes. Byte 1 and 2 make up a word variable which is received MSB first and is 10th of a volt. A returned value of 50 would equal 5V. Byte 3 is the checksum. It is calculated the same way as sending a command and can be used to validate the data.

26 - Set Minimum Logic Voltage Level

Sets logic input (LB- / LB+) minimum voltage level. If the battery voltages drops below the set voltage level RoboClaw will shut down. The value is cleared at start up and must set after each power up. The voltage is set in .2 volt increments. A value of 0 sets the minimum value allowed which is 3V. The valid data range is 0 - 120 (6V - 28V). The formula for calculating the voltage is: (Desired Volts - 6) x 5 = Value. Examples of valid values are 3V = 0, 8V = 10 and 11V = 25.

```
Send: [128, 26, 0, (154 & 0X7F)]
```

27 - Set Maximum Logic Voltage Level

Sets logic input (LB- / LB+) maximum voltage level. The valid data range is 0 - 144 (0V - 28V). By setting the maximum voltage level RoboClaw will go into shut down and requires a hard reset to recovers. The formula for calculating the voltage is: Desired Volts x 5.12 = Value. Examples of valid values are 12V = 62, 16V = 82 and 24V = 123.

```
Send: [128, 27, 82, (213 & 0X7F)]
```

49 - Read Motor Currents

Read the current draw from each motor in 10ma increments. Command syntax:

```
Send: [Address, 49]
Receive: [M1Cur.Byte1, M1Cur.Byte0, M2Cur.Byte1, M2Cur.Byte0, Checksum]
```

The command will return 5 bytes. Bytes 1 and 2 combine to represent the current in 10ma increments of motor1. Bytes 3 and 4 combine to represent the current in 10ma increments of motor2 . Byte 5 is the checksum.

55 - Read Motor 1 Velocity P, I, D Constants

Read the PID and QPPS Settings. Command syntax:

```
Send: [Address, 55]
Receive: [P(4 bytes), I(4 bytes), D(4 bytes), QPPS(4 byte), Checksum]
```

56 - Read Motor 2 Velocity P, I, D Constants

Read the PID and QPPS Settings. Command syntax:

```
Send: [Address, 56]
Receive: [P(4 bytes), I(4 bytes), D(4 bytes), QPPS(4 byte), Checksum]
```

57 - Set Main Battery Voltages

Set the Main Battery Voltages cutoffs, Min and Max. Command syntax:

```
Send: [Address, 57, Min(2 bytes), Max(2bytes, Checksum]
```

58 - Set Logic Battery Voltages

Set the Logic Battery Voltages cutoffs, Min and Max. Command syntax:

```
Send: [Address, 58, Min(2 bytes), Max(2bytes, Checksum]
```

59 - Read Main Battery Voltage Settings

Read the Main Battery Voltage Settings. Command syntax:

```
Send: [Address, 59]
Receive: [Min(2 bytes), Max(2 bytes), Checksum]
```

60 - Read Logic Battery Voltage Settings

Read the Main Battery Voltage Settings. Command syntax:

```
Send: [Address, 60]
Receive: [Min(2 bytes), Max(2 bytes), Checksum]
```

63 - Read Motor 1 Position P, I, D Constants

Read the Position PID Settings. Command syntax:

```
Send: [Address, 63]
Receive: [P(4 bytes), I(4 bytes), D(4 bytes), MaxI(4 byte), Deadzone(4 byte),
          MinPos(4 byte), MaxPos(4 byte), Checksum]
```

64 - Read Motor 2 Position P, I, D Constants

Read the Position PID Settings. Command syntax:

```
Send: [Address, 64]
Receive: [P(4 bytes), I(4 bytes), D(4 bytes), MaxI(4 byte), Deadzone(4 byte),
          MinPos(4 byte), MaxPos(4 byte), Checksum]
```

82 - Read Temperature

Read the board temperature. Value returned is in 0.1 degree increments. Command syntax:

```
Send: [Address, 82]
Receive: [Temperature(2 bytes), Checksum]
```

90 - Read Error Status

Read the current error status. Command syntax:

```
Send: [Address, 90]
Receive: [Error, Checksum]
```

Error Mask

Normal	0x00
M1 OverCurrent	0x01
M2 OverCurrent	0x02
E-Stop	0x04
Temperature	0x08
Main Battery High	0x10
Main Battery Low	0x20
Logic Battery High	0x40
Logic Battery Low	0x80

91 - Read Encoder Mode

Read the encoder mode for both motors. Command syntax:

```
Send: [Address, 91]
Receive: [Mode1, Mode2, Checksum]
```

92 - Set Motor 1 Encoder Mode

Set the Encoder Mode for motor 1. Command syntax:

```
Send: [Address, 92, Mode, Checksum]
```

93 - Set Motor 2 Encoder Mode

Set the Encoder Mode for motor 1. Command syntax:

```
Send: [Address, 93, Mode, Checksum]
```

Encoder Mode bits

Bit 7	Enable RC/Analog Encoder support
Bit 6-1	N/A
Bit 0	Quadrature(0)/Absolute(1)

94 - Write Settings to EEPROM

Writes all settings to non-volatile memory. Command syntax:

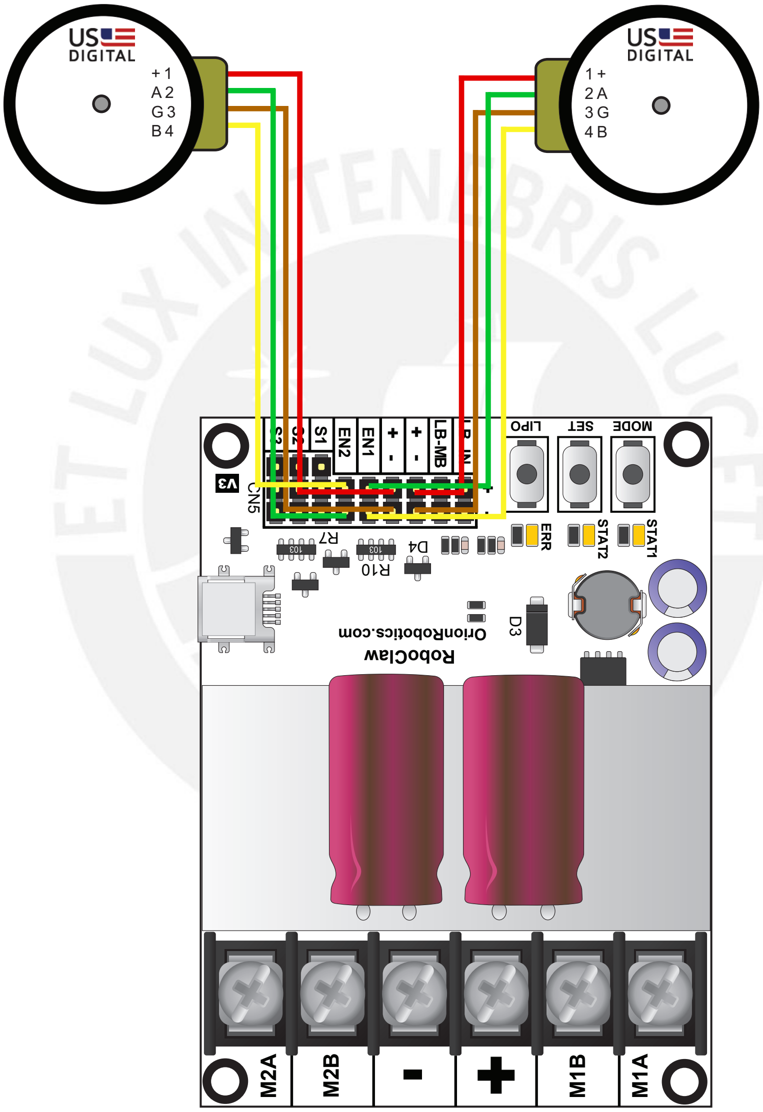
```
Send: [Address, 94]
Receive: [Checksum]
```

Quadrature Decoding

Quadrature Encoder Wiring

RoboClaw is capable of reading two quadrature encoders one for each motor channel. The main RoboClaw header provides two +5VDC connections with dual A and B input signals.

