Anexo 1: Estado del Arte

Los robots en un futuro cercano aumentarán en número, pues las personas contarán con un humanoide que los ayude en los quehaceres de su hogar, con los cuidados de las personas mayores, discapacitadas o niños. Los robots se acercarán a las personas y por eso se dan estudios basados en el HRI, Interacción Humano Robot, por sus siglas en inglés, de la reacción de las personas cuando interactúan con un robot.

Para las características del sistema mecatrónico diseñado se tomó como referencia a los robots mencionados a continuación. El primer humanoide seleccionado es Wakamaru [2] (figura 1-1) desarrollado en el 2005 por Mitsubishi para ayudar en los hogares, este se comunicaba con las personas y fue diseñado especialmente para atender a personas de la tercera edad y discapacitados.

El segundo robot seleccionado se desarrolló en el 2007 por Toyota nombrada Robina [3] (figura 1-2) una robot guía capaz de dirigir y realizar un tour en Toyota Kaikan Exhibition Hall en Toyota City, Aichi Prefecture, Japan, tiene movimiento autónomo, evade obstáculos, cuenta también con dedos articulados que le permiten ser capaz de firmar autógrafos, gestos y comunicación no verbal. Toyota desde el 2005 viene desarrollando esta clase de robots en 4 diferentes campos para ayudar al ser humano: asistencia de quehaceres domésticos, asistencia de enfermería y cuidados médicos, asistencia en manufactura y asistencia de transporte de personas de corta distancia.

En tercera instancia, diseñados en la Universidad de Tsukuba por Hiroshi Kasai, se encuentran Gemini (figura 1-3) que son un par de robots gemelos conceptuales, cuya función es el de conversar sobre obras de arte en una galería o en una exhibición de museo, esto para llamar la atención a las personas y así sientan curiosidad de lo que hablan los dos robots y de esta manera ayudarles a que aprendan, a manera de intriga, sin que perciban el método de enseñanza.

En cuarto lugar se encuentra LUNA [4] (figura 1-4) desarrollado por RoboDynamics, California, diseñado para ser un robot doméstico de ayuda en el hogar. La empresa tiene como proyección producirlos en masa en el 2021.

Como quinto robot se seleccionó a FURO-S [5] (figura 1-5) desarrollado en Korea por la empresa FutureRobots, es un robot de ayuda, guía y también de pagos. Cuenta con una entrada de tarjeta e impresión de bouchers, tiene dos pantallas una interactiva, touch para que las personas ubiquen lo que deseen encontrar y la otra es su rostro la cual presenta un avatar mujer con expresiones según sea la ocasión.

Por último está Robovie-R3 [6] (figura 1-6) es un humanoide desarrollado por ATR el cual es un asistente para centros comerciales, se encarga de cargar las compras, brindar información de productos, guía por si la personas se pierden, entre otros. El conjunto de humanoides hoy en día que se están generando son diversos y lo que buscan en el fondo es una mejor interacción con las personas, mientras los ayudan con quehaceres rutinarios.

CUADRO COMPARATIVO:

	ALTURA/PESO	GESTOS	ARTICULACIONES	VOZ	VISIÓN	OBSERVACIONES
WAKAMARU	1m/30Kg		brazos, cuello	si	10 caras	llama en caso de emergencia recuerda horario de medicinas
ROBINA	1,2m/60Kg		brazos, cuello,dedos	si	nametags	firma autógrafos robot guía
GEMINI			brazos, cuello	entre ellos		robots conceptuales hablan entre ellos, enseñando en base a la intriga
LUNA	1,52m	si	brazos	si	si	uso doméstico
FURO-S	1,5m	si	brazos, cuello, cuerpo	si	si-kinect	brinda información genera tickets
ROBOVIE-R3	1,08m/35Kg		brazos, cuello	si	si	ayudar personas mayores y discapacitados
Iomi	1,5m	si	brazos, cuello	si	si	guía de biblioteca atrae al público

Tabla 1-1: Cuadro comparativo de características

En la Tabla 1-1 se muestra un cuadro comparativo de los robots escogidos para tener de base en el diseño del sistema mecatrónico.

Figura 1-1 Wakamaru (Fuente: www.mitsubishi.com)   Figura 1-2 Robina (Fuente: www.toyota.com)

Figura 1-3 Gemini (Fuente: www.realrobot.com)   Figura 1-4 Luna (Fuente: www.robots.nu)

Figura 1-5 Furo-S(Fuente: www.fr.co.kr)   Figura 1-6 RobovieR3 (Fuente: www.cnet.com)

Anexo 2: Diseño Exterior

Figura 2-1 : Vista isométrica (Fuente: Propia)   Figura 2-2 : Vista frontal (Fuente: Propia)

En la figura 2-1 se muestra al sistema mecatrónico en una vista isométrica, donde se puede observar la forma del brazo, curvo y con concavidad hacia el exterior.

En la figura 2-2 se observa la vista frontal del robot en el cual también se puede observar la forma curva del brazo. Entonces se puede extraer la conjetura de que cuando el hombro rote el brazo va a señalar hacia afuera.

La forma curva del brazo da la impresión de la existencia de codo, pues no sólo está curvado, sino que la concavidad está hacia el exterior, como fue mencionado, lo cual cuando el servomotor de giro se mueve el brazo da la presencia de un codo sin necesidad de utilizar un servomotor más, lo cual ayuda en el proceso de diseño pues lo simplifica.

Anexo 3: Pioneer P3-DX

adept

mobilerobots®

Pioneer 3-DX

Pioneer 3-DX is a small lightweight two-wheel two-motor differential drive robot ideal for indoor laboratory or classroom use. The robot comes complete with front SONAR, one battery, wheel encoders, a microcontroller with ARCOS firmware, and the Pioneer SDK advanced mobile robotics software development package.

Pioneer research robots are the world's most popular intelligent mobile robots for education and research. Their versatility, reliability and durability have made them the preferred platform for advanced intelligent robotics. Pioneers are pre-assembled, customizable, upgradeable, and rugged enough to last through years of laboratory and classroom use.

Product Features and Benefits

- **Easy to Use** - Comes assembled and integrated with its accessory packages.
- **Reliable** - Construction is durable and rugged. Easily handles the small gaps, minor bumping, jarring, or other obstacles that hinder other robotic platforms. Some Pioneer robots have been in service for over 15 years.
- **Pioneer Software Development Kit** - All Adept MobileRobots platforms include Pioneer SDK, a complete set of robotics applications and libraries that accelerate the development of robotics projects. Pioneer SDK is backed by our product support team.
- **Customizable** - Easily accessorize by choosing from dozens of supported and tested accessories that integrate with the robotic platform. Additional help is available for future upgrades or added accessories.
- **Reference Platform** - Pioneer robots are a standard in intelligent mobile platforms. Search your preferred robotics journal or conference listings to find many examples of Pioneer platforms in research applications.
- **Technical Support** - Pioneer software and hardware comes fully documented with additional help available through our product support team.

Specifications

Construction
- Body: 1.6 mm aluminum (powder-coated)
- Tires: Foam-filled rubber

Operation
- Robot Weight: 9 kg
- Operating Payload: 17 kg

Differential Drive Movement
- Turn Radius: 0 cm
- Swing Radius: 26.7 cm
- Max. Forward/Backward Speed: 1.2 m/s
- Rotation Speed: 300°/s
- Max. Traversable Step: 2.5 cm
- Max. Traversable Gap: 5 cm
- Max. Traversable Grade: 25%
- Traversable Terrain: Indoor, wheelchair accessible

Power
- Run Time: 8-10 hours w/3 batteries (with no accessories)
- Charge Time: 12 hours (standard) or 2.4 hrs (optional high-capacity charger)
- Available Power Supplies:
  - 5 V @ 1.5 A switched
  - 12 V @ 2.5 A switched

Batteries
- Supports up to 3 at a time
- Voltage: 12 V
- Capacity: 7.2 Ah (each)
- Chemistry: lead acid
- Hot-swappable Batteries: Yes

Available Recharge Options:
- Direct plug-in
- Docking station
- Powercube (3-battery charging bay)

* Batteries are accessible through hinged latched access panel for hot-swapping (continuous operation)

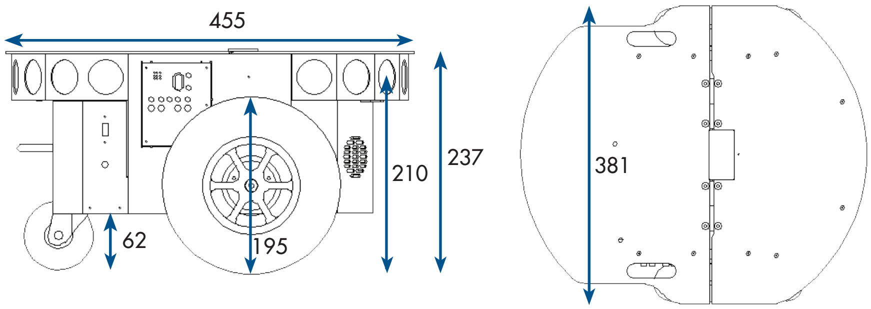
Microcontroller I/O
- System Serial
- 32 digital inputs
- 8 digital outputs
- 7 analog inputs
- 3 serial expansion ports

*Some ports may not be available if certain accessories are included with the robot

User Control Panel
- MIDI programmable piezo buzzer
- Main power indicator
- Battery charge indicator
- 2 AUX power switches
- System reset
- Motor enable pushbutton

Pioneer 3-DX

Dimensions (mm)



Core Software - included with all research platforms

ARIA provides a framework for controlling and receiving data from all MobileRobots platforms, as well as most accessories. Includes open source infrastructures and utilities useful for writing robot control software, support for network sockets, and an extensible framework for client-server network programming.

MobileSim open-source simulator which includes all MobileRobots platforms and many accessories.

MobileEyes graphical user interface client for remote operation and monitoring of the robot.

Mapper 3-Basic tool for creating and editing map files for use with ARIA, MobileSim, and navigation software.

SONARNL provides sonar-based approximate localization and navigation.

Accessory Support Software - bundled with purchase of robotic accessory

ARNL enables robust, laser-based autonomous localization and navigation.

Robotic Arm Support Pioneer arms are packaged with integrated software support.

Speech Recognition and Synthesis Library: Easy-to-use C++ development library for speech recognition based on the open source Sphinx2 system. Speech synthesis (text-to-speech) based on Cepstral synthesizer.

ACTS Color Tracking System: Software application which reads images from a camera and tracks the positions and sizes of multiple color regions. Information can be incorporated into your own software via ARIA.

Optional Industrial Grade Internally Mounted Computers

Mamba EBX-37 (Dual Core 2.26 GHz - 2-8 GB RAM)
6 X USB2.0 Ports
2 X PC/104+ Slots
4 X RS-232 Serial Ports
2 X 10/100/1000 Ethernet Ports
Onboard Audio & Video
Solid State Drive
Optional Wireless Ethernet

Optional Accessories:

- Laser-range finders
- Mono- and stereo-vision cameras
- Rear SONAR
- Wireless serial to Ethernet for remote operation
- Robotic arms and grippers
- Gyroscope
- Segmented bumper arrays
- Speakers and microphones
- Joystick
- Many more...

Include our integrated & supported accessories with your Pioneer 3-DX.

Here are some popular configurations to choose from:

Mapping & Vision

Gripping & Manipulation

Audio & Speech

More Information:

See our website www.mobilerobots.com for a full range of supported accessories or contact our sales department to discuss your application.

adept

Adept Technology, Inc. 10 Columbia Drive, Amherst, NH 03031
Tel: 603-881-7960 Email: sales@mobilerobots.com
www.mobilerobots.com

Specifications subject to change without notice.

Anexo 4: Sensor Ultrasónico MB1000 LV MaxSonar EZ0

LV-MaxSonar®-EZ™ Series

High Performance Sonar Range Finder

MB1000, MB1010, MB1020, MB1030, MB1040

With 2.5V - 5.5V power the LV-MaxSonar-EZ provides very short to long-range detection and ranging in a very small package. The LV-MaxSonar-EZ detects objects from 0-inches to 254-inches (6.45-meters) and provides sonar range information from 6-inches out to 254-inches with 1-inch resolution. Objects from 0-inches to 6-inches typically range as 6-inches[1]. The interface output formats included are pulse width output, analog voltage output, and RS232 serial output. Factory calibration and testing is completed with a flat object. ***[1]See Close Range Operation***

Features

- Continuously variable gain for control and side lobe suppression
- Object detection to zero range objects
- 2.5V to 5.5V supply with 2mA typical current draw
- Readings can occur up to every 50mS, (20-Hz rate)
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- Interfaces are active simultaneously
- Serial, 0 to Vcc, 9600 Baud, 81N
- Analog, (Vcc/512) / inch
- Pulse width, (147uS/inch)
- Learns ringdown pattern when commanded to start ranging
- Designed for protected indoor environments
- Sensor operates at 42KHz
- High output square wave sensor drive (double Vcc)

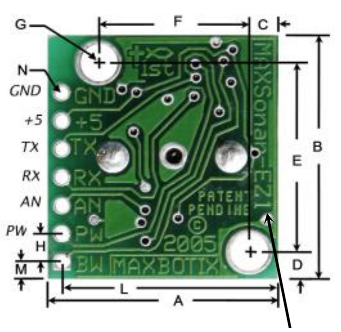
Benefits

- Very low cost ultrasonic rangefinder
- Reliable and stable range data
- Quality beam characteristics
- Mounting holes provided on the circuit board
- Very low power ranger, excellent for multiple sensor or battery-based systems
- Fast measurement cycles
- Sensor reports the range reading directly and frees up user processor
- Choose one of three sensor outputs
- Triggered externally or internally

Applications and Uses

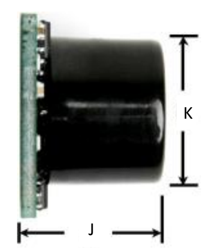
- UAV blimps, micro planes and some helicopters
- Bin level measurement
- Proximity zone detection
- People detection
- Robot ranging sensor
- Autonomous navigation
- Multi-sensor arrays
- Distance measuring
- Long range object detection
- Wide beam sensitivity

LV-MaxSonar-EZ Mechanical Dimensions



A	0.785"	19.9 mm	H	0.100"	2.54 mm
B	0.870"	22.1 mm	J	0.610"	15.5 mm
C	0.100"	2.54 mm	K	0.645"	16.4 mm
D	0.100"	2.54 mm	L	0.735"	18.7 mm
E	0.670"	17.0 mm	M	0.065"	1.7 mm
F	0.510"	12.6 mm	N	0.038" dia.	1.0 mm dia.
G	0.124" dia.	3.1 mm dia.	weight, 4.3 grams		

Part Number	MB1000	MB1010	MB1020	MB1030	MB1040
Paint Dot Color	Black	Brown	Red	Orange	Yellow



Close Range Operation

Applications requiring 100% reading-to-reading reliability should not use MaxSonar sensors at a distance closer than 6 inches. Although most users find MaxSonar sensors to work reliably from 0 to 6 inches for detecting objects in many applications, MaxBotix® Inc. does not guarantee operational reliability for objects closer than the minimum reported distance. Because of ultrasonic physics, these sensors are unable to achieve 100% reliability at close distances.

Warning: Personal Safety Applications

We do not recommend or endorse this product be used as a component in any personal safety applications. This product is not designed, intended or authorized for such use. These sensors and controls do not include the self-checking redundant circuitry needed for such use. Such unauthorized use may create a failure of the MaxBotix® Inc. product which may result in personal injury or death. MaxBotix® Inc. will not be held liable for unauthorized use of this component.

About Ultrasonic Sensors

Our ultrasonic sensors are in air, non-contact object detection and ranging sensors that detect objects within an area. These sensors are not affected by the color or other visual characteristics of the detected object. Ultrasonic sensors use high frequency sound to detect and localize objects in a variety of environments. Ultrasonic sensors measure the time of flight for sound that has been transmitted to and reflected back from nearby objects. Based upon the time of flight, the sensor then outputs a range reading.

Pin Out Description

Pin 1-BW-*Leave open or hold low for serial output on the TX output. When BW pin is held high the TX output sends a pulse (instead of serial data), suitable for low noise chaining.

Pin 2-PW- This pin outputs a pulse width representation of range. The distance can be calculated using the scale factor of 147uS per inch.

Pin 3-AN- Outputs analog voltage with a scaling factor of (Vcc/512) per inch. A supply of 5V yields ~9.8mV/in. and 3.3V yields ~6.4mV/in. The output is buffered and corresponds to the most recent range data.

Pin 4-RX– This pin is internally pulled high. The LV-MaxSonar-EZ will continually measure range and output if RX data is left unconnected or held high. If held low the sensor will stop ranging. Bring high for 20uS or more to command a range reading.

Pin 5-TX- When the *BW is open or held low, the TX output delivers asynchronous serial with an RS232 format, except voltages are 0-Vcc. The output is an ASCII capital "R", followed by three ASCII character digits representing the range in inches up to a maximum of 255, followed by a carriage return (ASCII 13). The baud rate is 9600, 8 bits, no parity, with one stop bit. Although the voltage of 0-Vcc is outside the RS232 standard, most RS232 devices have sufficient margin to read 0-Vcc serial data. If standard voltage level RS232 is desired, invert, and connect an RS232 converter such as a MAX232. When BW pin is held high the TX output sends a single pulse, suitable for low noise chaining. (no serial data)

Pin 6-+5V- Vcc – Operates on 2.5V - 5.5V. Recommended current capability of 3mA for 5V, and 2mA for 3V.

Pin 7-GND- Return for the DC power supply. GND (& Vcc) must be ripple and noise free for best operation.

Range "0" Location

The LV-MaxSonar-EZ reports the range to distant targets starting from the front of the sensor as shown in the diagram below.

Range Zero

The range is measured from the front of the transducer.

In general, the LV-MaxSonar-EZ will report the range to the leading edge of the closest detectable object. Target detection has been characterized in the sensor beam patterns.

Sensor Minimum Distance

The sensor minimum reported distance is 6-inches (15.2 cm). However, the LV-MaxSonar-EZ will range and report targets to the front sensor face. Large targets closer than 6-inches will typically range as 6-inches.

Sensor Operation from 6-inches to 20-inches

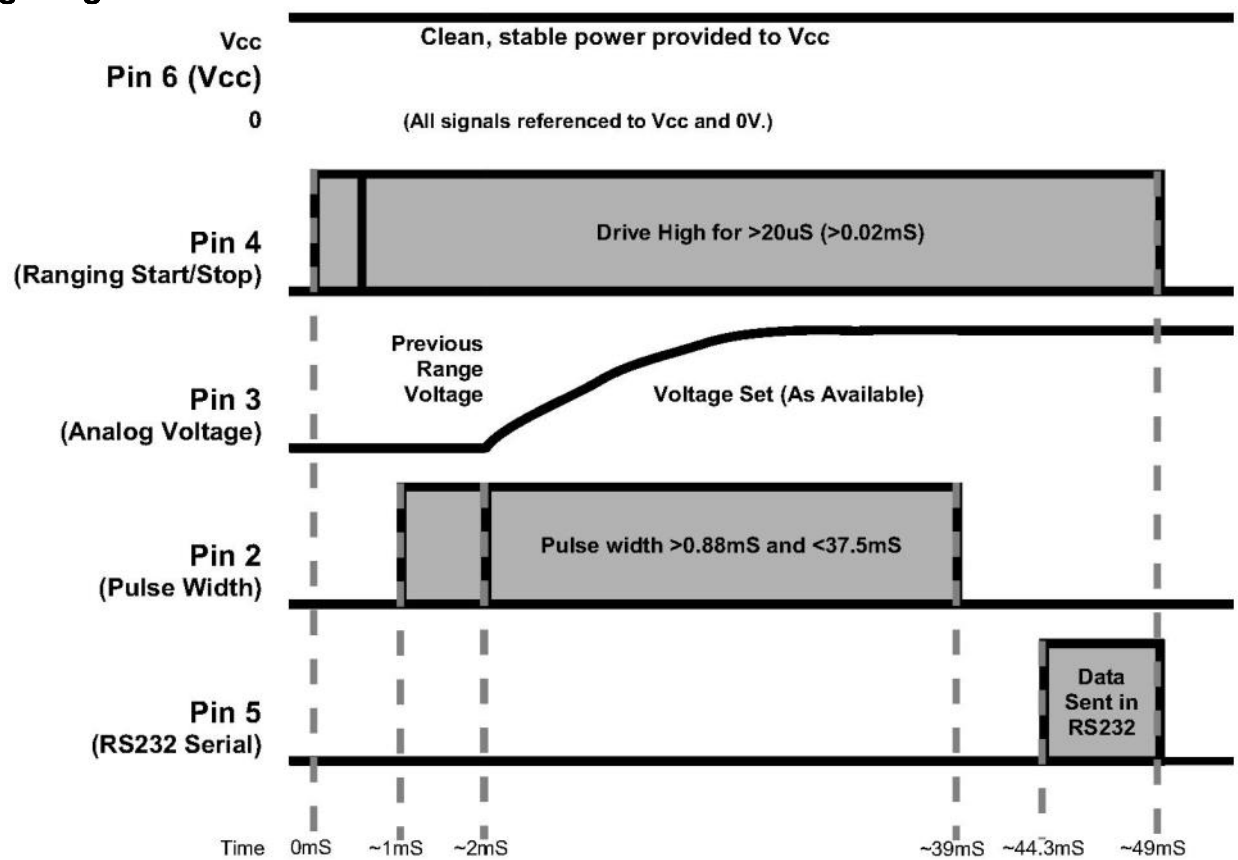
Because of acoustic phase effects in the near field, objects between 6-inches and 20-inches may experience acoustic phase cancellation of the returning waveform resulting in inaccuracies of up to 2-inches. These effects become less prevalent as the target distance increases, and has not been observed past 20-inches.

General Power-Up Instruction

Each time the LV-MaxSonar-EZ is powered up, it will calibrate during its first read cycle. The sensor uses this stored information to range a close object. It is important that objects not be close to the sensor during this calibration cycle. The best sensitivity is obtained when the detection area is clear for fourteen inches, but good results are common when clear for at least seven inches. If an object is too close during the calibration cycle, the sensor may ignore objects at that distance.

The LV-MaxSonar-EZ does not use the calibration data to temperature compensate for range, but instead to compensate for the sensor ringdown pattern. If the temperature, humidity, or applied voltage changes during operation, the sensor may require recalibration to reacquire the ringdown pattern. Unless recalibrated, if the temperature increases, the sensor is more likely to have false close readings. If the temperature decreases, the sensor is more likely to have reduced up close sensitivity. To recalibrate the LV-MaxSonar-EZ, cycle power, then command a read cycle.

Timing Diagram



Timing Description

250mS after power-up, the LV-MaxSonar-EZ is ready to accept the RX command. If the RX pin is left open or held high, the sensor will first run a calibration cycle (49mS), and then it will take a range reading (49mS). After the power up delay, the first reading will take an additional ~100mS. Subsequent readings will take 49mS. The LV-MaxSonar-EZ checks the RX pin at the end of every cycle. Range data can be acquired once every 49mS.

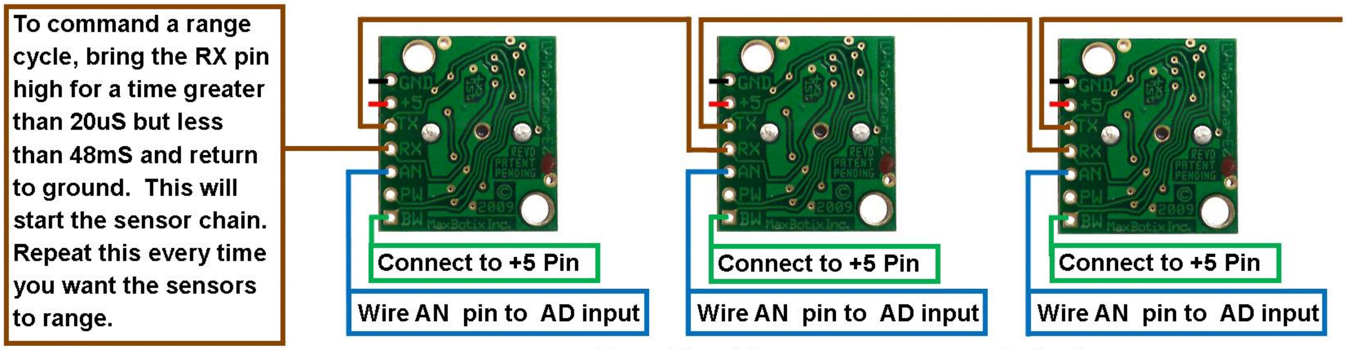
Each 49mS period starts by the RX being high or open, after which the LV-MaxSonar-EZ sends the transmit burst, after which the pulse width pin (PW) is set high. When a target is detected the PW pin is pulled low. The PW pin is high for up to 37.5mS if no target is detected. The remainder of the 49mS time (less 4.7mS) is spent adjusting the analog voltage to the correct level. When a long distance is measured immediately after a short distance reading, the analog voltage may not reach the exact level within one read cycle. During the last 4.7mS, the serial data is sent.

The LV-MaxSonar-EZ timing is factory calibrated to one percent at five volts, and in use is better than two percent. In addition, operation at 3.3V typically causes the objects range, to be reported, one to two percent further than actual.

Using Multiple Sensors in a single system

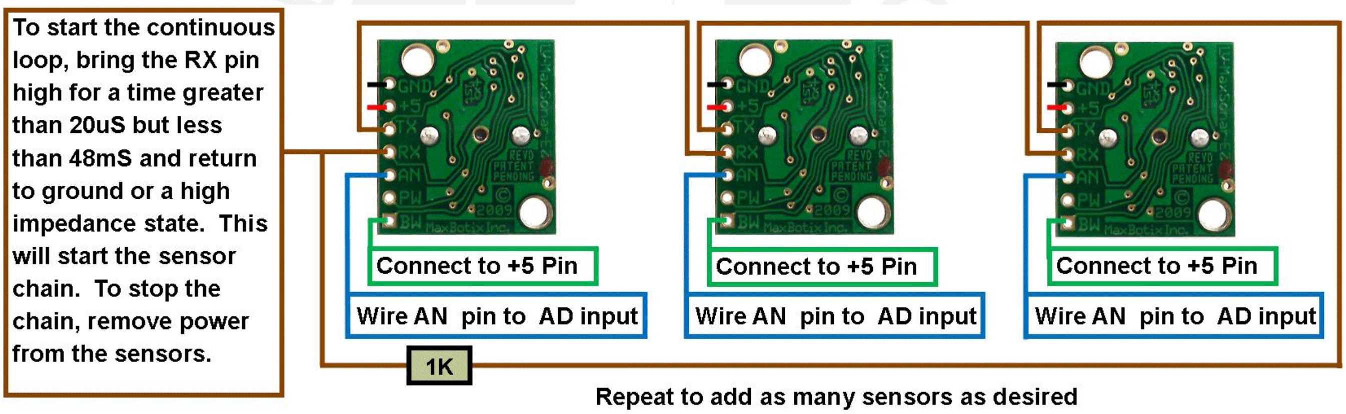
When using multiple ultrasonic sensors in a single system, there can be interference (cross-talk) from the other sensors. MaxBotix Inc., has engineered and supplied a solution to this problem for the LV-MaxSonar-EZ sensors. The solution is referred to as chaining. We have 3 methods of chaining that work well to avoid the issue of cross-talk.

The first method is AN Output Commanded Loop. The first sensor will range, then trigger the next sensor to range and so on for all the sensor in the array. Once the last sensor has ranged, the array stops until the first sensor is triggered to range again. Below is a diagram on how to set this up.



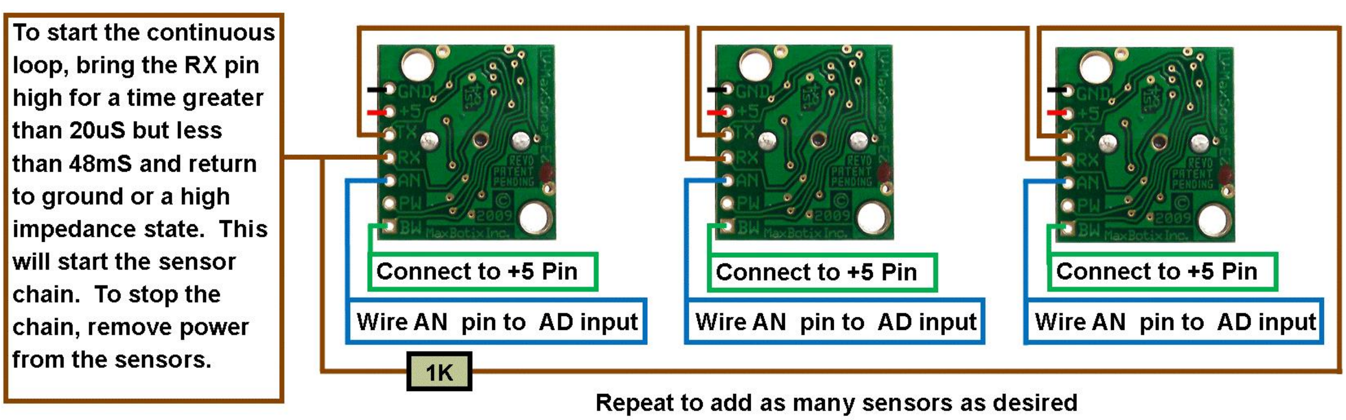
Repeat to add as many sensors as desired

The next method is AN Output Constantly Looping. The first sensor will range, then trigger the next sensor to range and so on for all the sensor in the array. Once the last sensor has ranged, it will trigger the first sensor in the array to range again and will continue this loop indefinitely. Below is a diagram on how to set this up.



Repeat to add as many sensors as desired

The final method is AN Output Simultaneous Operation. This method does not work in all applications and is sensitive to how the other sensors in the array are positioned in comparison to each other. Testing is recommend to verify this method will work for your application. All the sensors RX pins are conned together and triggered at the same time

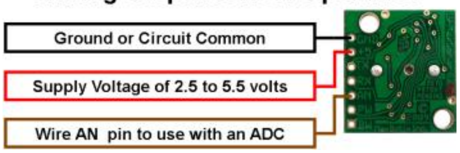


Repeat to add as many sensors as desired

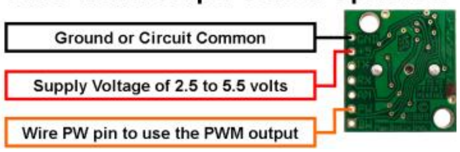
Independent Sensor Operation

The LV-MaxSonar-EZ sensors have the capability to operate independently when the user desires. When using the LV-MaxSonar-EZ sensors in single or independent sensor operation, it is easiest to allow the sensor to free-run. Free-run is the default mode of operation for all of the MaxBotix Inc., sensors. The LV-MaxSonar-EZ sensors have three separate outputs that update the range data simultaneously: Analog Voltage, Pulse Width, and RS232 Serial. Below are diagrams on how to connect the sensor for each of the three outputs when operating in a single or independent sensor operating environment.

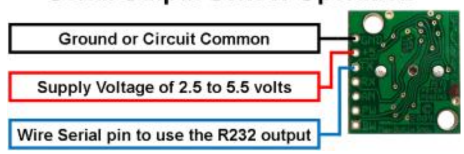
Analog Output Sensor Operation



Pulse Width Output Sensor Operation



Serial Output Sensor Operation



Selecting an LV-MaxSonar-EZ

Different applications require different sensors. The LV-MaxSonar-EZ product line offers varied sensitivity to allow you to select the best sensor to meet your needs.

The LV-MaxSonar-EZ Sensors At a Glance

People Detection Wide Beam High Sensitivity		Best Balance		Large Targets Narrow Beam Noise Tolerance
MB1000	MB1010	MB1020	MB1030	MB1040

The diagram above shows how each product balances sensitivity and noise tolerance. This does not effect the maximum range, pin outputs, or other operations of the sensor. To view how each sensor will function to different sized targets reference the LV-MaxSonar-EZ Beam Patterns.

Background Information Regarding our Beam Patterns

Each LV-MaxSonar-EZ sensor has a calibrated beam pattern. Each sensor is matched to provide the approximate detection pattern shown in this datasheet. This allows end users to select the part number that matches their given sensing application. Each part number has a consistent field of detection so additional units of the same part number will have similar beam patterns. The beam plots are provided to help identify an estimated detection zone for an application based on the acoustic properties of a target versus the plotted beam patterns.