In a two motor robot configuration one motor will spin clock wise (CW) while the other motor will spin counter clock wise (CCW). The A and B inputs for one of the two encoders must be reversed as shown. If either encoder is connected wrong one will count up and the other down this will cause commands like mix drive forward to not work properly.



Quadrature Encoder Commands

The following commands are used in dealing with the quadrature decoding counter registers. The quadrature decoder is a simple counter that counts the incoming pulses, tracks the direction and speed of each pulse. There are two registers one each for M1 and M2. (Note: A microcontroller with a hardware UART is recommended for use with packet serial modes).

Command	Description
16	Read Quadrature Encoder Register for M1.
17	Read Quadrature Encoder Register for M2.
18	Read M1 Speed in Pulses Per Second.
19	Read M2 Speed in Pulses Per Second.
20	Resets Quadrature Encoder Registers for M1 and M2.

16 - Read Quadrature Encoder Register M1

Read decoder M1 counter. Since CMD 16 is a read command it does not require a checksum. However a checksum value will be returned from RoboClaw and can be used to validate the data. Command syntax:

```
Send: [Address, CMD]
Receive: [Value1.Byte3, Value1.Byte2, Value1.Byte1, Value1.Byte0, Value2,
Checksum]
```

The command will return 6 bytes. Byte 1,2,3 and 4 make up a long variable which is received MSB first and represents the current count which can be any value from 0 - 4,294,967,295. Each pulse from the quadrature encoder will increment or decrement the counter depending on the direction of rotation.

Byte 5 is the status byte for M1 decoder. It tracks counter underflow, direction, overflow and if the encoder is operational. The byte value represents:

Bit0 - Counter Underflow (1= Underflow Occurred, Clear After Reading)
Bit1 - Direction (0 = Forward, 1 = Backwards)
Bit2 - Counter Overflow (1= Underflow Occurred, Clear After Reading)
Bit3 - Reserved
Bit4 - Reserved
Bit5 - Reserved
Bit6 - Reserved
Bit7 - Reserved

Byte 6 is the checksum. It is calculated the same way as sending a command, Sum all the values sent and received except the checksum and mask the 8th bit.

17 - Read Quadrature Encoder Register M2

Read decoder M2 counter. Since CMD 16 is a read command it does not require a checksum. However a checksum value will be returned from RoboClaw and can be used to validate the data. Command syntax:

```
Send: [Address, CMD]
Receive: [Value1.Byte3, Value1.Byte2, Value1.Byte1, Value1.Byte0, Value2, Checksum]
```

The command will return 6 bytes. Byte 1,2,3 and 4 make up a long variable which is received MSB first and represents the current count which can be any value from 0 - 4,294,967,295. Each pulse from the quadrature encoder will increment or decrement the counter depending on the direction of rotation.

Byte 5 is the status byte for M1 decoder. It tracks counter underflow, direction, overflow and if the encoder is operational. The byte value represents:

Bit0 - Counter Underflow (1= Underflow Occurred, Clear After Reading)
Bit1 - Direction (0 = Forward, 1 = Backwards)
Bit2 - Counter Overflow (1= Underflow Occurred, Clear After Reading)
Bit3 - Reserved
Bit4 - Reserved
Bit5 - Reserved
Bit6 - Reserved
Bit7 - Reserved

Byte 6 is the checksum.

18 - Read Speed M1

Read M1 counter speed. Returned value is in pulses per second. RoboClaw keeps track of how many pulses received per second for both decoder channels. Since CMD 18 is a read command it does not require a checksum to be sent. However a checksum value will be returned from RoboClaw and can be used to validate the data. Command syntax:

```
Send: [Address, CMD]
Receive: [Value1.Byte3, Value1.Byte2, Value1.Byte1, Value1.Byte0, Value2, Checksum]
```

The command will return 6 bytes. Byte 1,2,3 and 4 make up a long variable which is received MSB first and is the current ticks per second which can be any value from 0 - 4,294,967,295. Byte 5 is the direction (0 – forward, 1 - backward). Byte 6 is the checksum.

19 - Read Speed M2

Read M2 counter speed. Returned value is in pulses per second. RoboClaw keeps track of how many pulses received per second for both decoder channels. Since CMD 19 is a read command it does not require a checksum to be sent. However a checksum value will be returned from RoboClaw and can be used to validate the data. Command syntax:

```
Send: [Address, CMD]
Receive: [Value1.Byte3, Value1.Byte2, Value1.Byte1, Value1.Byte0, Value2, Checksum]
```

The command will return 6 bytes. Byte 1,2,3 and 4 make up a long variable which is received MSB first and is the current ticks per second which can be any value from 0 - 4,294,967,295. Byte 5 is the direction (0 – forward, 1 - backward). Byte 6 is the checksum.

20 - Reset Quadrature Encoder Counters

Will reset both quadrature decoder counters to zero.

```
Send: [128, 20, ((128+20) & 0x7F)]
```

Advanced Motor Control

The following commands are used to control motor speeds, acceleration and distance using the quadrature encoders. All speeds are given in quad pulses per second (QPPS) unless otherwise stated. Quadrature encoders of different types and manufactures can be used. However many have different resolutions and maximum speeds at which they operate. So each quadrature encoder will produce a different range of pulses per second.

Command	Description
28	Set Velocity PID Constants for M1.
29	Set Velocity PID Constants for M2.
30	Read Current M1 Speed Resolution 125th of a Second.
31	Read Current M2 Speed Resolution 125th of a Second.
32	Drive M1 With Signed Duty Cycle. (Encoders not required)
33	Drive M2 With Signed Duty Cycle. (Encoders not required)
34	Mix Mode Drive M1 / M2 With Signed Duty Cycle. (Encoders not required)
35	Drive M1 With Signed Speed.
36	Drive M2 With Signed Speed.
37	Mix Mode Drive M1 / M2 With Signed Speed.
38	Drive M1 With Signed Speed And Acceleration.
39	Drive M2 With Signed Speed And Acceleration.
40	Mix Mode Drive M1 / M2 With Speed And Acceleration.
41	Drive M1 With Signed Speed And Distance. Buffered.
42	Drive M2 With Signed Speed And Distance. Buffered.
43	Mix Mode Drive M1 / M2 With Speed And Distance. Buffered.
44	Drive M1 With Signed Speed, Acceleration and Distance. Buffered.
45	Drive M2 With Signed Speed, Acceleration and Distance. Buffered.
46	Mix Mode Drive M1 / M2 With Speed, Acceleration And Distance. Buffered.
47	Read Buffer Length.
50	Mix Drive M1 / M2 With Individual Speed and Acceleration
51	Mix Drive M1 / M2 With Individual Speed, Accel and Distance
52	Drive M1 With Duty and Accel. (Encoders not required)
53	Drive M2 With Duty and Accel. (Encoders not required)
54	Mix Drive M1 / M2 With Duty and Accel. (Encoders not required)
61	Set Position PID Constants for M1.
62	Set Position PID Constants for M2
65	Drive M1 with signed Speed, Accel, Deccel and Position
66	Drive M2 with signed Speed, Accel, Deccel and Position
67	Drive M1 & M2 with signed Speed, Accel, Deccel and Position

28 - Set PID Constants M1

Several motor and quadrature combinations can be used with RoboClaw. In some cases the default PID values will need to be tuned for the systems being driven. This gives greater flexibility in what motor and encoder combinations can be used. The RoboClaw PID system consist of four constants starting with QPPS, P = Proportional, I= Integral and D= Derivative. The defaults values are:

```
QPPS = 44000
P = 0x00010000
I = 0x00008000
D = 0x00004000
```

QPPS is the speed of the encoder when the motor is at 100% power. P, I, D are the default values used after a reset. Command syntax:

```
Send: [Address, 28, D(4 bytes), P(4 bytes), I(4 bytes), QPPS(4 byte), Checksum]
```

Each value is made up of 4 bytes for a long. To write the registers a checksum value is used. This prevents an accidental write.