Each beam pattern is a 2D representation of the detection area of the sensor. The beam pattern is actually shaped like a 3D cone (having the same detection pattern both vertically and horizontally). Detection patterns for dowels are used to show the beam pattern of each sensor. Dowels are long cylindered targets of a given diameter. The dowels provide consistent target detection characteristics for a given size target which allows easy comparison of one MaxSonar sensor to another MaxSonar sensor.

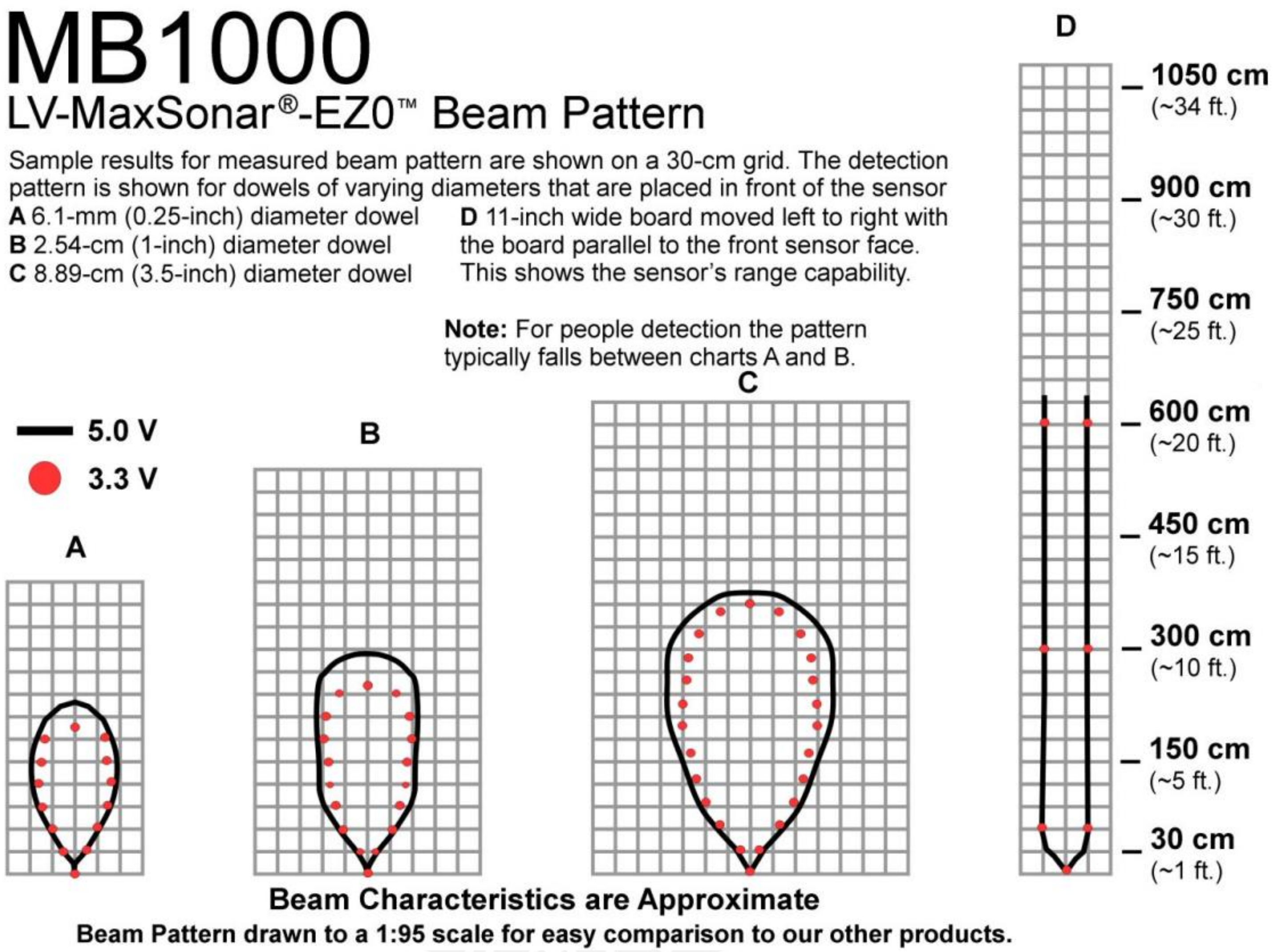
People Sensing: For users that desire to detect people, the detection area to the 1-inch diameter dowel, in general, represents the area that the sensor will reliably detect people.

For each part number, the four patterns (A, B, C, and D) represent the detection zone for a given target size. Each beam pattern shown is determined by the sensor's part number and target size.

The actual beam angle changes over the full range. Use the beam pattern for a specific target at any given distance to calculate the beam angle for that target at the specific distance. Generally, smaller targets are detected over a narrower beam angle and a shorter distance. Larger targets are detected over a wider beam angle and a longer range.

MB1000 LV-MaxSonar-EZ0

The LV-MaxSonar-EZ0 is the highest sensitivity and widest beam sensor of the LV-MaxSonar-EZ sensor series. The wide beam makes this sensor ideal for a variety of applications including people detection, autonomous navigation, and wide beam applications.



MB1000 Features and Benefits

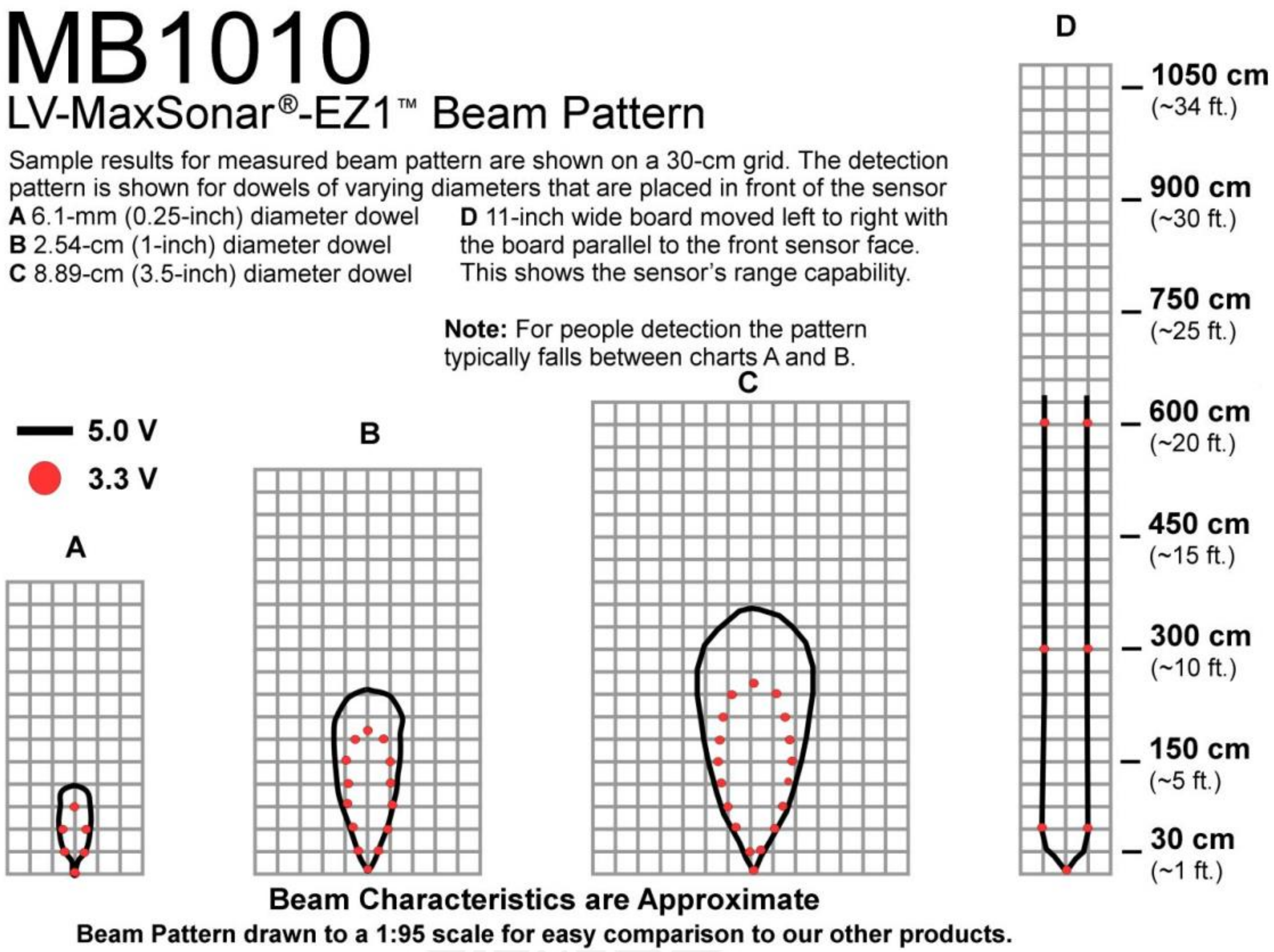
- Widest and most sensitive beam pattern in LV-MaxSonar-EZ line
- Low power consumption
- Easy to use interface
- Will pick up the most noise clutter when compared to other sensors in the LV-MaxSonar-EZ line
- Detects smaller objects
- Best sensor to detect soft object in LV-MaxSonar-EZ line
- Requires use of less sensors to cover a given area
- Can be powered by many different types of power sources
- Can detect people up to approximately 10 feet

MB1000 Applications and Uses

- Great for people detection
- Security
- Motion detection
- Used with battery power
- Autonomous navigation
- Educational and hobby robotics
- Collision avoidance

MB1010 LV-MaxSonar-EZ1

The LV-MaxSonar-EZ1 is the original MaxSonar product. This is our most popular indoor ultrasonic sensor and is a great low-cost general-purpose sensor for a customer not sure of which LV-MaxSonar-EZ sensor to use.



MB1010 Features and Benefits

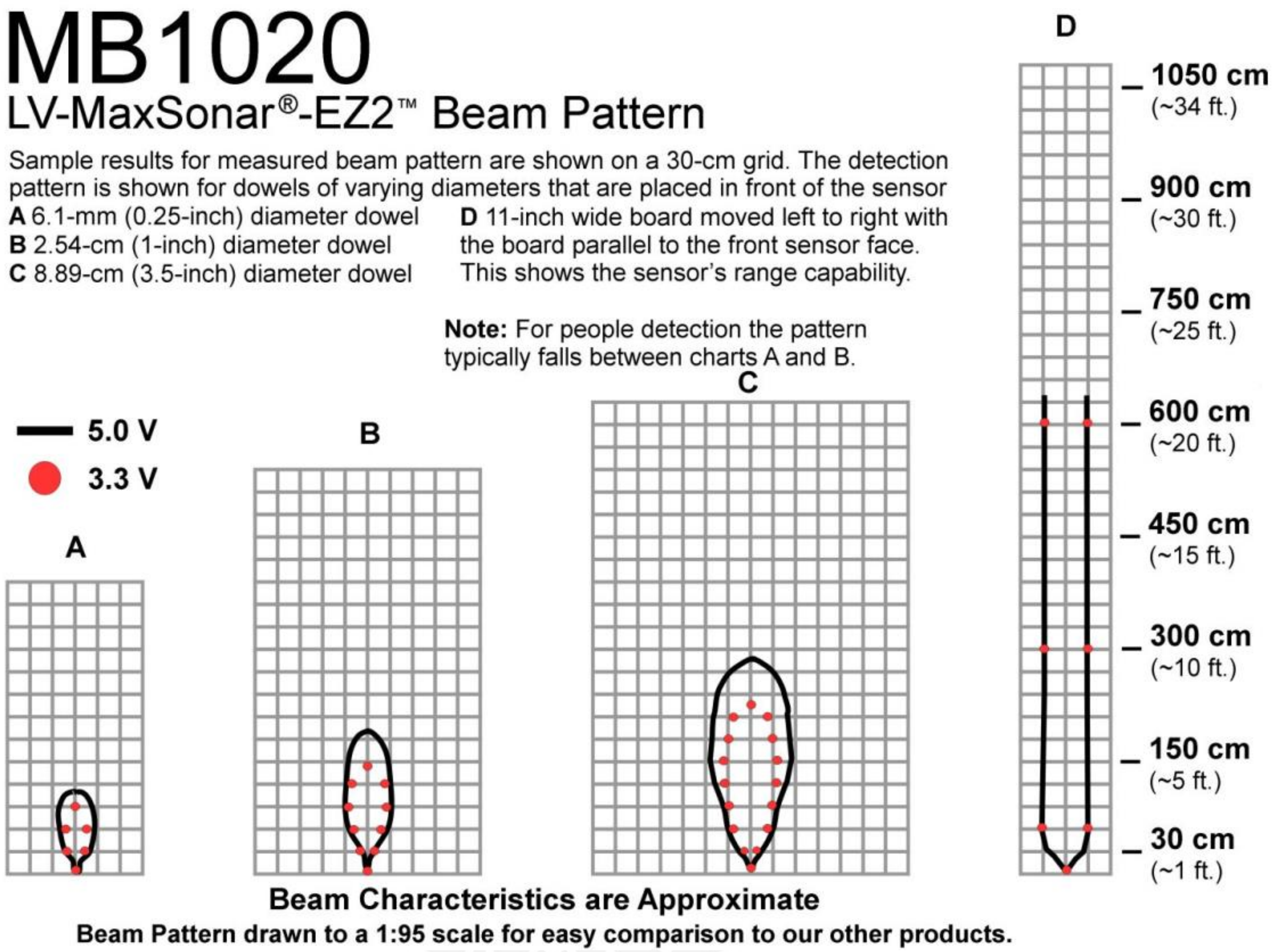
- Most popular ultrasonic sensor
- Low power consumption
- Easy to use interface
- Can detect people to 8 feet
- Great balance between sensitivity and object rejection
- Can be powered by many different types of power sources

MB1010 Applications and Uses

- Great for people detection
- Security
- Motion detection
- Used with battery power
- Autonomous navigation
- Educational and hobby robotics
- Collision avoidance

MB1020 LV-MaxSonar-EZ2

The LV-MaxSonar-EZ2 is a good compromise between sensitivity and side object rejection. The LV-MaxSonar-EZ2 is an excellent choice for applications that require slightly less side object detection and sensitivity than the MB1010 LV-MaxSonar-EZ1.



MB1020 Features and Benefits

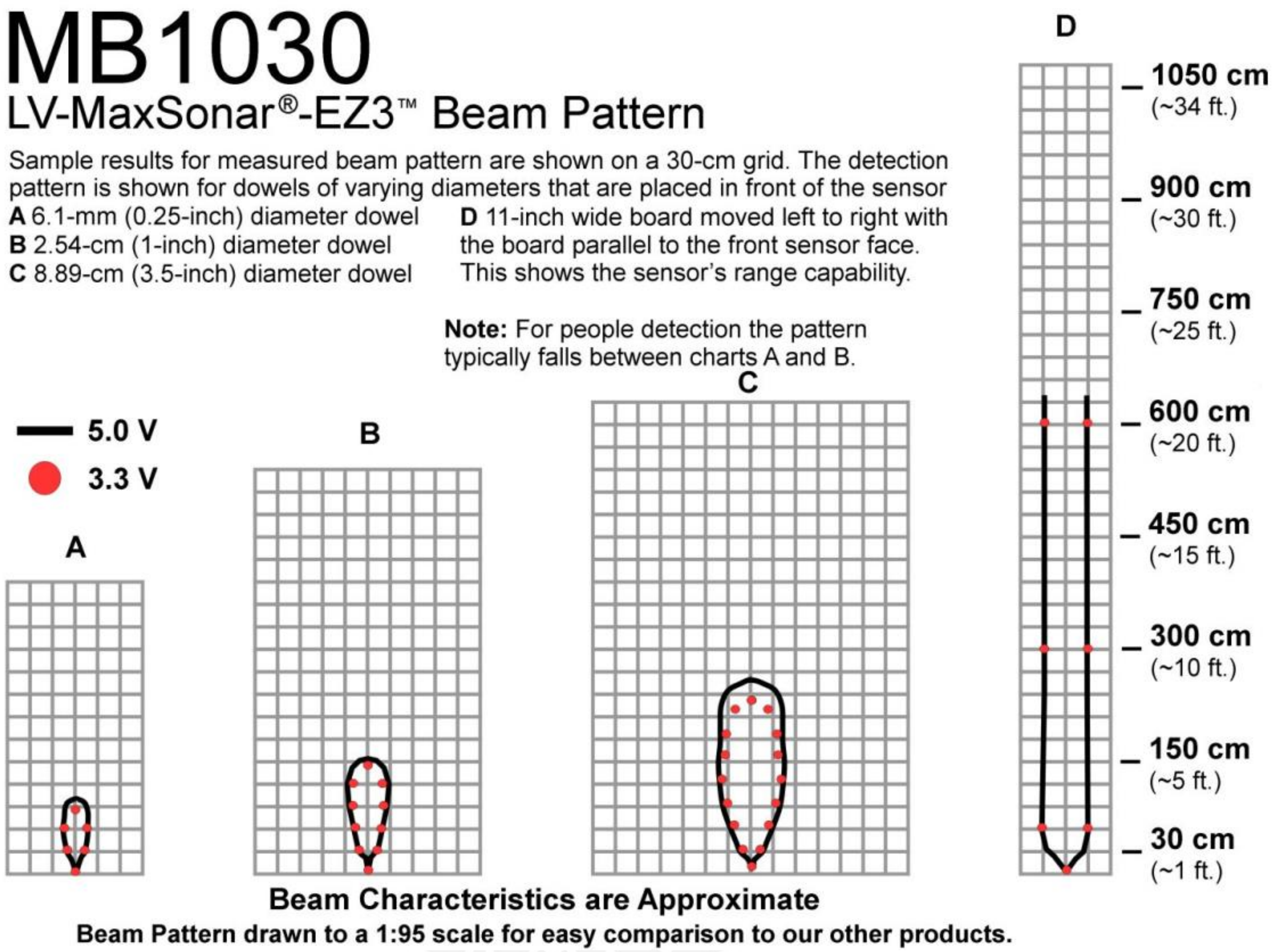
- Great for applications where the MB1010 is too sensitive.
- Excellent side object rejection
- Can be powered by many different types of power sources
- Can detect people up to approximately 6 feet

MB1020 Applications and Uses

- Landing flying objects
- Used with battery power
- Autonomous navigation
- Educational and hobby robotics
- Large object detection

MB1030 LV-MaxSonar-EZ3

The LV-MaxSonar-EZ3 is a narrow beam sensor with good side object rejection. The LV-MaxSonar-EZ3 has slightly wider beam width than theMB1040 LV-MaxSonar-EZ4 which makes it a good choice for when the LV-MaxSonar-EZ4 does not have enough sensitivity for the application.



MB1030 Features and Benefits

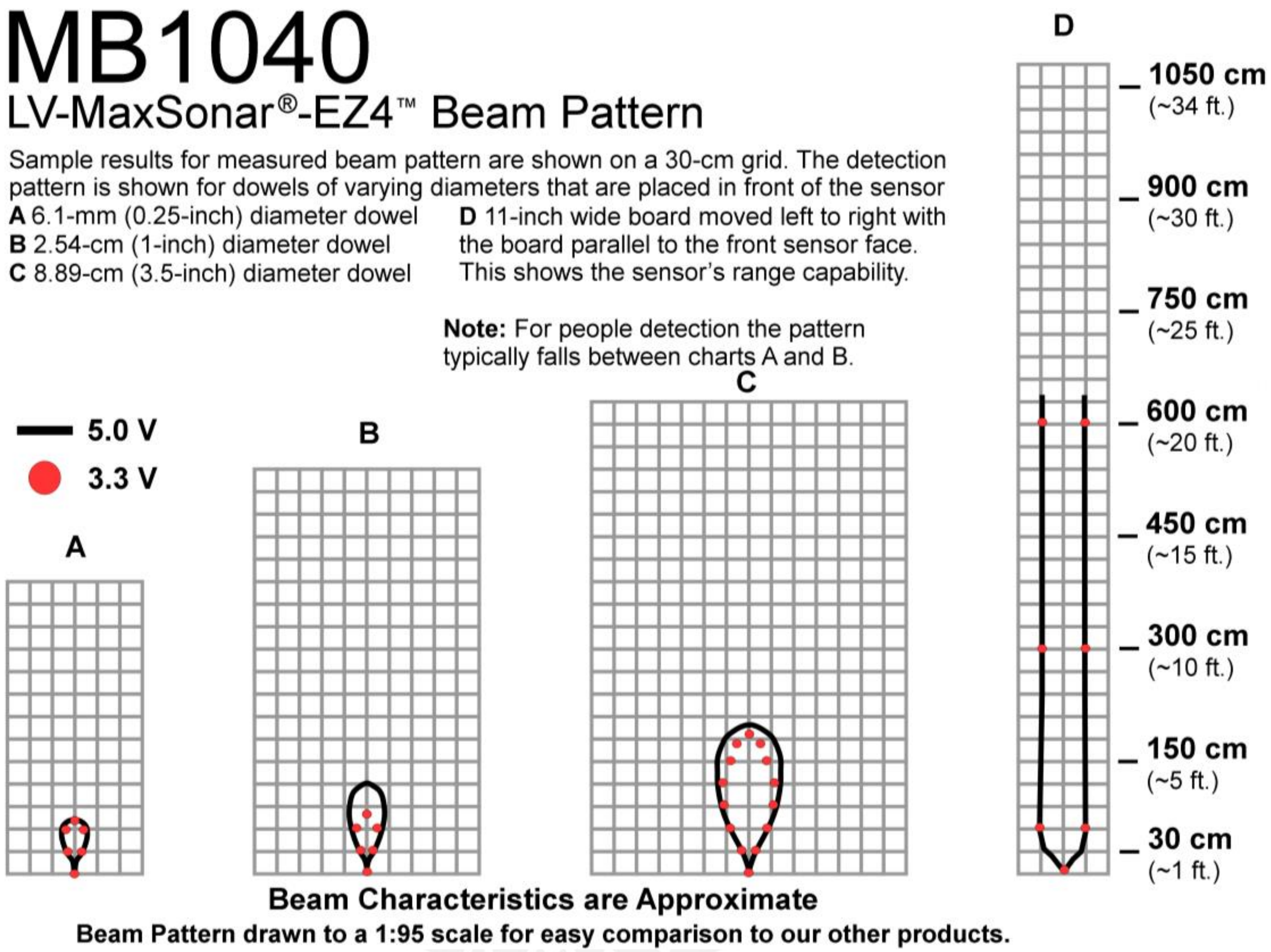
- Excellent side object rejection
- Low power consumption
- Easy to use interface
- Great for when MB1040 is not sensitive enough
- Large object detection
- Can be powered by many different types of power sources
- Can detect people up to approximately 5 feet

MB1030 Applications and Uses

- Landing flying objects
- Used with battery power
- Autonomous navigation
- Educational and hobby robotics

MB1040 LV-MaxSonar-EZ4

The LV-MaxSonar-EZ4 is the narrowest beam width sensor that is also the least sensitive to side objects offered in the LV-MaxSonar-EZ sensor line. The LV-MaxSonar-EZ4 is an excellent choice when only larger objects need to be detected.



MB1040 Features and Benefits

- Best side object rejection in the LV-MaxSonar-EZ sensor line
- Low power consumption
- Easy to use interface
- Best for large object detection
- Can be powered by many different types of power sources
- Can detect people up to approximately 4 feet

MB1040 Applications and Uses

- Landing flying objects
- Used with battery power
- Autonomous navigation
- Educational and hobby robotics
- Collision avoidance

Have the right sensor for your application?

Select from this product list for Protected and Non-Protected Environments.

Protected Environments

1 mm Resolution
HRLV-MaxSonar-EZ

1 in Resolution
LV-MaxSonar-EZ
LV-ProxSonar-EZ

1 cm Resolution
XL-MaxSonar-EZ
XL-MaxSonar-AE
XL-MaxSonar-EZL
XL-MaxSonar-AEL

1 mm Resolution
HRUSB-MaxSonar-EZ

1 in Resolution
USB-ProxSonar-EZ

Non-Protected Environments

1 mm Resolution
HRXL-MaxSonar-WR
HRXL-MaxSonar-WRS
HRXL-MaxSonar-WRT
HRXL-MaxSonar-WRM
HRXL-MaxSonar-WRMT
HRXL-MaxSonar-WRL
HRXL-MaxSonar-WRLT
HRXL-MaxSonar-WRLS
HRXL-MaxSonar-WRLST
SCXL-MaxSonar-WR
SCXL-MaxSonar-WRS
SCXL-MaxSonar-WRT
SCXL-MaxSonar-WRM
SCXL-MaxSonar-WRMT
SCXL-MaxSonar-WRL
SCXL-MaxSonar-WRLT
SCXL-MaxSonar-WRLS
SCXL-MaxSonar-WRLST
4-20HR-MaxSonar-WR

1 cm Resolution
XL-MaxSonar-WR
XL-MaxSonar-WRL
XL-MaxSonar-WRA
XL-MaxSonar-WRLA
I2CXL-MaxSonar-WR

1 mm Resolution
HRXL-MaxSonar-WRC
HRXL-MaxSonar-WRCT

1 cm Resolution
XL-MaxSonar-WRC
XL-MaxSonar-WRCA
I2CXL-MaxSonar-WRC

1 cm Resolution
UCXL-MaxSonar-WR
UCXL-MaxSonar-WRC
I2C-UCXL-MaxSonar-WR

Chemical Shield

F-Option. Available for WR models except UCXL. For additional protection when necessary in hazardous chemical environments.

Accessories — More information is online.

MB7954 — Shielded Cable

The MaxSonar Connection Wire is used to reduce interference caused by electrical noise on the lines. This cable is a great solution to use when running the sensors at a long distance or in an area with a lot of EMI and electrical noise.

MB7950 — XL-MaxSonar-WR Mounting Hardware

The MB7950 Mounting Hardware is selected for use with our outdoor ultrasonic sensors. The mounting hardware includes a steel lock nut and two O-ring (Buna-N and Neoprene) each optimal for different applications.

MB7955 / MB7956 / MB7957 / MB7958 / MB7972 — HR-MaxTemp

The HR-MaxTemp is an optional accessory for the HR-MaxSonar. The HR-MaxTemp connects to the HR-MaxSonar for automatic temperature compensation without self heating.

MB7961 — Power Supply Filter

The power supply filter is recommended for applications with unclean power or electrical noise.

MB7962 / MB7963 / MB7964 / MB7965 — Micro-B USB Connection Cable

The MB7962, MB7963, MB7964 and MB7965 Micro-B USB cables are USB 2.0 compliant and backwards compatible with USB 1.0 standards. Varying lengths.

MB7973 — CE Lightning/Surge Protector

The MB7973 adds protection required to meet the Lightning/Surge IEC61000-4-5 specification.

Anexo 5: Sensor Ultrasónico HC-SR04

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

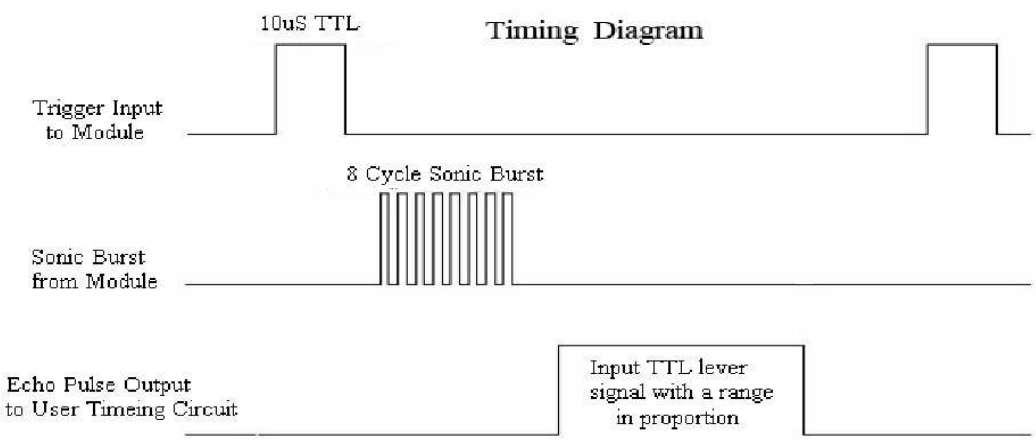
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.Elecfreaks.com

Anexo 6: Sensor Inercial IMU 10 DOF

INTRODUCTION

At the beginning,the inertial measurement unit is an electronic device that measures and reports on a craft's velocity, orientation, and gravitational forces, using a combination of accelerometers,gyroscopes, and magnetometers. Now IMUs are commonly used in the Human-computer interaction(HCI), navigational purposes and balancing technology used in the Segway Personal Transporter as we all known.

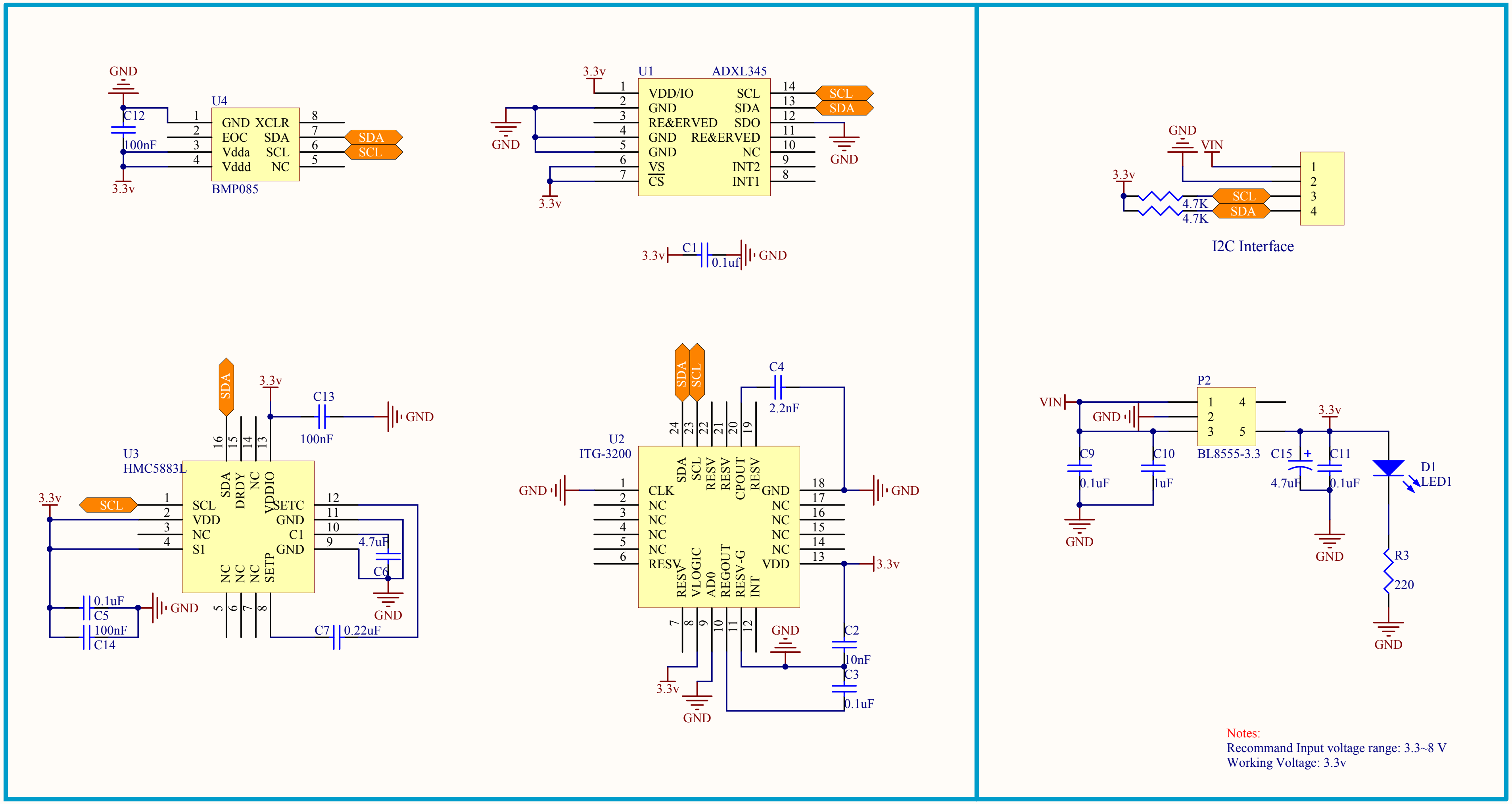
The 10 DOF(degrees of the freedom) sensor is a compact and low cost IMU from DFRobot. It integrates the ADXL345 accelerometer, the HMC5883L magnetometer,the ITG-3205 gyro and the BMP085 barometric pressure sensor. It's suitable for most of the controlling system because of the small dimension. The mounting holes make it possible to provide the highly accurate and stable sensor data.It embeds a low noise LDO regulator for supplying a wide range power input.Works with 3~8 volts power input.Certainly,the 10 DOF sensor is directly compatible with your Arduino boards.