29 - Set PID Constants M2

Several motor and quadrature combinations can be used with RoboClaw. In some cases the default PID values will need to be tuned for the systems being driven. This gives greater flexibility in what motor and encoder combinations can be used. The RoboClaw PID system consist of four constants starting with QPPS, P = Proportional, I= Integral and D= Derivative. The defaults values are:

```
QPPS = 44000
P = 0x00010000
I = 0x00008000
D = 0x00004000
```

QPPS is the speed of the encoder when the motor is at 100% power. P, I, D are the default values used after a reset. Command syntax:

```
Send: [Address, 29, D(4 bytes), P(4 bytes), I(4 bytes), QPPS(4 byte), Checksum]
```

Each value is made up of 4 bytes for a long. To write the registers a checksum value is used. This prevents an accidental write.

30 - Read Current Speed M1

Read the current pulse per 125th of a second. This is a high resolution version of command 18 and 19. Command 30 can be used to make a independent PID routine. The resolution of the command is required to create a PID routine using any microcontroller or PC used to drive RoboClaw. The command syntax:

```
Send: [Address, CMD]
Receive: [Value1.Byte3, Value1.Byte2, Value1.Byte1, Value1.Byte0, Value2, Checksum]
```

The command will return 5 bytes, MSB sent first for a long. The first 4 bytes are a 32 byte value (long) that repersent the speed. The 5th byte (Value2) is direction (0 – forward, 1 - backward). is A checksum is returned in order to validate the data returned.

31 - Read Current Speed M2

Read the current pulse per 125th of a second. This is a high resolution version of command 18 and 19. Command 31 can be used to make a independent PID routine. The resolution of the command is required to create a PID routine using any microcontroller or PC used to drive RoboClaw. The command syntax:

```
Send: [Address, CMD]
Receive: [Value1.Byte3, Value1.Byte2, Value1.Byte1, Value1.Byte0, Value2, Checksum]
```

The command will return 5 bytes, MSB sent first for a long. The first 4 bytes are a 32 byte value (long) that repersent the speed. The 5th byte (Value2) is direction (0 – forward, 1 - backward). is A checksum is returned in order to validate the data returned.

32 - Drive M1 With Signed Duty Cycle

Drive M1 using a duty cycle value. The duty cycle is used to control the speed of the motor without a quadrature encoder. The command syntax:

```
Send: [Address, CMD, Duty(2 Bytes), Checksum]
```

The duty value is signed and the range is +-1500.

33 - Drive M2 With Signed Duty Cycle

Drive M2 using a duty cycle value. The duty cycle is used to control the speed of the motor without a quadrature encoder. The command syntax:

```
Send: [Address, CMD, Duty(2 Bytes), Checksum]
```

The duty value is signed and the range is +-1500.

34 - Mix Mode Drive M1 / M2 With Signed Duty Cycle

Drive both M1 and M2 using a duty cycle value. The duty cycle is used to control the speed of the motor without a quadrature encoder. The command syntax:

```
Send: [Address, CMD, DutyM1(2 Bytes), DutyM2(2 Bytes), Checksum]
```

The duty value is signed and the range is +-1500.

35 - Drive M1 With Signed Speed

Drive M1 using a speed value. The sign indicates which direction the motor will turn. This command is used to drive the motor by quad pulses per second. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent the motor will begin to accelerate as fast as possible until the defined rate is reached. The command syntax:

```
Send: [Address, CMD, Qspeed(4 Bytes), Checksum]
```

4 Bytes (long) are used to express the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses.

36 - Drive M2 With Signed Speed

Drive M2 with a speed value. The sign indicates which direction the motor will turn. This command is used to drive the motor by quad pulses per second. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent, the motor will begin to accelerate as fast as possible until the rate defined is reached. The command syntax:

```
Send: [Address, CMD, Qspeed(4 Bytes), Checksum]
```

4 Bytes (long) are used to expressed the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses.

37 - Mix Mode Drive M1 / M2 With Signed Speed

Drive M1 and M2 in the same command using a signed speed value. The sign indicates which direction the motor will turn. This command is used to drive both motors by quad pulses per second. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent the motor will begin to accelerate as fast as possible until the rate defined is reached. The command syntax:

```
Send: [Address, CMD, QspeedM1(4 Bytes), QspeedM2(4 Bytes), Checksum]
```

4 Bytes (long) are used to express the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses.

38 - Drive M1 With Signed Speed And Acceleration

Drive M1 with a signed speed and acceleration value. The sign indicates which direction the motor will run. The acceleration values are not signed. This command is used to drive the motor by quad pulses per second and using an acceleration value for ramping. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent the motor will begin to accelerate incrementally until the rate defined is reached. The command syntax:

```
Send: [Address, CMD, Accel(4 Bytes), Qspeed(4 Bytes), Checksum]
```

4 Bytes (long) are used to express the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses. The acceleration is measured in speed per second. An acceleration value of 12,000 QPPS with a speed of 12,000 QPPS would accelerate a motor from 0 to 12,000 QPPS in 1 second. Another example would be an acceleration value of 24,000 QPPS and a speed value of 12,000 QPPS would accelerate the motor to 12,000 QPPS in 0.5 seconds.

39 - Drive M2 With Signed Speed And Acceleration

Drive M2 with a signed speed and acceleration value. The sign indicates which direction the motor will run. The acceleration value is not signed. This command is used to drive the motor by quad pulses per second and using an acceleration value for ramping. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent the motor will begin to accelerate incrementally until the rate defined is reached. The command syntax:

```
Send: [Address, CMD, Accel(4 Bytes), Qspeed(4 Bytes), Checksum]
```

4 Bytes (long) are used to express the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses. The acceleration is measured in speed per second. An acceleration value of 12,000 QPPS with a speed of 12,000 QPPS would accelerate a motor from 0 to 12,000 QPPS in 1 second. Another example would be an acceleration value of 24,000 QPPS and a speed value of 12,000 QPPS would accelerate the motor to 12,000 QPPS in 0.5 seconds.

40 - Mix Mode Drive M1 / M2 With Signed Speed And Acceleration

Drive M1 and M2 in the same command using one value for acceleration and two signed speed values for each motor. The sign indicates which direction the motor will run. The acceleration value is not signed. The motors are sync during acceleration. This command is used to drive the motor by quad pulses per second and using an acceleration value for ramping. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent the motor will begin to accelerate incrementally until the rate defined is reached. The command syntax:

```
Send: [Address, CMD, Accel(4 Bytes), QspeedM1(4 Bytes), QspeedM2(4 Bytes), Checksum]
```

4 Bytes (long) are used to express the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses. The acceleration is measured in speed per second. An acceleration value of 12,000 QPPS with a speed of 12,000 QPPS would accelerate a motor from 0 to 12,000 QPPS in 1 second. Another example would be an acceleration value of 24,000 QPPS and a speed value of 12,000 QPPS would accelerate the motor to 12,000 QPPS in 0.5 seconds.

41 - Buffered M1 Drive With Signed Speed And Distance

Drive M1 with a signed speed and distance value. The sign indicates which direction the motor will run. The distance value is not signed. This command is buffered. This command is used to control the top speed and total distance traveled by the motor. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
Send: [Address, CMD, QSpeed(4 Bytes), Distance(4 Bytes), Buffer(1 Byte), Checksum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

42 - Buffered M2 Drive With Signed Speed And Distance

Drive M2 with a speed and distance value. The sign indicates which direction the motor will run. The distance value is not signed. This command is buffered. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
Send: [Address, CMD, QSpeed(4 Bytes), Distance(4 Bytes), Buffer(1 Byte), Checksum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

43 - Buffered Mix Mode Drive M1 / M2 With Signed Speed And Distance

Drive M1 and M2 with a speed and distance value. The sign indicates which direction the motor will run. The distance value is not signed. This command is buffered. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
    Send: [Address, CMD, QSpeedM1(4 Bytes), DistanceM1(4 Bytes),
            QSpeedM2(4 Bytes), DistanceM2(4 Bytes), Buffer(1 Byte), Checksum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

44 - Buffered M1 Drive With Signed Speed, Accel And Distance

Drive M1 with a speed, acceleration and distance value. The sign indicates which direction the motor will run. The acceleration and distance values are not signed. This command is used to control the motors top speed, total distanced traveled and at what incremental acceleration value to use until the top speed is reached. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
    Send: [Address, CMD, Accel(4 bytes), QSpeed(4 Bytes), Distance(4 Bytes),
            Buffer(1 Byte), Checksum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

45 - Buffered M2 Drive With Signed Speed, Accel And Distance

Drive M2 with a speed, acceleration and distance value. The sign indicates which direction the motor will run. The acceleration and distance values are not signed. This command is used to control the motors top speed, total distanced traveled and at what incremental acceleration value to use until the top speed is reached. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
    Send: [Address, CMD, Accel(4 bytes), QSpeed(4 Bytes), Distance(4 Bytes),
            Buffer(1 Byte), Checksum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

46 - Buffered Mix Mode Drive M1 / M2 With Signed Speed, Accel And Distance

Drive M1 and M2 with a speed, acceleration and distance value. The sign indicates which direction the motor will run. The acceleration and distance values are not signed. This command is used to control both motors top speed, total distanced traveled and at what incremental acceleration value to use until the top speed is reached. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
Send: [Address, CMD, Accel(4 Bytes), QSpeedM1(4 Bytes), DistanceM1(4 Bytes),
QSpeedM2(4 bytes), DistanceM2(4 Bytes), Buffer(1 Byte), Checksum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

47 - Read Buffer Length

Read both motor M1 and M2 buffer lengths. This command can be used to determine how many commands are waiting to execute.

```
Send: [Address, CMD]
Receive: [BufferM1(1 Bytes), BufferM2(1 Bytes), Checksum]
```

The return values represent how many commands per buffer are waiting to be executed. The maximum buffer size per motor is 31 commands. A return value of 0x80(128) indicates the buffer is empty. A return value of 0 indiciates the last command sent is executing. A value of 0x80 indicates the last command buffered has finished.

50 - Mix Mode Drive M1 / M2 With Signed Speed And Individual Accelerations

Drive M1 and M2 in the same command using one value for acceleration and two signed speed values for each motor. The sign indicates which direction the motor will run. The acceleration value is not signed. The motors are sync during acceleration. This command is used to drive the motor by quad pulses per second and using an acceleration value for ramping. Different quadrature encoders will have different rates at which they generate the incoming pulses. The values used will differ from one encoder to another. Once a value is sent the motor will begin to accelerate incrementally until the rate defined is reached. The command syntax:

```
Send: [Address, CMD, AccelM1(4 Bytes), QspeedM1(4 Bytes), AccelM2(4 Bytes),
QspeedM2(4 Bytes), Checksum]
```

4 Bytes (long) are used to express the pulses per second. Quadrature encoders send 4 pulses per tick. So 1000 ticks would be counted as 4000 pulses. The acceleration is measured in speed per second. An acceleration value of 12,000 QPPS with a speed of 12,000 QPPS would accelerate a motor from 0 to 12,000 QPPS in 1 second. Another example would be an acceleration value of 24,000 QPPS and a speed value of 12,000 QPPS would accelerate the motor to 12,000 QPPS in 0.5 seconds.

51 - Buffered Mix Mode Drive M1 / M2 With Signed Speed, Individual Accel And Distance

Drive M1 and M2 with a speed, acceleration and distance value. The sign indicates which direction the motor will run. The acceleration and distance values are not signed. This command is used to control both motors top speed, total distanced traveled and at what incremental acceleration value to use until the top speed is reached. Each motor channel M1 and M2 have separate buffers. This command will execute immediately if no other command for that channel is executing, otherwise the command will be buffered in the order it was sent. Any buffered or executing command can be stopped when a new command is issued by setting the Buffer argument. All values used are in quad pulses per second. The command syntax:

```
Send: [Address, CMD, AccelM1(4 Bytes), QSpeedM1(4 Bytes), DistanceM1(4 Bytes),
AccelM2(4 Bytes), QSpeedM2(4 bytes), DistanceM2(4 Bytes), Buffer(1 Byte), Check-
sum]
```

4 Bytes(long) are used to express the pulses per second. The Buffer argument can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used the current running command is stopped, any other commands in the buffer are deleted and the new command is executed.

52 - Drive M1 With Signed Duty And Acceleration

Drive M1 with a signed duty and acceleration value. The sign indicates which direction the motor will run. The acceleration values are not signed. This command is used to drive the motor by PWM and using an acceleration value for ramping. Accel is the rate per second at which the duty changes from the current duty to the specified duty. The command syntax:

```
Send: [Address, CMD, Duty(2 bytes), Accel(2 Bytes), Checksum]
```

The duty value is signed and the range is +-1500.
The accel value range is 0 to 65535

53 - Drive M2 With Signed Duty And Acceleration

Drive M1 with a signed duty and acceleration value. The sign indicates which direction the motor will run. The acceleration values are not signed. This command is used to drive the motor by PWM and using an acceleration value for ramping. Accel is the rate at which the duty changes from the current duty to the specified dury. The command syntax:

```
Send: [Address, CMD, Duty(2 bytes), Accel(2 Bytes), Checksum]
```

The duty value is signed and the range is +-1500.
The accel value range is 0 to 65535

54 - Mix Mode Drive M1 / M2 With Signed Duty And Acceleration

Drive M1 and M2 in the same command using acceleration and duty values for each motor. The sign indicates which direction the motor will run. The acceleration value is not signed. This command is used to drive the motor by PWM using an acceleration value for ramping. The command syntax:

```
Send: [Address, CMD, DutyM1(2 bytes), Accelm1(4 Bytes), DutyM2(2 bytes), AccelM1(4
bytes), Checksum]
```

The duty value is signed and the range is +-1500.
The accel value range is 0 to 65535

61 - Set Motor 1 Position PID Constants

The RoboClaw Position PID system consist of seven constants starting with P = Proportional, I= Integral and D= Derivative, MaxI = Maximum Integral windup, Deadzone in encoder counts, MinPos = Minimum Position and MaxPos = Maximum Position. The defaults values are all zero.

```
Send: [Address, CMD, P(4 bytes), I(4 bytes), D(4 bytes), MaxI(4 bytes), Deadzone(4
bytes), MinPos(4 bytes), MaxPos(4 bytes)
```

Position constants are used only with the Position commands, 65,66 and 67 and RC or Analog mode when in absolute mode with encoders or potentiometers.

62 - Set Motor 2 Position PID Constants

The RoboClaw Position PID system consist of seven constants starting with P = Proportional, I= Integral and D= Derivative, MaxI = Maximum Integral windup, Deadzone in encoder counts, MinPos = Minimum Position and MaxPos = Maximum Position. The defaults values are all zero.

```
Send: [Address, CMD, P(4 bytes), I(4 bytes), D(4 bytes), MaxI(4 bytes), Deadzone(4
bytes), MinPos(4 bytes), MaxPos(4 bytes)
```

Position constants are used only with the Position commands, 65,66 and 67 and RC or Analog mode when in absolute mode with encoders or potentiometers.