APPLICATIONS

- Aircraft
- Balancing robots
- Indoor inertial navigation
- Altimeter
- Human–computer Interaction (HCI)

SPECIFICATION

- Wide power input range from 3 to 8 volts
- Low noise LDO regulator
- Low cost IMU
- Interface: I2C
- M3x2 holes for easily mounted on your mobile platforms,robots,HCI or UAVs
- LED power indication
- Integrate 10 dof sensors
  - Adxl345 accelerometer
  - ITG3200 gyro
  - HMC5883L Compass
  - BMP085 pressure sensor
- Compact size design and easy-to-use
- Compatible with Arduino controllers
- Electricity gold PCB
- Size: 26x18mm

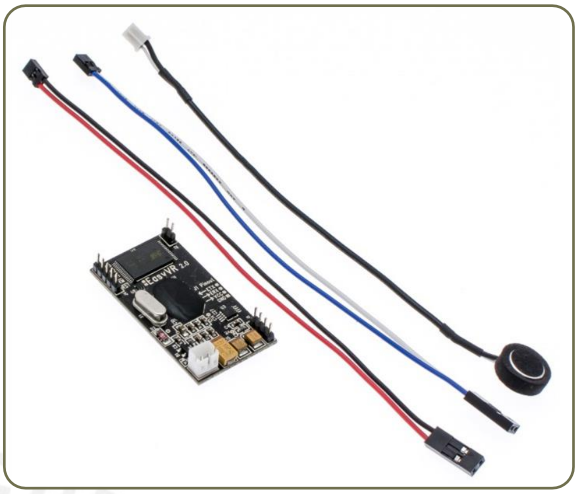


Title		
Fly maple -10DOF		
Size: A4	Number	Revision: v1.0
Date: 11/10/2012		Sheet of
File: E:\Backup of Project Codes\..\10 Dof sensor Rev 0 2.SchDoc		Drawn By:

Anexo 7: Easy VR 2.0

EasyVR Module

Product Description



EasyVR 2.0 is a multi-purpose speech recognition module designed to easily add versatile, robust and cost effective speech recognition capabilities to almost any application.

The EasyVR 2.0 module can be used with any host with an UART interface powered at 3.3V – 5V, such as PIC and Arduino boards. Some application examples include home automation, such as voice controlled light switches, locks, curtains or kitchen appliances, or adding "hearing" to the most popular robots on the market.

EasyVR 2.0 Features

- Supports up to 28 custom Speaker Independent (SI) command vocabularies[1,2] (new to EasyVR 2.0)
  Supported Languages:
  - US English
  - UK English
  - German
  - French
  - Italian
  - LA Spanish
  - Korean
  - Japanese
- Supports up to 32 user-defined Speaker Dependent (SD) triggers or commands as well as Voice Passwords. SD custom commands can be spoken in ANY language.
- A variety of built-in Speaker Independent (SI) commands for ready to run basic controls, in the followings languages:
  - English (US)
  - Italian
  - German
  - French
  - Spanish
  - Japanese
- SonicNet technology for wireless communications between modules or any other sound source (Audio CD, DVD, MP3 Player) [2] (new to EasyVR 2.0)
- DTMF tone generation[2] (new to EasyVR 2.0)
- Easy-to-use and simple Graphical User Interface to program Voice Commands and audio.
- Module can be used with any host with an UART interface (powered at 3.3V - 5V)
- Simple and robust documented serial protocol to access and program through the host board
- 3 x GPIO lines (IO1, IO2, IO3) that can be controlled by new protocol commands.
- PWM audio output that directly supports 8Ω speakers.
- Sound playback of up to 9 minutes of recorded sounds or speech.

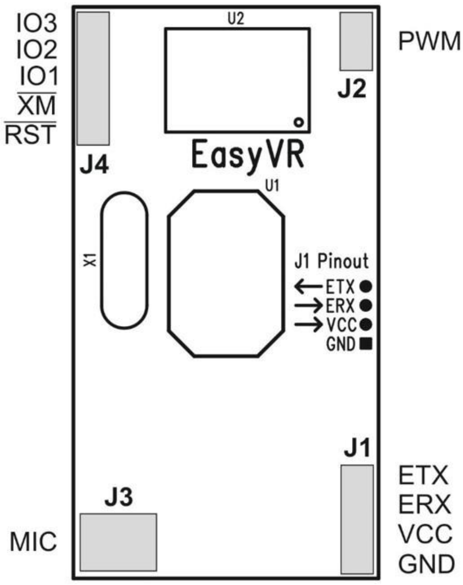
---

[1] A *QuickT2SI™ Lite* license (sold separately) is required to enable creation of Speaker Independent vocabularies (maximum 12 commands per set).

[2] Custom Speaker Independent vocabularies, SonicNet™ and DTMF generation are available since firmware version 2 (a free update from version 1 is available for download).

Technical specifications

Physical dimensions and pin assignment



Connector	Number	Name	Type	Description
J1	**1**	GND	-	Ground
	2	VCC	I	Voltage DC input
	3	ERX	I	Serial Data Receive (TTL level)
	4	ETX	O	Serial Data Transmit (TTL level)
J2	**1-2**	PWM	O	Differential audio output (can directly drive 8Ω speaker)
J3	**1**	MIC_RET	-	Microphone reference ground
	2	MIC_IN	I	Microphone input signal
J4	**1**	/RST	I	Active low asynchronous reset (internal 100K pull-up)
	2	/XM	I	Boot select (internal 1K pull-down)
	3	IO1	I/O	General purpose I/O (**3.0 VDC** TTL level)
	4	IO2	I/O	General purpose I/O (**3.0 VDC** TTL level)
	5	IO3	I/O	General purpose I/O (**3.0 VDC** TTL level)

Note: the GPIO (J4.3, J4.4, and J4.5) are at nominal 3.0VDC level. Do not connect 5VDC directly to these pins!

Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VCC	Voltage DC Input	3.3	5.0	5.5	V
Ta	Ambient Operating Temperature Range	0	25	70	°C
ERX	Serial Port Receive Data	0	-	VCC	V
ETX	Serial Port Transmit Data	0	-	VCC	V

Electrical Characteristics

These are applicable to J4 pins only, including IO1-3, /XM and /RST.

Symbol	Parameter		Min	Typ	Max	Unit
V_{IH}	Input High Voltage		2.4	3.0	3.3	V
V_{IL}	Input Low Voltage		-0.1	0.0	0.75	V
I_{IL}	Input Leakage Current (0 < V_{IO} < 3V, Hi-Z Input)			<1	10	μA
R_{PU}	Pull-up Resistance	Strong		10		kΩ
		Weak		200		kΩ
V_{OH}	Output High Voltage (I_{OH} = -5 mA)		2.4			V
V_{OL}	Output Low Voltage (I_{OL} = 8 mA)				0.6	V

Power Supply Requirements

Symbol	Parameter	Min	Typ	Max	Unit
I_{Sleep}	Sleep current		< 1		mA
I_{Oper}	Operating current		12		mA
$I_{Speaker}$	Audio playback current (with 8Ω speaker)		180		mA (RMS)

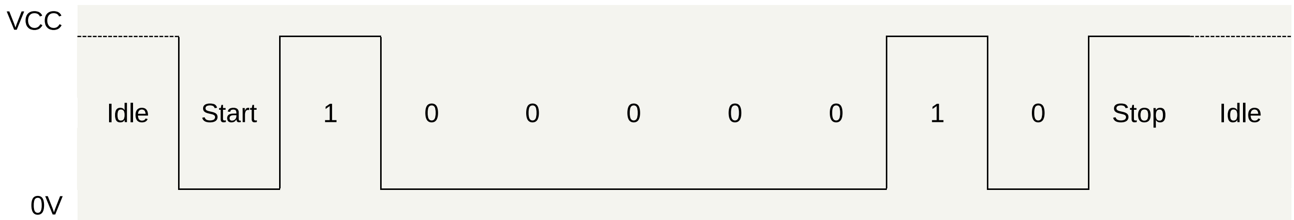
Serial Interface

The EasyVR is a "slave" module communicating via an asynchronous serial interface (commonly known as UART interface), with the following features:

- Baud Rate: **9600** (default), 19200, 38700, 57600, 115200
- Frame: **8** Data bits, **N**o parity, **1** Stop bit

The receiver input data line is ERX, while the transmitter output data line is ETX. No handshake lines are used.

Example of a serial data frame representing character "A" (decimal 65 or hexadecimal 41):



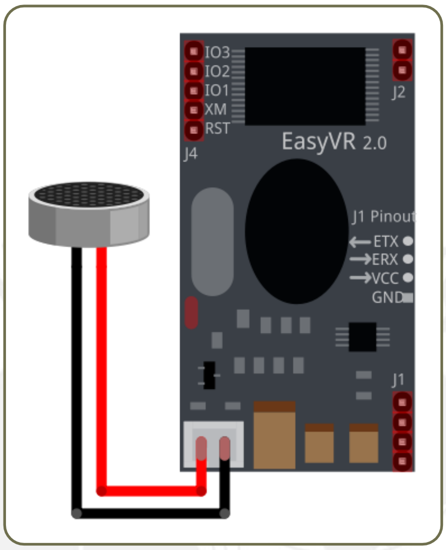
See also chapter *Communication Protocol* later on this manual for communication details.

Microphone

The microphone provided with the EasyVR module is an omnidirectional electret condenser microphone (Horn EM9745P-382):

- Sensitivity -38dB (0dB=1V/Pa @1KHz)
- Load Impedance 2.2K
- Operating Voltage 3V
- Almost flat frequency response in the range 100Hz – 20kHz

If you use a microphone with different specifications the recognition accuracy may be adversely affected. No other kind of microphone is supported by the EasyVR module.

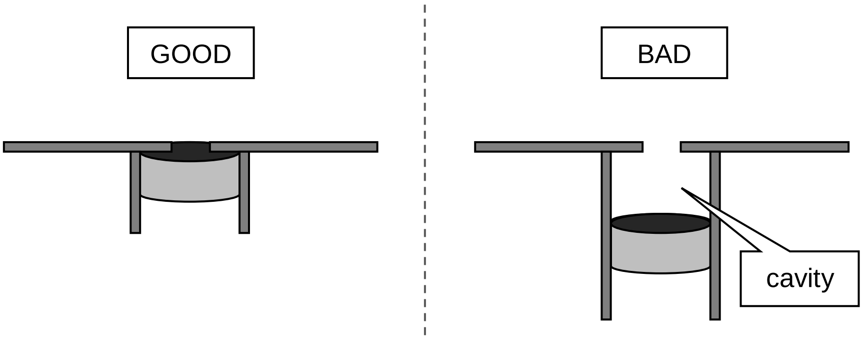


Note: Vocal commands should be given from about 60cm from the microphone, but you can try at greater distances by talking louder.

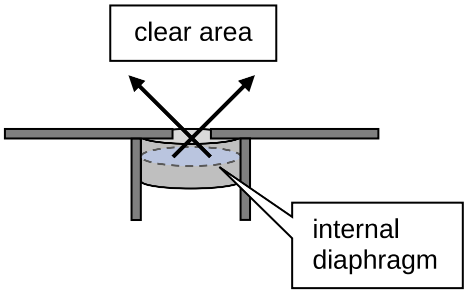
Positioning guidelines

Please note that improper acoustic positioning of the microphone will reduce recognition accuracy. Many mechanical arrangements are possible for the microphone element, and some will work better than others. When mounting the microphone in the final device, keep in mind the following guidelines:

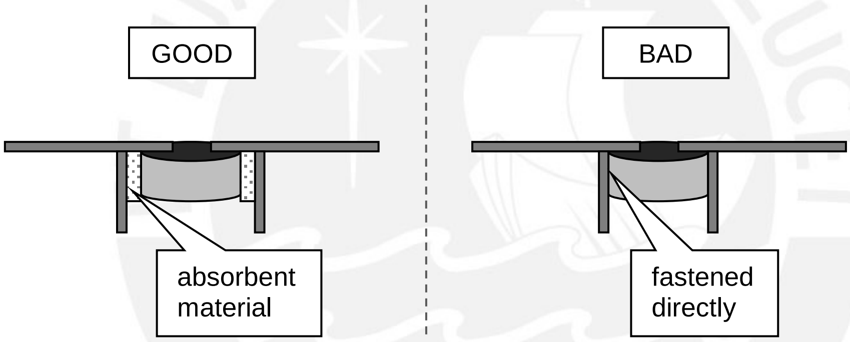
1. **Flush Mounting** - The microphone element should be positioned as close to the mounting surface as possible and should be fully seated in the plastic housing. There must be no airspace between the microphone element and the housing. Having such airspace can lead to acoustic resonance, which can reduce recognition accuracy.



2. **No Obstructions, Large Hole** - The area in front of the microphone element must be kept clear of obstructions to avoid interference with recognition. The diameter of the hole in the housing in front of the microphone should be at least 5 mm. Any necessary plastic surface in front of the microphone should be as thin as possible, being no more than 0.7 mm, if possible.



3. **Insulation** - The microphone should be acoustically isolated from the housing if possible. This can be accomplished by surrounding the microphone element with a spongy material such as rubber or foam. The provided microphone has this kind of insulating foam. The purpose is to prevent auditory noises produced by handling or jarring the device from being “picked up” by the microphone. Such extraneous noises can reduce recognition accuracy.

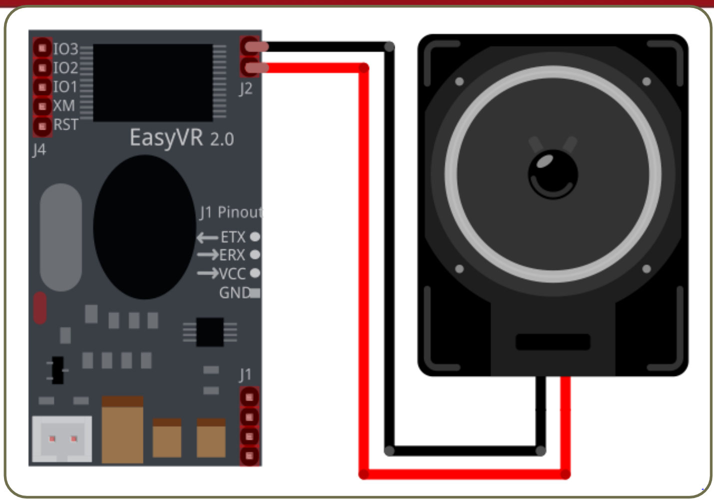


4. **Distance** - If the microphone is moved from 15 cm to 30 cm from the speaker’s mouth, the signal power decreases by a factor of four. The difference between a loud and a soft voice can also be more than a factor of four. Although the internal preamplifier of the EasyVR compensates for a wide dynamic range of input signal strength, if its range is exceeded, the user application can provide feedback to the speaker about the voice volume (see appendix *Error codes*).

Audio Output

The EasyVR audio output interface is capable of directly driving an 8Ω speaker. It can also be connected to an external audio amplifier to drive lower impedance loudspeakers.

Note: Connecting speakers with lower impedance directly to the module may permanently damage the EasyVR audio output or the whole module.



It is possible to connect higher impedance loads such as headphones, provided that you scale down the output power according to the speaker ratings, for example using a series resistor. The exact resistor value depends on the headphone power ratings and the desired output volume (usually in the order of 10kΩ).

Note: Connecting headphone speakers directly to the EasyVR audio output may damage your hearing.