65 - Drive M1 with signed Speed, Accel, Deccel and Position

Move M1 position from the current position to the specified new position and hold the new position. Accel sets the acceleration value and deccel the decceleration value. QSpeed sets the speed in quadrature pulses the motor will run at after acceleration and before decceleration. The command syntax:

```
Send: [Address, CMD, Accel(4 bytes), QSpeed(4 Bytes), Deccel(4 bytes), Position(4
Bytes), Buffer(1 Byte), Checksum]
```

66 - Drive M2 with signed Speed, Accel, Deccel and Position

Move M2 position from the current position to the specified new position and hold the new position. Accel sets the acceleration value and deccel the decceleration value. QSpeed sets the speed in quadrature pulses the motor will run at after acceleration and before decceleration. The command syntax:

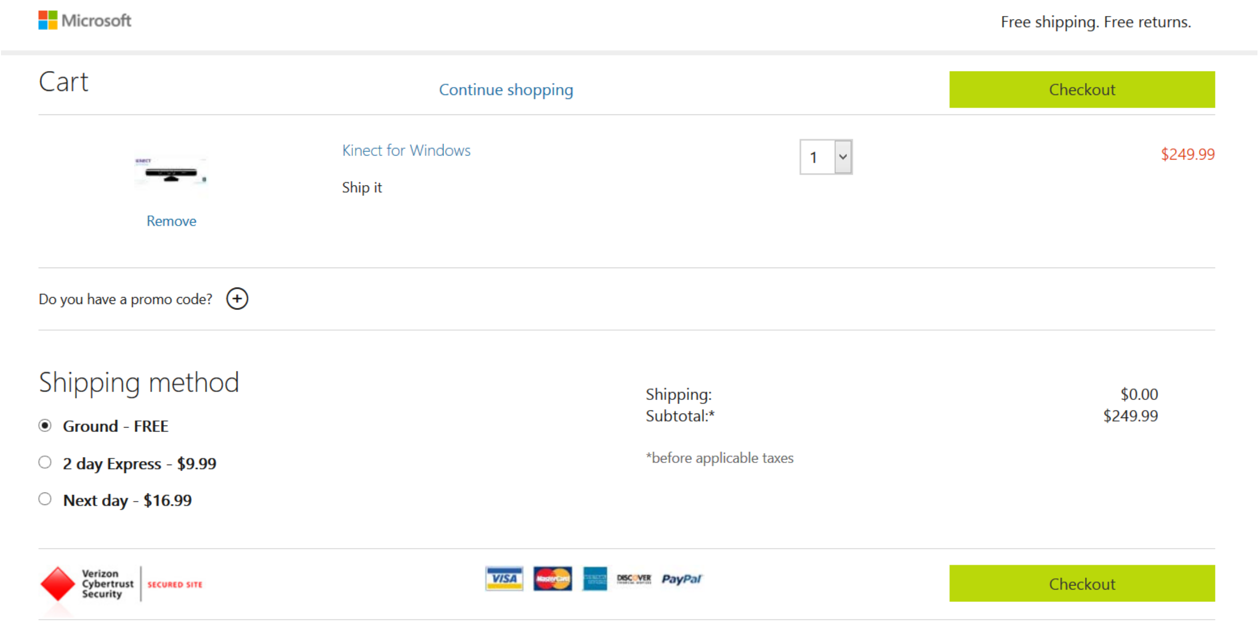
```
Send: [Address, CMD, Accel(4 bytes), QSpeed(4 Bytes), Deccel(4 bytes), Position(4
Bytes), Buffer(1 Byte), Checksum]
```

67 - Drive M1 & M2 with signed Speed, Accel, Deccel and Position

Move M1 & M2 positions from their current positions to the specified new positions and hold the new positions. Accel sets the acceleration value and deccel the decceleration value. QSpeed sets the speed in quadrature pulses the motor will run at after acceleration and before decceleration. The command syntax:

```
Send: [Address, CMD, Accel(4 bytes), QSpeed(4 Bytes), Deccel(4 bytes), Position(4
Bytes),Buffer(1 Byte), Checksum]
```

Cámara Kinect



Free shipping. Free returns.

Cart

Kinect for Windows

Ship it

1

$249.99

Do you have a promo code?

Shipping method

● Ground - FREE

○ 2 day Express - $9.99

○ Next day - $16.99

Shipping:	$0.00
Subtotal:*	$249.99

*before applicable taxes

Amortiguadores

HPI 87595 Sport Shock Set (42-55mm/10mm) (2)

by HPI Racing

Be the first to review this item

List Price: ~~$21.00~~

Price: $12.99 + $5.95 shipping

You Save: $8.01 (38%)

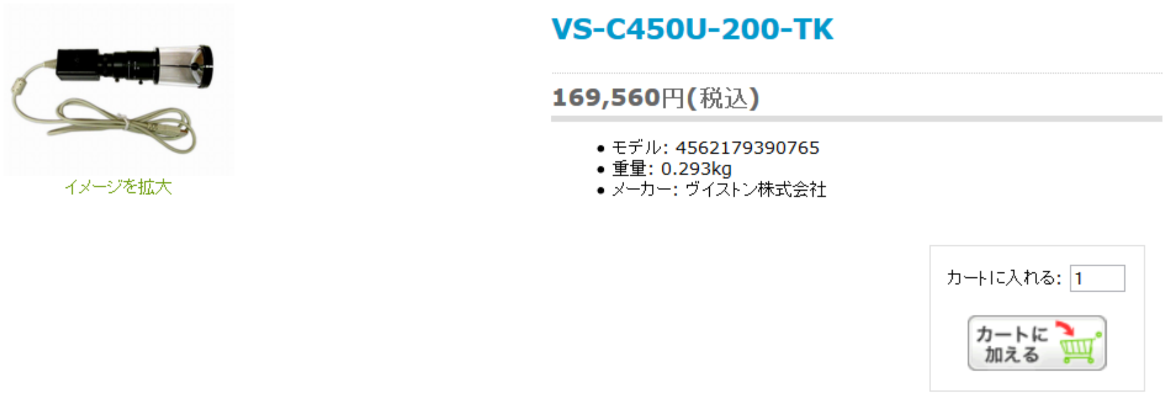
Only 1 left in stock.

Ships from and sold by NitroKillers.

- HPI

8 new from $12.99

Cámara Omnidireccional (Precio en Yenes)



イメージを拡大

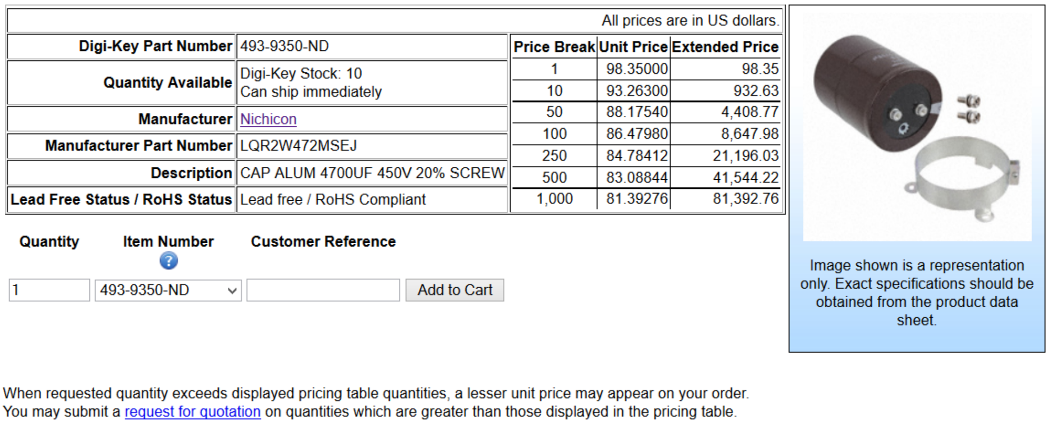
VS-C450U-200-TK

169,560円(税込)

- モデル: 4562179390765
- 重量: 0.293kg
- メーカー: ヴィストン株式会社

カートに入れる: 1

Condensador de Aluminio



All prices are in US dollars.