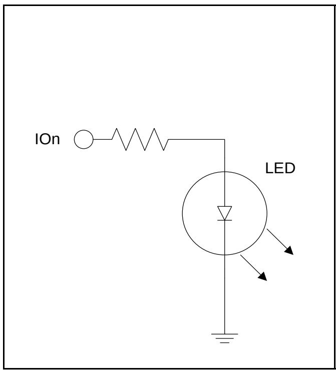
General Purpose I/O

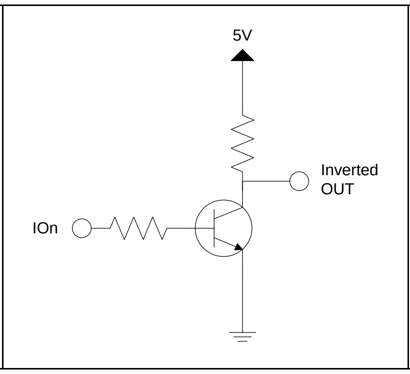
Since the EasyVR communication interface takes two pins of the host controller, a few spare I/O pins are provided, which can be controlled with the communication protocol, to get those pins back for basic tasks, such as lighting an LED.

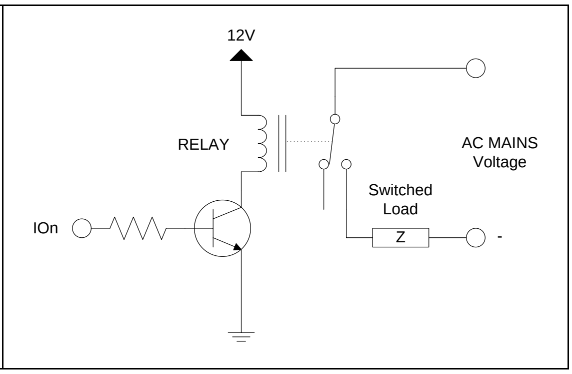
The three I/O pins IO1–IO3 are connected directly to the embedded microcontroller on the EasyVR module, so they are referenced to the internal 3.0V regulated power supply. If you need to interface to circuits using a different supply, there are a number of solutions you can adopt. Some of these are outlined below (here IOn indicates any one of the three I/O pins of the EasyVR).

Use a pin as an output

All the I/O pins are inputs with weak internal pull-up after power on. You must explicitly configure a pin before you can use it as an output (see the example code *Use general purpose I/O pins*).







I/O pin directly driving a low-current LED	*I/O pin connected to high impedance 5V circuit (such as MCU input pin)*	*I/O pin switching a load on a high voltage line using a 12V relay*

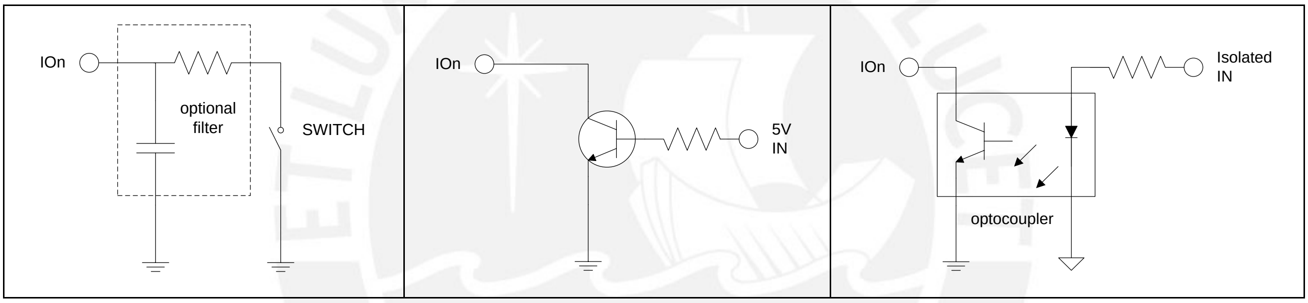
The exact components values in these circuits may vary. You need to calculate required values for your application and choice of components. For example, resistor value for the LED circuit can be calculated approximately as:

Where V_{LED} is the LED forward voltage, as reported on the LED datasheet, at the driving current I_{OH} (see section *Electrical Characteristics*). Let's assume a typical low-current LED has a V_F=1.8V at 5mA, the resistor value is:

Now stay on the safe side and choose a slightly larger resistor, such as 150Ω.
If you want to drive higher current LEDs, you need a circuit like the second one, where you put the LED between the output resistor and the collector of the NPN transistor.

Use a pin as an input

All the I/O pins are inputs with weak internal pull-up after power on or reset. You may also configure the pin to have a strong pull-up or no pull-up at all (see the example code *Use general purpose I/O pins*).



I/O pin connected to a switch (or switching sensor) | *I/O pin connected 5V source (such as MCU output pin)* | *I/O pin with isolated input (for safety circuits)*

All these circuits assume the EasyVR pin has been configured with an internal pull-up (passive components value can be adjusted to account for weak or strong pull-up).

Disabling the internal pull-up could be used to put the pin in high-impedance state, for example to simulate a tri-state or open-drain output port.

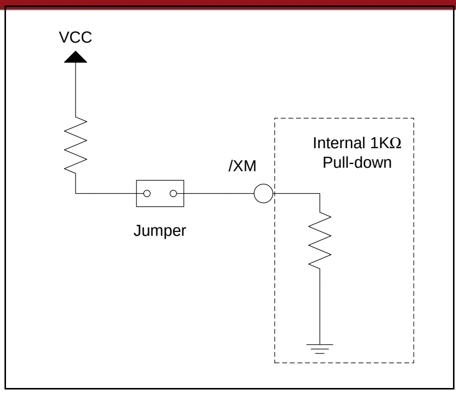
Again, you should refer to the manufacturer's datasheet when interfacing any external components and to calculate required resistors values or other passive components.

Flash Update

The EasyVR module includes a boot loader that allows to update the firmware and to download new sound tables to the on-board memory.

The *boot mode* is activated by keeping the **/XM** signal to a high logical level at power on or reset. This can be easily done with a jumper (or switch) taking the signal to a suitable pull-up resistor.

To download a firmware update, a sound table or a custom grammar to the EasyVR, power on the module with the jumper closed. For normal operation, just leave the jumper open. Do not change the jumper position while the module is already powered on. It is safe to change **/XM** level while the module is reset (**/RST** low).



Boot mode selection circuit

The pull-up resistor value to use depends on the VCC power supply voltage. For the voltage of the **/XM** pin when the jumper is closed (short) the following relation holds (note you have a voltage divider circuit):

Now if you want **/XM** to be at 3V (logic high) and solving for *R*, you get:

That makes 100Ω for 3.3V and around 680Ω for 5V power supplies. Other kinds of circuit are possible, that is just an example and one of the simplest to realize.

To learn how to download new sound tables or custom grammars to your EasyVR 2.0 module, have a look at the section *Using Custom Data*.

Anexo 8: Logitech Webcam C615 y ángulo de colocación

System Requirements

- Windows® 8
- Windows® 7
- Windows Vista®
- Windows® XP (SP2+)
- Mac OS® X 10.5 - 10.6x

Basic requirements:

- 1 GHz
- 512 MB RAM or more
- 200 MB hard drive space
- Internet connection
- USB 1.1 port (2.0 recommended)

For HD 720p video calling and Full HD 1080p video recording:

- 2.4 GHz Intel® Core™2 Duo
- 2 GB RAM
- 200 MB hard drive space
- USB 2.0 port
- 1 Mbps upload speed or higher
- 1280 x 720 screen resolution

Warranty Information

2-year limited hardware warranty

Package Contents

- Webcam with 3-foot cable
- 3-foot extension cable
- User documentation

Part Number

- PN 960-000733

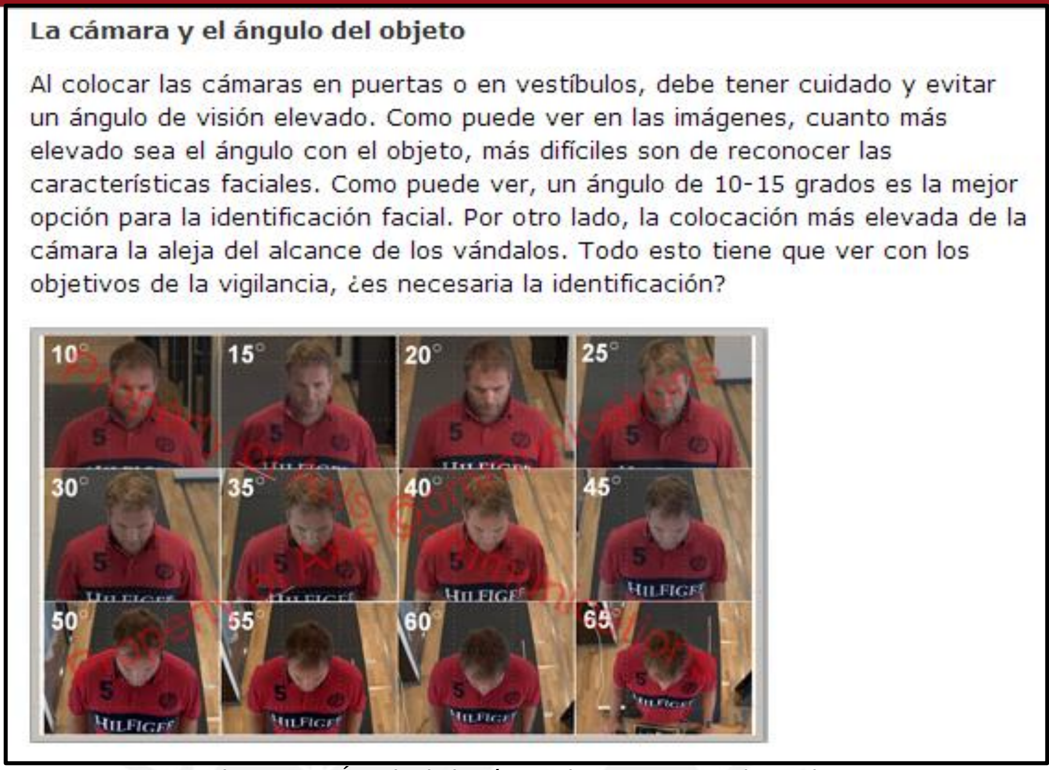
Technical Specifications

- Full HD 1080p video capture (up to 1920 x 1080 pixels) with recommended system
- HD video calling (1280 x 720 pixels) with recommended system
- Logitech Fluid Crystal™ Technology*
- Autofocus
- Photos: Up to 8 megapixels (software enhanced)
- Built-in mics with automatic noise reduction
- Hi-Speed USB 2.0 certified (recommended)
- Universal clip fits laptops, LCD or CRT monitors

Logitech webcam software:*

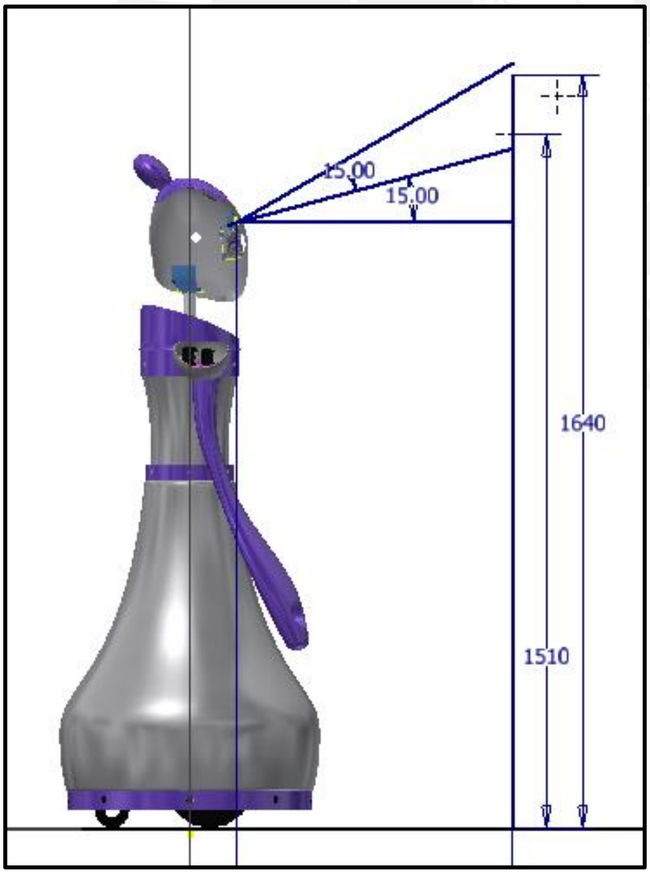
- Pan, tilt, and zoom controls
- Video and photo capture
- Face tracking
- Motion detection

** Requires installation of software available for download at www.logitech.com/support/hd-webcam-c615.*


La cámara y el ángulo del objeto

Al colocar las cámaras en puertas o en vestíbulos, debe tener cuidado y evitar un ángulo de visión elevado. Como puede ver en las imágenes, cuanto más elevado sea el ángulo con el objeto, más difíciles son de reconocer las características faciales. Como puede ver, un ángulo de 10-15 grados es la mejor opción para la identificación facial. Por otro lado, la colocación más elevada de la cámara la aleja del alcance de los vándalos. Todo esto tiene que ver con los objetivos de la vigilancia, ¿es necesaria la identificación?

Figura 8-1 : Ángulo de la cámara (Fuente: www.axis.com)



En la figura 8-2 se muestra el ángulo de visión de la cámara. Esta está colocada con un ángulo de 15° respecto a la horizontal pues es la manera óptima de colocarla para visualizar los rostros de las personas según la figura 8-1 que fue extraída de una página especializada en cámaras de seguridad [17], el ángulo de visión óptimo es de 30°. El promedio de la altura de las mujeres en el Perú es de 1.51m y el de hombres es de 1.64m, por lo que esta disposición es la adecuada.

Figura 8-2 : Vista frontal (Fuente: Propia)

Anexo 9: Servomotor Dynamixel AX-12A

AX-12/ AX-12+/ AX-12A

Part Photo

[AX-12/ 12+]

[AX-12A]

※ AX-12+ is the improved version of existing AX-12; the design of circuit, material, and wheel gear are specially improved.
※ AX-12A is a new version of the AX-12+ with the same performance but more advanced external design. Only the AX-12A is now being sold.

H/W Specification

- Weight : 53.5g (AX-12/AX-12+), 54.6g (AX-12A)
- Dimension : 32mm * 50mm * 40mm
- Resolution : 0.29°
- Gear Reduction Ratio : 254 : 1
- Stall Torque : 1.5N.m (at 12.0V, 1.5A)
- No load speed : 59rpm (at 12V)
- Running Degree
  - 0° ~ 300°
  - Endless Turn
- Running Temperature : -5℃ ~ +70℃
- Voltage : 9 ~ 12V (Recommended Voltage 11.1V)
- Command Signal : Digital Packet
- Protocol Type : Half duplex Asynchronous Serial Communication (8bit,1stop,No Parity)
- Link (Physical) : TTL Level Multi Drop (daisy chain type Connector)
- ID : 254 ID (0~253)
- Communication Speed : 7343bps ~ 1 Mbps
- Feedback : Position, Temperature, Load, Input Voltage, etc.
- Material : Engineering Plastic

Stall torque is the maximum instantaneous and static torque
Stable motions are possible with robots designed for loads with 1/5 or less of the stall torque

Control Table

Control Table consists of data regarding the current status and operation, which exists inside of Dynamixel. The user can control Dynamixel by changing data of Control Table via Instruction Packet.

EEPROM and RAM

Data in RAM area is reset to the initial value whenever the power is turned on while data in EEPROM area is kept once the value is set even if the power is turned off.

Address

It represents the location of data. To read from or write data to Control Table, the user should assign the correct address in the Instruction Packet.

Access

Dynamixel has two kinds of data: Read-only data, which is mainly used for sensing, and Read-and-Write data, which is used for driving.

Initial Value

In case of data in the EEPROM Area, the initial values on the right side of the below Control Table are the factory default settings. In case of data in the RAM Area, the initial values on the right side of the above Control Tables are the ones when the power is turned on.

Highest/Lowest Byte

In the Control table, some data share the same name, but they are attached with (L) or (H) at the end of each name to distinguish the address. This data requires 16bit, but it is divided into 8bit each for the addresses (low) and (high). These two addresses should be written with one Instruction Packet at the same time.

Area	Address (Hexadecimal)	Name	Description	Access	Initial Value (Hexadecimal)
EEPROM	0 (0X00)	Model Number(L)	Lowest byte of model number	R	12 (0X0C)
	1 (0X01)	Model Number(H)	Highest byte of model number	R	0 (0X00)
	2 (0X02)	Version of Firmware	Information on the version of firmware	R	-
	3 (0X03)	ID	ID of Dynamixel	RW	1 (0X01)
	4 (0X04)	Baud Rate	Baud Rate of Dynamixel	RW	1 (0X01)
	5 (0X05)	Return Delay Time	Return Delay Time	RW	250 (0XFA)
	6 (0X06)	CW Angle Limit(L)	Lowest byte of clockwise Angle Limit	RW	0 (0X00)
	7 (0X07)	CW Angle Limit(H)	Highest byte of clockwise Angle Limit	RW	0 (0X00)
	8 (0X08)	CCW Angle Limit(L)	Lowest byte of counterclockwise Angle Limit	RW	255 (0XFF)
	9 (0X09)	CCW Angle Limit(H)	Highest byte of counterclockwise Angle Limit	RW	3 (0X03)
	11 (0X0B)	the Highest Limit Temperature	Internal Limit Temperature	RW	70 (0X46)
	12 (0X0C)	the Lowest Limit Voltage	Lowest Limit Voltage	RW	60 (0X3C)
	13 (0X0D)	the Highest Limit Voltage	Highest Limit Voltage	RW	140 (0XBE)
	14 (0X0E)	Max Torque(L)	Lowest byte of Max. Torque	RW	255 (0XFF)
	15 (0X0F)	Max Torque(H)	Highest byte of Max. Torque	RW	3 (0X03)
	16 (0X10)	Status Return Level	Status Return Level	RW	2 (0X02)
	17 (0X11)	Alarm LED	LED for Alarm	RW	36(0x24)

	18 (0X12)	Alarm Shutdown	Shutdown for Alarm	RW	36(0X24)
	24 (0X18)	Torque Enable	Torque On/Off	RW	0 (0X00)
	25 (0X19)	LED	LED On/Off	RW	0 (0X00)
	26 (0X1A)	CW Compliance Margin	CW Compliance margin	RW	1 (0X01)
	27 (0X1B)	CCW Compliance Margin	CCW Compliance margin	RW	1 (0X01)
	28 (0X1C)	CW Compliance Slope	CW Compliance slope	RW	32 (0X20)
	29 (0X1D)	CCW Compliance Slope	CCW Compliance slope	RW	32 (0X20)
	30 (0X1E)	Goal Position(L)	Lowest byte of Goal Position	RW	–
	31 (0X1F)	Goal Position(H)	Highest byte of Goal Position	RW	–
	32 (0X20)	Moving Speed(L)	Lowest byte of Moving Speed	RW	–
	33 (0X21)	Moving Speed(H)	Highest byte of Moving Speed	RW	–
	34 (0X22)	Torque Limit(L)	Lowest byte of Torque Limit	RW	ADD14
R	35 (0X23)	Torque Limit(H)	Highest byte of Torque Limit	RW	ADD15
A	36 (0X24)	Present Position(L)	Lowest byte of Current Position	R	–
M	37 (0X25)	Present Position(H)	Highest byte of Current Position	R	–
	38 (0X26)	Present Speed(L)	Lowest byte of Current Speed	R	–
	39 (0X27)	Present Speed(H)	Highest byte of Current Speed	R	–
	40 (0X28)	Present Load(L)	Lowest byte of Current Load	R	–
	41 (0X29)	Present Load(H)	Highest byte of Current Load	R	–
	42 (0X2A)	Present Voltage	Current Voltage	R	–
	43 (0X2B)	Present Temperature	Current Temperature	R	–
	44 (0X2C)	Registered	Means if Instruction is registered	R	0 (0X00)
	46 (0X2E)	Moving	Means if there is any movement	R	0 (0X00)
	47 (0X2F)	Lock	Locking EEPROM	RW	0 (0X00)
	48 (0X30)	Punch(L)	Lowest byte of Punch	RW	32 (0X20)
	49 (0X31)	Punch(H)	Highest byte of Punch	RW	0 (0X00)

Address Function Help

EEPROM Area

Model Number

It represents the Model Number.

Firmware Version

It represents the firmware version.

ID

It is a unique number to identify Dynamixel.

The range from 0 to 252 (0xFC) can be used, and, especially, 254(0xFE) is used as the Broadcast ID.

If the Broadcast ID is used to transmit Instruction Packet, we can command to all Dynamixels.

> Please be careful not to duplicate the ID of connected Dynamixel.

Baud Rate

It represents the communication speed. 0 to 254 (0xFE) can be used for it.

This speed is calculated by using the below formula.

Speed(BPS) = 2000000/(Data+1)

Data	Set BPS	Target BPS	Tolerance
1	1000000.0	1000000.0	0.000 %
3	500000.0	500000.0	0.000 %
4	400000.0	400000.0	0.000 %
7	250000.0	250000.0	0.000 %
9	200000.0	200000.0	0.000 %
16	117647.1	115200.0	-2.124 %
34	57142.9	57600.0	0.794 %
103	19230.8	19200.0	-0.160 %
207	9615.4	9600.0	-0.160 %

Note : Maximum Baud Rate error of 3% is within the tolerance of UART communication.

Return Delay Time

It is the delay time per data value that takes from the transmission of Instruction Packet until the return of Status Packet.

0 to 254 (0xFE) can be used, and the delay time per data value is 2 usec.

That is to say, if the data value is 10, 20 usec is delayed. The initial value is 250 (0xFA) (i.e., 0.5 msec).

CW/CCW Angle Limit

The angle limit allows the motion to be restrained.

The range and the unit of the value is the same as Goal Position(Address 30, 31).

- CW Angle Limit: the minimum value of Goal Position(Address 30, 31)
- CCW Angle Limit: the maximum value of Goal Position(Address 30, 31)

The following two modes can be set pursuant to the value of CW and CCW.

Operation Type	CW / CCW
Wheel Mode	both are 0
Joint Mode	neither at 0
Multi-turn Mode	both are 4095

The wheel mode can be used to wheel-type operation robots since motors of the robots spin infinitely.

The joint mode can be used to multi-joints robot since the robots can be controlled with specific angles.

The Highest Limit Temperature

Caution : Do not set the temperature lower/higher than the default value.
When the temperature alarm shutdown occurs, wait 20 minutes to cool the temperature before re-use.
Using the product when the temperature is high may and can cause damage.

The Lowest (Highest) Limit Voltage

It is the operation range of voltage.

50 to 250 (0x32 ~ 0x96) can be used. The unit is 0.1V.

For example, if the value is 80, it is 8V.

If Present Voltage (Address42) is out of the range, Voltage Range Error Bit (Bit0) of Status Packet is returned as '1' and Alarm is triggered as set in the addresses 17 and 18.

Max Torque

It is the torque value of maximum output. 0 to 1023 (0x3FF) can be used, and the unit is about 0.1%.

For example, Data 1023 (0x3FF) means that Dynamixel will use 100% of the maximum torque it can produce while Data 512 (0x200) means that Dynamixel will use 50% of the maximum torque. When the power is turned on, Torque Limit (Addresses 34 and 35) uses the value as the initial value.

Status Return Level

It decides how to return Status Packet. There are three ways like the below table.

Value	Return of Status Packet
0	No return against all commands (Except PING Command)
1	Return only for the READ command
2	Return for all commands

When Instruction Packet is Broadcast ID, Status Packet is not returned regardless of Status Return Level.

Alarm LED

Alarm Shutdown

Dynamixel can protect itself by detecting errors occur during the operation.

The errors can be set are as the table below.

Bit	Name	Contents
Bit 7	0	–
Bit 6	Instruction Error	When undefined Instruction is transmitted or the Action command is delivered without the reg_write command
Bit 5	Overload Error	When the current load cannot be controlled with the set maximum torque
Bit 4	CheckSum Error	When the Checksum of the transmitted Instruction Packet is invalid
Bit 3	Range Error	When the command is given beyond the range of usage
Bit 2	OverHeating Error	When the internal temperature is out of the range of operating temperature set in the Control Table
Bit 1	Angle Limit Error	When Goal Position is written with the value that is not between CW Angle Limit and CCW Angle Limit
Bit 0	Input Voltage Error	When the applied voltage is out of the range of operating voltage set in the Control Table

It is possible to make duplicate set since the function of each bit is run by the logic of 'OR'. That is, if 0X05 (binary 00000101) is set, both Input Voltage Error and Overheating Error can be detected.

If errors occur, in case of Alarm LED, the LED blinks; in case of Alarm Shutdown, the motor output becomes 0 % by making the value of Torque Limit(Address 34, 35) as 0.

RAM Area

Torque Enable

Value	Meaning
0	Keeps Torque from generating by interrupting the power of motor.
1	Generates Torque by impressing the power to the motor.

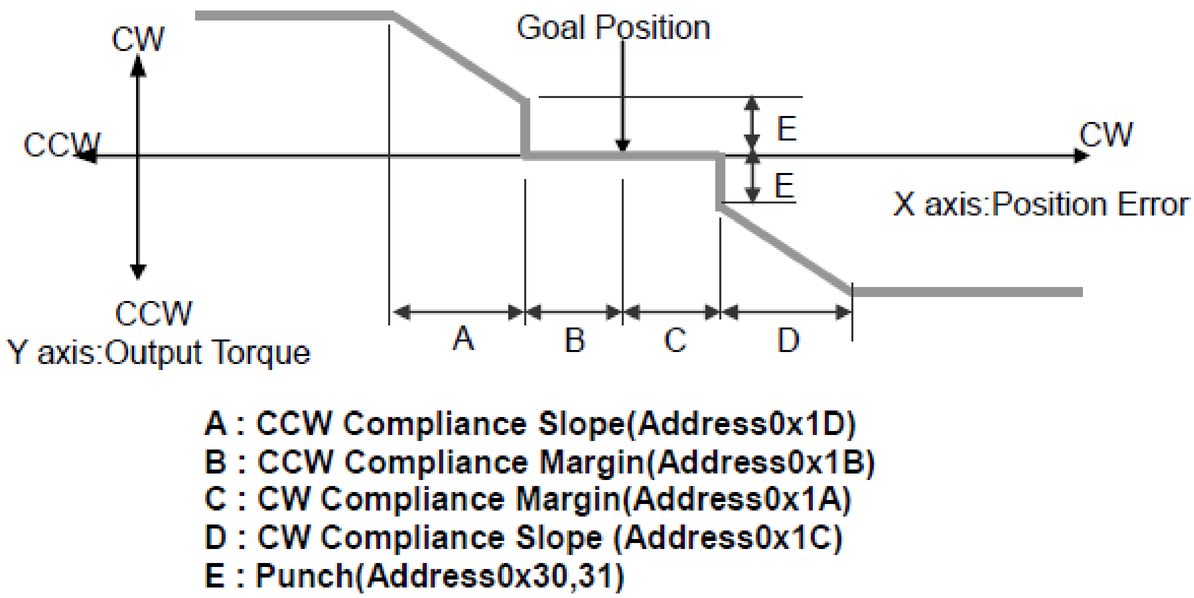
LED

Bit	Meaning	Meaning
bit7		
bit6		
bit5		
bit4		
bit3		
bit2	BLUE LED	When *Bit* is set the blue LED turns on
bit1	GREEN LED	When *Bit* is set the green LED turns on
bit0	RED LED	When *Bit* is set the red LED turns on

Compliance

Compliance is to set the control flexibility of the motor.

The following diagram shows the relationship between output torque and position of the motor.



Compliance Margin

It exists in each direction of CW/CCW and means the error between goal position and present position.

The range of the value is 0~255, and the unit is the same as Goal Position.(Address 30,31)

The greater the value, the more difference occurs.

Compliance Slope

It exists in each direction of CW/CCW and sets the level of Torque near the goal position.

Compliance Slope is set in 7 steps, the higher the value, the more flexibility is obtained.

Data representative value is actually used value. That is, even if the value is set to 25, 16 is used internally as the representative value.

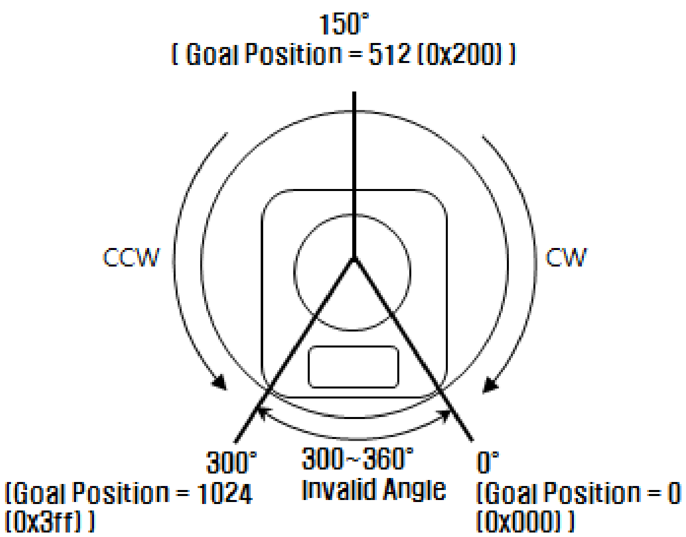
Step	Data Value	Data Representative Value
1	0 (0x00) ~ 3(0x03)	2 (0x02)
2	4(0x04) ~ 7(0x07)	4 (0x04)
3	8(0x08)~15(0x0F)	8 (0x08)
4	16(0x10)~31(0x1F)	16 (0x10)
5	32(0x20)~63(0x3F)	32 (0x20)
6	64(0x40)~127(0x7F)	64 (0x40)
7	128(0x80)~254(0xFE)	128 (0x80)

Goal Position

It is a position value of destination.

0 to 1023 (0x3FF) is available. The unit is 0.29 degree.

If Goal Position is out of the range, Angle Limit Error Bit (Bit1) of Status Packet is returned as '1' and Alarm is triggered as set in Alarm LED/Shutdown.



<The picture above is based on the front of relevant model>

If it is set to Wheel Mode, this value is not used.

Moving Speed

It is a moving speed to Goal Position.

The range and the unit of the value may vary depending on the operation mode.

- Join Mode

0~1023 (0X3FF) can be used, and the unit is about 0.111rpm.
If it is set to 0, it means the maximum rpm of the motor is used without controlling the speed.
If it is 1023, it is about 114rpm.
For example, if it is set to 300, it is about 33.3 rpm.

Notes: Please check the maximum rpm of relevant model in Joint Mode. Even if the motor is set to more than maximum rpm, it cannot generate the torque more than the maximum rpm.

- Wheel Mode
  0~2047(0X7FF) can be used, the unit is about 0.1%.
  If a value in the range of 0~1023 is used, it is stopped by setting to 0 while rotating to CCW direction.
  If a value in the range of 1024~2047 is used, it is stopped by setting to 1024 while rotating to CW direction.
  That is, the 10th bit becomes the direction bit to control the direction.
  In Wheel Mode, only the output control is possible, not speed.
  For example, if it is set to 512, it means the output is controlled by 50% of the maximum output.

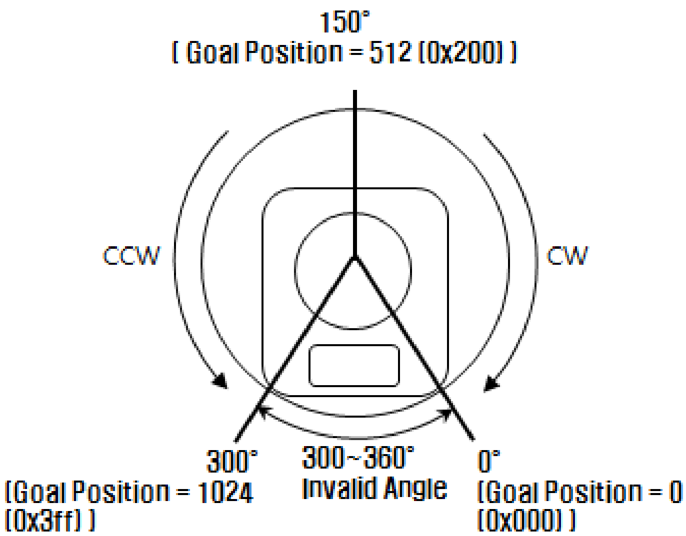
Torque Limit

It is the value of the maximum torque limit.
0 to 1023 (0x3FF) is available, and the unit is about 0.1%.
For example, if the value is 512, it is about 50%; that means only 50% of the maximum torque will be used.
If the power is turned on, the value of Max Torque (Address 14, 15) is used as the initial value.