Digi-Key Part Number	493-9350-ND
Quantity Available	Digi-Key Stock: 10 Can ship immediately
Manufacturer	Nichicon
Manufacturer Part Number	LQR2W472MSEJ
Description	CAP ALUM 4700UF 450V 20% SCREW
Lead Free Status / RoHS Status	Lead free / RoHS Compliant

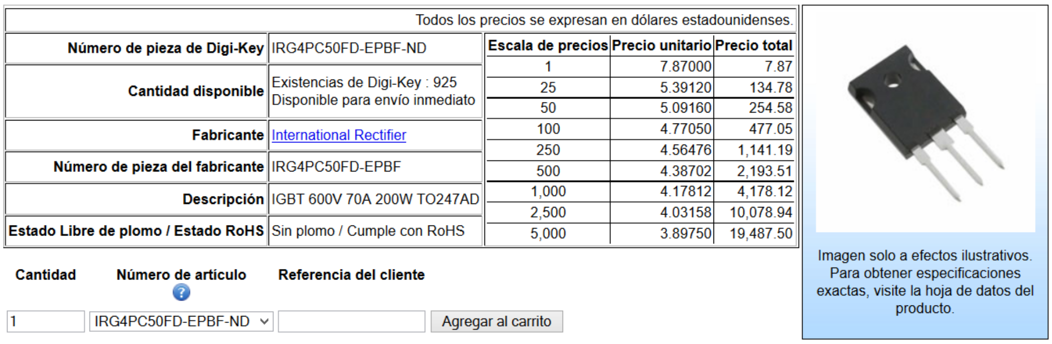
Price Break	Unit Price	Extended Price
1	98.35000	98.35
10	93.26300	932.63
50	88.17540	4,408.77
100	86.47980	8,647.98
250	84.78412	21,196.03
500	83.08844	41,544.22
1,000	81.39276	81,392.76

Quantity	Item Number	Customer Reference	
1	493-9350-ND		Add to Cart

Image shown is a representation only. Exact specifications should be obtained from the product data sheet.

When requested quantity exceeds displayed pricing table quantities, a lesser unit price may appear on your order. You may submit a request for quotation on quantities which are greater than those displayed in the pricing table.

IGBT



Todos los precios se expresan en dólares estadounidenses.

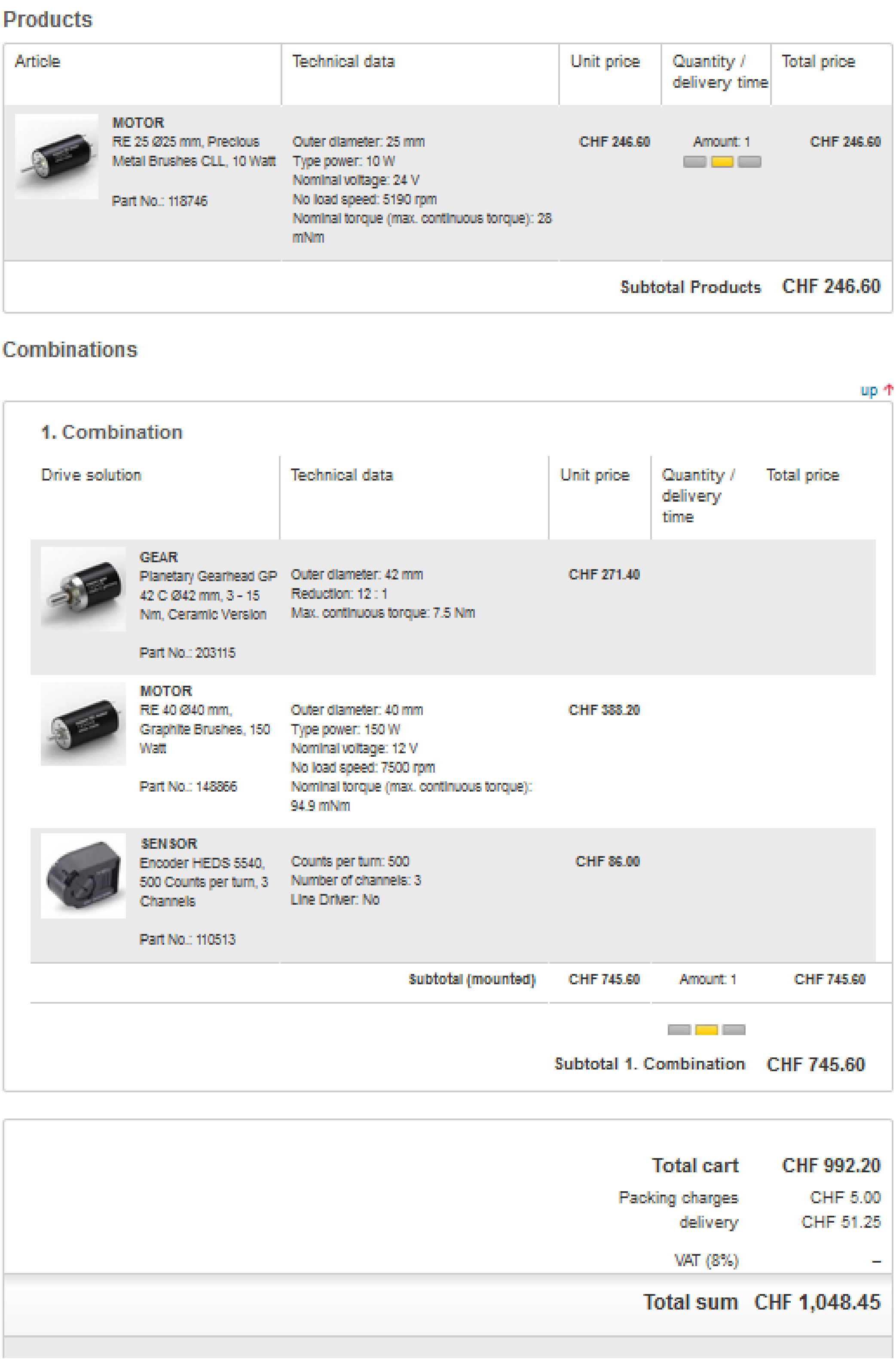
Número de pieza de Digi-Key	IRG4PC50FD-EPBF-ND
Cantidad disponible	Existencias de Digi-Key : 925 Disponible para envío inmediato
Fabricante	International Rectifier
Número de pieza del fabricante	IRG4PC50FD-EPBF
Descripción	IGBT 600V 70A 200W TO247AD
Estado Libre de plomo / Estado RoHS	Sin plomo / Cumple con RoHS

Escala de precios	Precio unitario	Precio total
1	7.87000	7.87
25	5.39120	134.78
50	5.09160	254.58
100	4.77050	477.05
250	4.56476	1,141.19
500	4.38702	2,193.51
1,000	4.17812	4,178.12
2,500	4.03158	10,078.94
5,000	3.89750	19,487.50

Cantidad	Número de artículo	Referencia del cliente	
1	IRG4PC50FD-EPBF-ND		Agregar al carrito

Imagen solo a efectos ilustrativos. Para obtener especificaciones exactas, visite la hoja de datos del producto.

Motores Maxon



Products

Article	Technical data	Unit price	Quantity / delivery time	Total price
MOTOR RE 25 Ø25 mm, Precious Metal Brushes CLL, 10 Watt Part No.: 118746	Outer diameter: 25 mm Type power: 10 W Nominal voltage: 24 V No load speed: 5190 rpm Nominal torque (max. continuous torque): 28 mNm	CHF 246.60	Amount: 1 ▭ 🟨 ▭	CHF 246.60
			Subtotal Products	**CHF 246.60**

Combinations

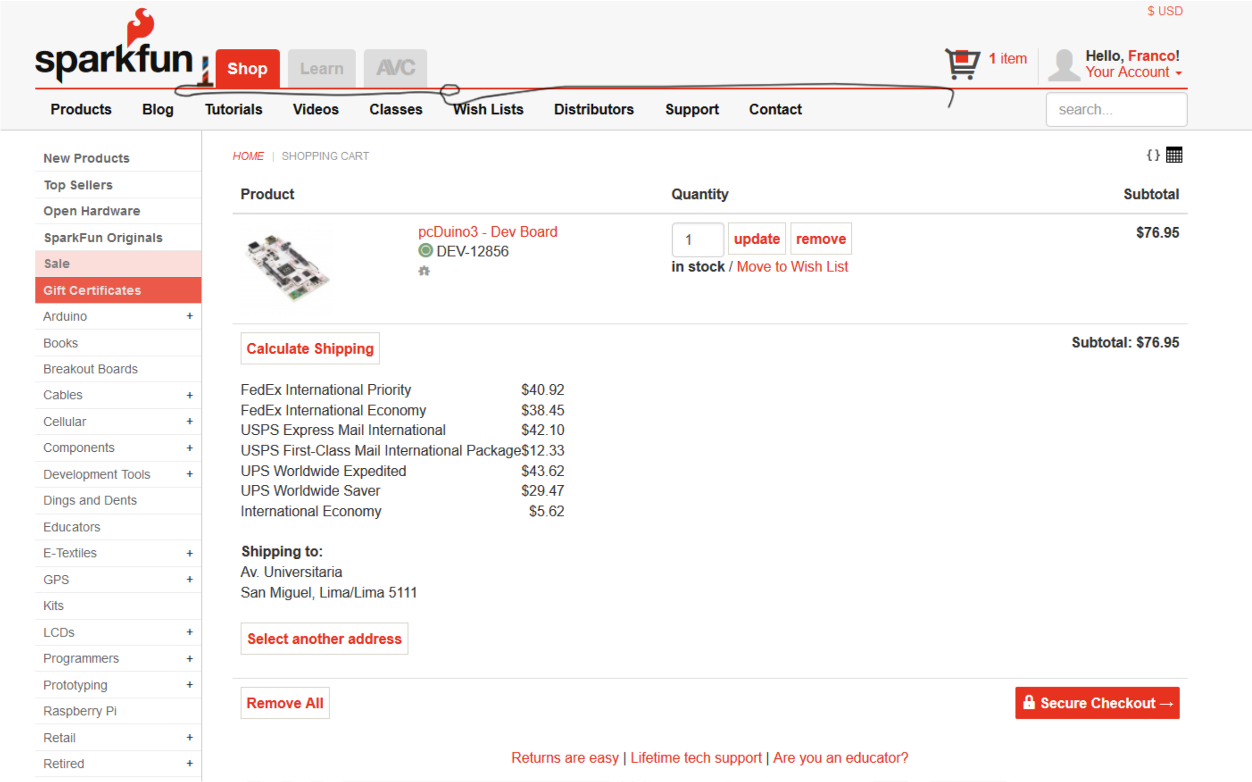
up ↑

1. Combination

Drive solution	Technical data	Unit price	Quantity / delivery time	Total price
GEAR Planetary Gearhead GP 42 C Ø42 mm, 3 - 15 Nm, Ceramic Version Part No.: 203115	Outer diameter: 42 mm Reduction: 12 : 1 Max. continuous torque: 7.5 Nm	CHF 271.40		
MOTOR RE 40 Ø40 mm, Graphite Brushes, 150 Watt Part No.: 148866	Outer diameter: 40 mm Type power: 150 W Nominal voltage: 12 V No load speed: 7500 rpm Nominal torque (max. continuous torque): 94.9 mNm	CHF 388.20		
SENSOR Encoder HEDS 5540, 500 Counts per turn, 3 Channels Part No.: 110513	Counts per turn: 500 Number of channels: 3 Line Driver: No	CHF 86.00		
	Subtotal (mounted)	CHF 745.60	Amount: 1	CHF 745.60
			▭ 🟨 ▭ **Subtotal 1. Combination**	**CHF 745.60**

Total cart	**CHF 992.20**
Packing charges	CHF 5.00
delivery	CHF 51.25
VAT (8%)	–
Total sum	**CHF 1,048.45**

PCduino3



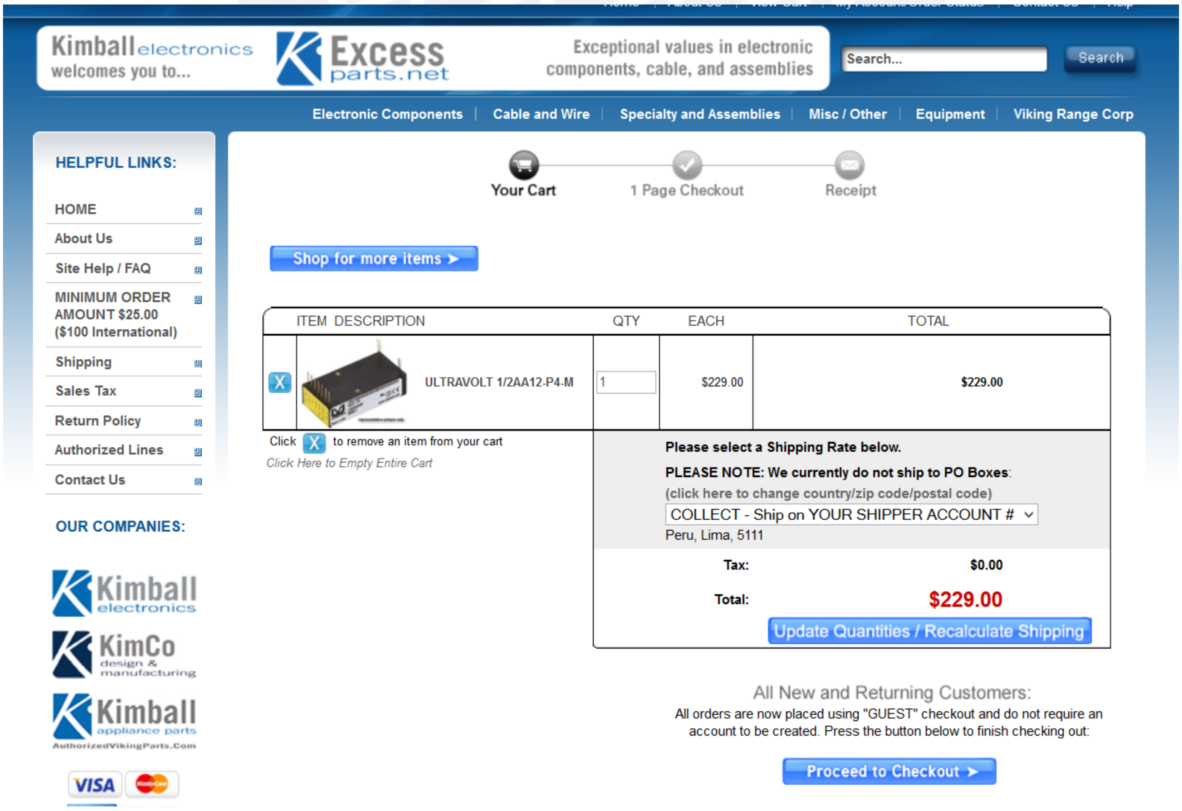
Product	Quantity	Subtotal
pcDuino3 - Dev Board DEV-12856	1 in stock / Move to Wish List	$76.95

Subtotal: $76.95

FedEx International Priority	$40.92
FedEx International Economy	$38.45
USPS Express Mail International	$42.10
USPS First-Class Mail International Package	$12.33
UPS Worldwide Expedited	$43.62
UPS Worldwide Saver	$29.47
International Economy	$5.62

Shipping to:
Av. Universitaria
San Miguel, Lima/Lima 5111

DC-DC



ITEM DESCRIPTION	QTY	EACH	TOTAL
ULTRAVOLT 1/2AA12-P4-M	1	$229.00	$229.00

Click X to remove an item from your cart

Click Here to Empty Entire Cart

Please select a Shipping Rate below.

PLEASE NOTE: We currently do not ship to PO Boxes:

(click here to change country/zip code/postal code)

COLLECT - Ship on YOUR SHIPPER ACCOUNT #

Peru, Lima, 5111

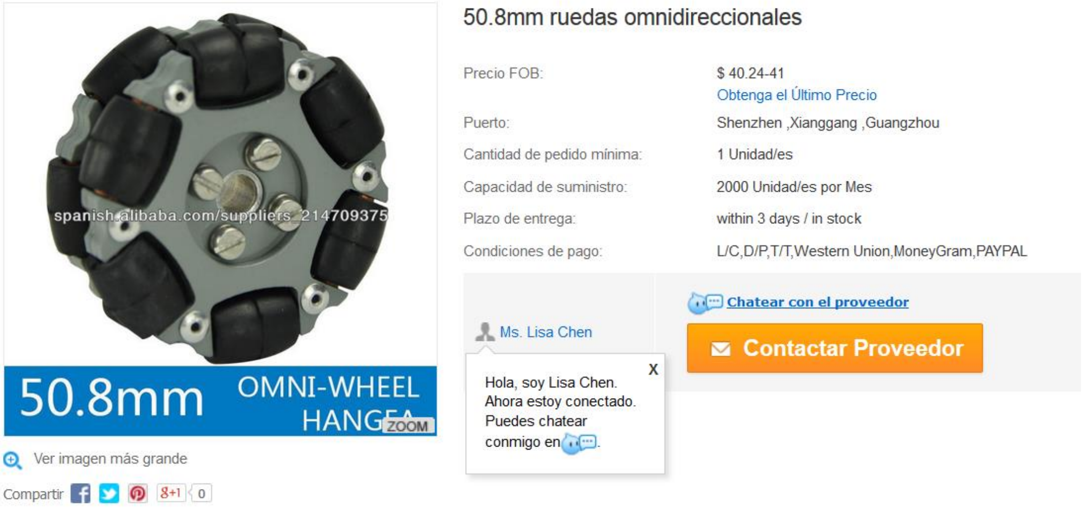
Tax: $0.00

Total: $229.00

All New and Returning Customers:

All orders are now placed using "GUEST" checkout and do not require an account to be created. Press the button below to finish checking out:

Rueda Omnidireccional



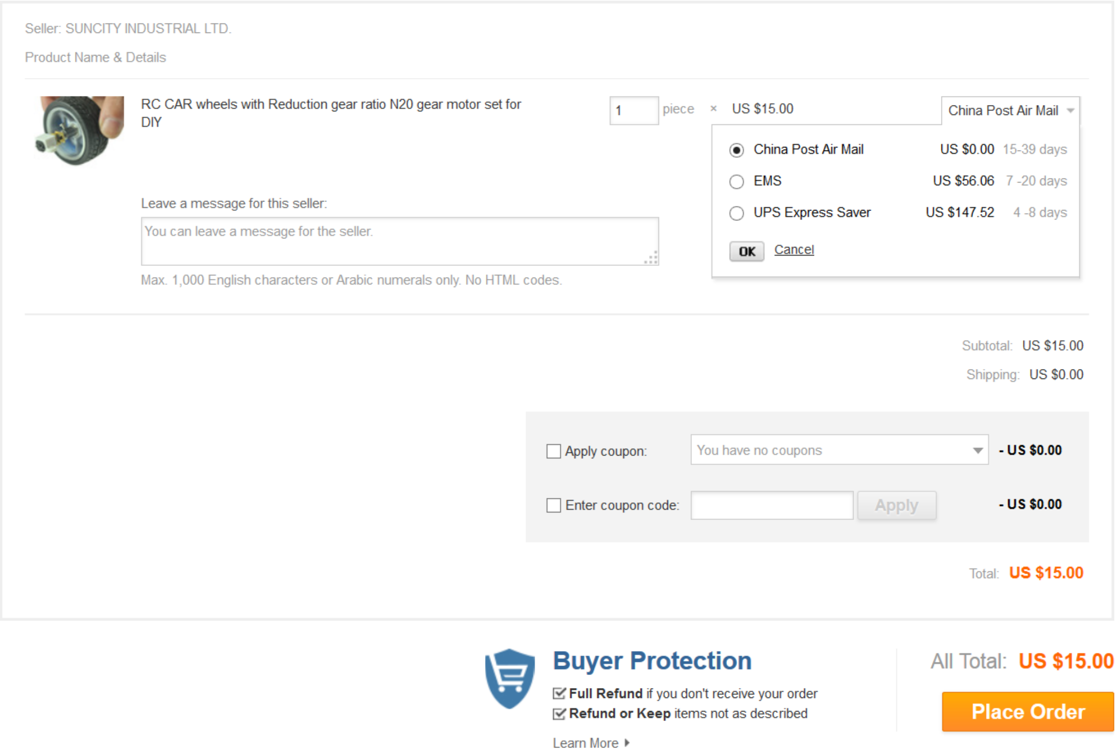
50.8mm ruedas omnidireccionales

Precio FOB:	$ 40.24-41 Obtenga el Último Precio
Puerto:	Shenzhen ,Xianggang ,Guangzhou
Cantidad de pedido mínima:	1 Unidad/es
Capacidad de suministro:	2000 Unidad/es por Mes
Plazo de entrega:	within 3 days / in stock
Condiciones de pago:	L/C,D/P,T/T,Western Union,MoneyGram,PAYPAL

Ms. Lisa Chen

Hola, soy Lisa Chen. Ahora estoy conectado. Puedes chatear conmigo en .

Rueda de sujeción de pelota



Seller: SUNCITY INDUSTRIAL LTD.

Product Name & Details

RC CAR wheels with Reduction gear ratio N20 gear motor set for DIY

1 piece × US $15.00 China Post Air Mail

● China Post Air Mail	US $0.00	15-39 days
○ EMS	US $56.06	7 -20 days
○ UPS Express Saver	US $147.52	4 -8 days

Leave a message for this seller:

You can leave a message for the seller.

Max. 1,000 English characters or Arabic numerals only. No HTML codes.

Subtotal: US $15.00

Shipping: US $0.00

☐ Apply coupon: You have no coupons - US $0.00

☐ Enter coupon code: - US $0.00

Total: US $15.00

☑ **Full Refund** if you don't receive your order

☑ **Refund or Keep** items not as described

All Total: US $15.00

Cotización N°2014-PUCP-V-073

STEIN TRICES
INGENIERÍA, CNC, AUTOMATIZACIÓN Y HERRAMIENTAS

Lima,04 de Julio de 2014

Atención: Ingeniería Mecatrónica
Franco Guillen Basantez

Pontificia Universidad Católica del Perú.

Dirección: Av. Universitaria 1801-San Miguel-Lima
Referencia Manufactura

De nuestra mayor consideración :
En atención a su amable solicitud de cotización, tenemos el agrado de presentarles nuestra oferta económica

Pos	Material	Descripción	Cant.	Precio unitario	Precio total
1	Varios	Mecanizado CNC, Proyecto: **Plataforma Omnidireccional** **Comprende la fabricación de los planos:** Actuador Base condensador Base frontal derecha Base frontal izquierda Base motor Base posterior derecha Base posterior izquierda Base rueda Base superior Cabezal de disparo Cuadrante Cubierta Eje 6mm Eje actuador Eje handling Eje mayor handling Eje minirueda Eje omniwheel Guia de retorno Lateral superior Laterales amortiguador Medio rueda Motor superior Omniwheel shaft Pasador-roller Protección eje solenoide Protección superior camara Ruedilla Soporte amortiguadores Soporte derecho Soporte izquierdo Soporte kinect1 Soporte kienct2 Soporte omniwheel1 Soporte omniwheel2 Soporte omniwheel	1	6651.50	6651.50

		Soporte superior motor Sujeción de rueda Sujetador base Sujetador protector solenoide Sujetador solenoide Sujetador condensador Tapa cuadrante Tapa rueda Tapa solenoide Tubo solenoide Unión 30 Union superior Union 20 Union 40 Panel de switch Guia de sujecion			

Moneda:	Nuevos Soles		
Anotaciones:	-Cualquier modificación adicional, luego de aceptado el diseño, generará cargos adicionales que serán comunicados debidamente al cliente. -Según diseño propuesto. -Fabricación de las cantidades de piezas que señalen los planos. -Posibles variaciones en los vértices, sujetos a evaluación del cliente.		
Marca:	--------------------------	**Valor Venta:**	**6651.50**
Forma de Pago:	Depósito bancario del 50% con la orden de compra, 50% restante Contra-entrega.	**Otros Cargos:**	**0.00**
Forma de entrega:	En sus instalaciones 25 días hábiles después de remitida la O/C y el pago.	**IGV 18%:**	**1197.27**
Validez de la oferta hasta:	18-08-2014	**Importe Total:**	**7848.77**

Agradeciendo a la atención que brinde a la presente, quedamos a su disposición para cualquier consulta que considere necesaria.

Atentamente
Harold R. La Chira Marquez
Chief Commercial Officer
STEINTRICES E.I.R.L.- 20546506470
harold.lachira@steintrices.com
Telephone: 511-4747393
Mobile Phone: 511-992627992
Calle Andrés Costello 190-San Luis //Av. Nicolás Arriola 1061-La Victoria-Lima-Perú
http://www.steintrices.com