Notes: If the function of Alarm Shutdown is triggered, the motor loses its torque because the value becomes 0. At this moment, if the value is changed to the value other than 0, the motor can be used again.

Present Position

It is the current position value of Dynamixel.
The range of the value is 0~1023 (0x3FF), and the unit is 0.29 degree.



<The picture above is based on the front of relevant model>

Caution: If it is set to Wheel Mode, the value cannot be used to measure the moving distance and the rotation frequency.

Present Speed

It is the current moving speed.

0~2047 (0X7FF) can be used.

If a value is in the rage of 0~1023, it means that the motor rotates to the CCW direction.

If a value is in the rage of 1024~2047, it means that the motor rotates to the CW direction.

That is, the 10th bit becomes the direction bit to control the direction, and 0 and 1024 are equal.

The unit of this value varies depending on operation mode.

- Joint Mode
  The unit is about 0.111rpm.
  For example, if it is set to 300, it means that the motor is moving to the CCW direction at a rate of about 33.3rpm.
- Wheel Mode
  The unit is about 0.1%.
  For example, if it is set to 512, it means that the torque is controlled by 50% of the maximum torque to the CCW direction.

Present Load

It means currently applied load.

The range of the value is 0~2047, and the unit is about 0.1%.

If the value is 0~1023, it means the load works to the CCW direction.

If the value is 1024~2047, it means the load works to the CW direction.

That is, the 10th bit becomes the direction bit to control the direction, and 1024 is equal to 0.

For example, the value is 512, it means the load is detected in the direction of CCW about 50% of the maximum torque.

BIT	15~11	10	9	8	7	6	5	4	3	2	1	0
Value	0	Load Direction	Data (Load Ratio)									

Load Direction = 0 : CCW Load, Load Direction = 1: CW Load

Notes: Current load is inferred from the internal torque value, not from Torque sensor etc.
For that reason, it cannot be used to measure weight or torque; however, it must be used only to detect which direction the force works.

Present Voltage

It is the size of the current voltage supplied.

This value is 10 times larger than the actual voltage. For example, when 10V is supplied, the data value is 100 (0x64)

Present Temperature

It is the internal temperature of Dynamixel in Celsius.

Data value is identical to the actual temperature in Celsius. For example, if the data value is 85 (0x55), the current internal temperature is 85℃.

Registered Instruction

Value	Meaning
0	There are no commands transmitted by

	REG_WRITE
1	There are commands transmitted by REG_WRITE.

Notes: If ACTION command is executed, the value is changed into 0.

Moving

Value	Meaning
0	Goal position command execution is completed.
1	Goal position command execution is in progress.

Lock

Value	Meaning
0	EEPROM area can be modified.
1	EEPROM area cannot be modified.

Caution: If Lock is set to 1, the power must be turned off and then turned on again to change into 0.

Punch

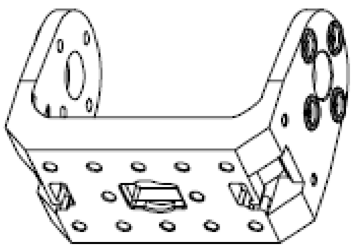
Current to drive motor is at minimum.

Can choose vales from 0x20 to 0x3FF.

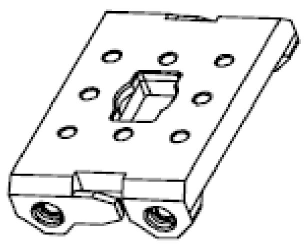
Option Frame

Basic-Offered Frames

Basic-offered frames with AX-12A are as follows.



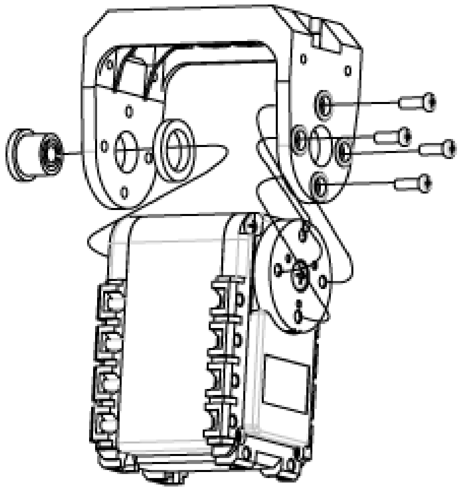
FP04-F2

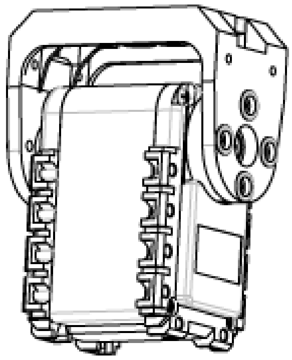


FP04-F3

Applying F2

F2 is applied as below.

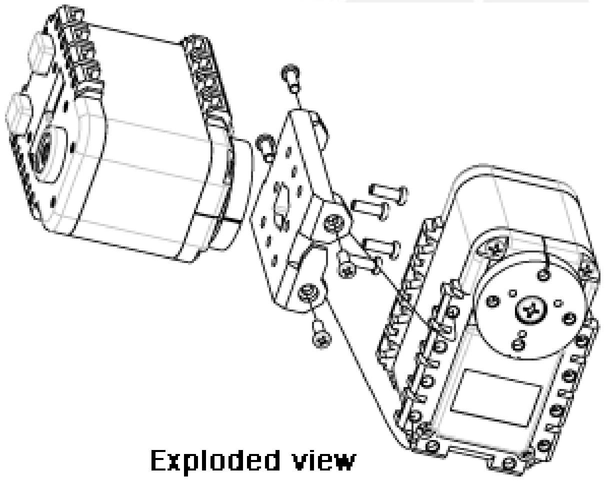


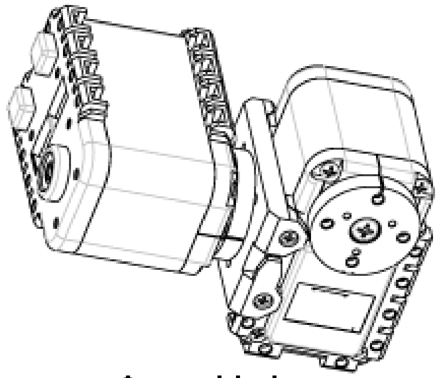


Exploded view Assembled

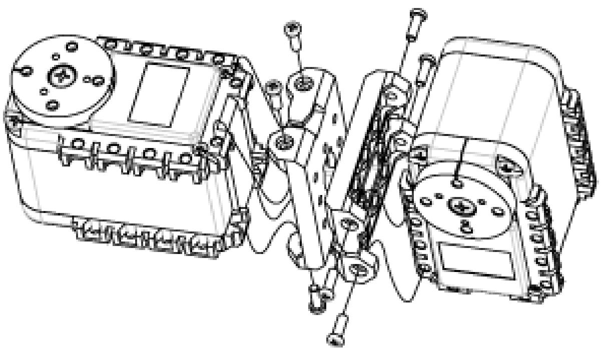
Applying F3

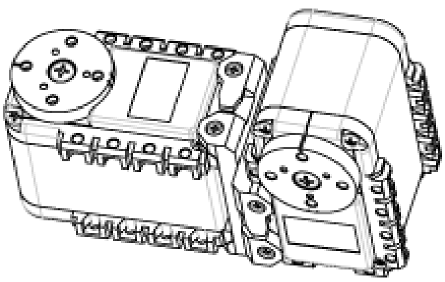
F3 is applied as below. F3 can be connected to 3 sides of AX-12A: Left, Right, and the bottom.





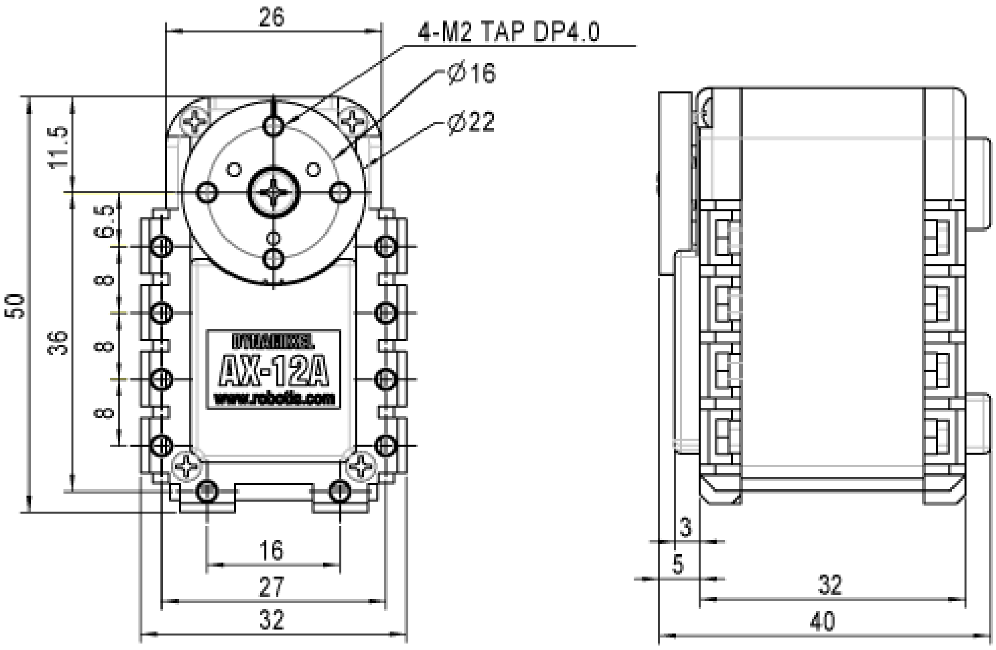
Exploded view Assembled





Exploded view Assembled

Dimension



Videos

HOW TO REPLACE GEARS

Anexo 10: Selección de Actuadores

Para calcular el torque necesario para levantar cada brazo se tiene la figura 10-1 que muestra el ángulo máximo a elevar que sería de 90°. Donde w es la velocidad angular y α es la aceleración angular.

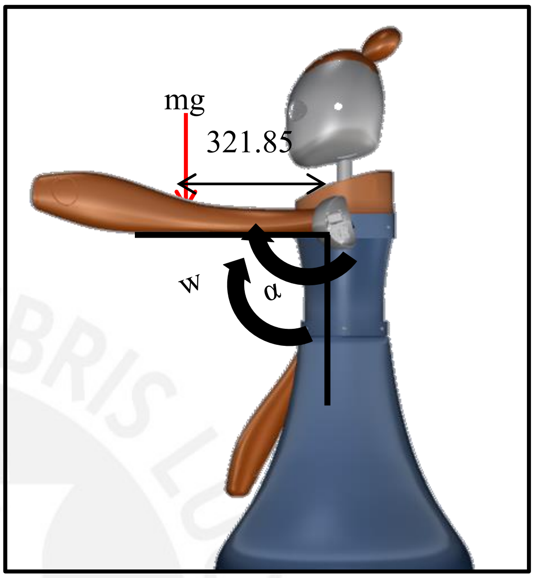


Figura 10-1 : Vista lateral (Fuente: Propia)

En la figura 10-2 se tienen las gráficas de distancia y velocidad versus tiempo, se asume un tiempo de 2 segundos para levantar el brazo al ángulo deseado.

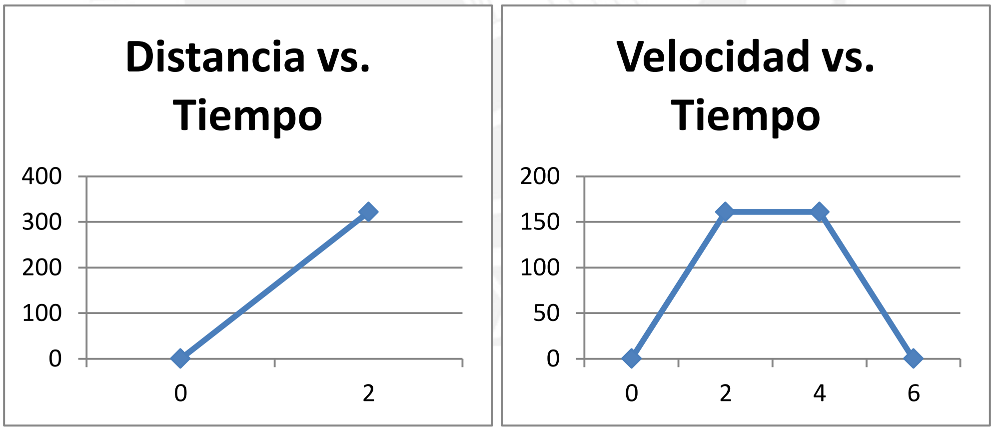


Figura 10-2 : Gráficas (Fuente: Propia)

Entonces se obtiene una velocidad máxima de 160.93 mm/s

Luego para hallar la velocidad promedio:

Entonces para hallar la velocidad y la aceleración angular

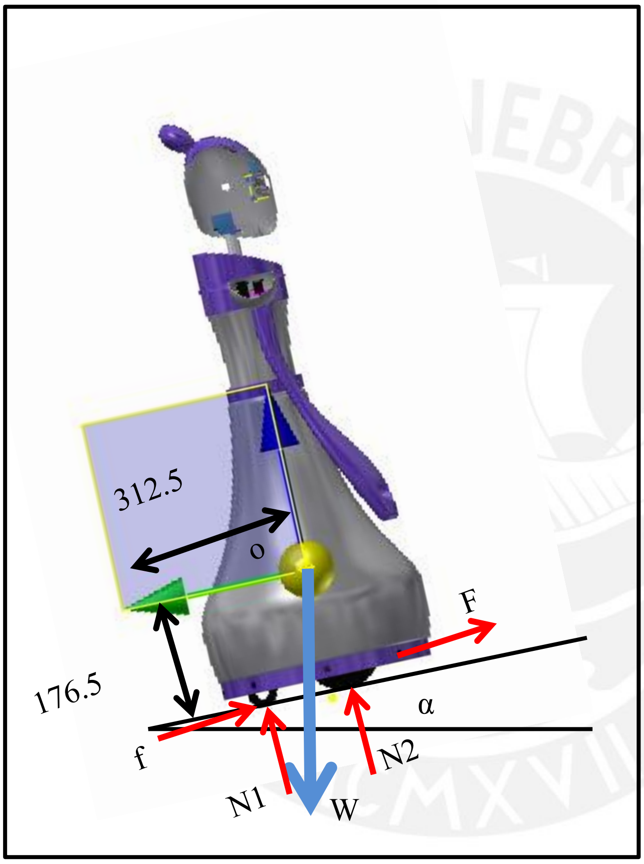
r w=0.33 rad/s y α=0.25 rad/s20.25 rad/s^2

Según la ecuación de Euler- Newton

[0 25]

Los servomotores deberían tener como mínimo un torque de 4.668x10^{-3}

Los servomotores Dynamixel AX-12ª tienen un torque máximo de 1.5Nm a 12V



En la figura 10-1 se muestra el Diagrama de cuerpo libre del sistema, esto para poder hablar el ángulo máximo de elevación de la rampa por la que se movilizará y hallar la potencia necesaria en los motores de la plataforma base que se encargar de dar el movimiento general.

Σ

176.5*senα*w=

312.5*cosα*w

α=60.54° , este es el ángulo máximo para que el sistema no sufra volcadura (N2=0)

Para hallar las potencias requeridas se asume un µ=0.1 y la m =10(hallado del diseño en inventor)

$\sum Fy$ $sin(60\ 54)$

$\sum Fx$ $s(60\ 54)$

Entonces el tor ue necesario en el eje ser a de 0.1983 Nm

Los motores seleccionados cuentan con una potencia el ctrica de 30Wy una velocidad m ima de 1.2m/s entonces la fuerza m ima ue ofrece es de 27 N lo cual supera lo necesario con el m imo ngulo de inclinaci n.

Anexo 11: Análisis de la Estructura Mecánica y Planos de Despiece

La estructura interna del sistema mecatrónico soporta la carcasa, entonces esta debe ser analizada con las fuerzas de compresión que presenta por tener atornillada a ella todos los componentes.
Esta estructura se analiza según Von Mises en Inventor, la simulación muestra que el esfuerzo máximo a soportar será de 0.1905 Mpa.

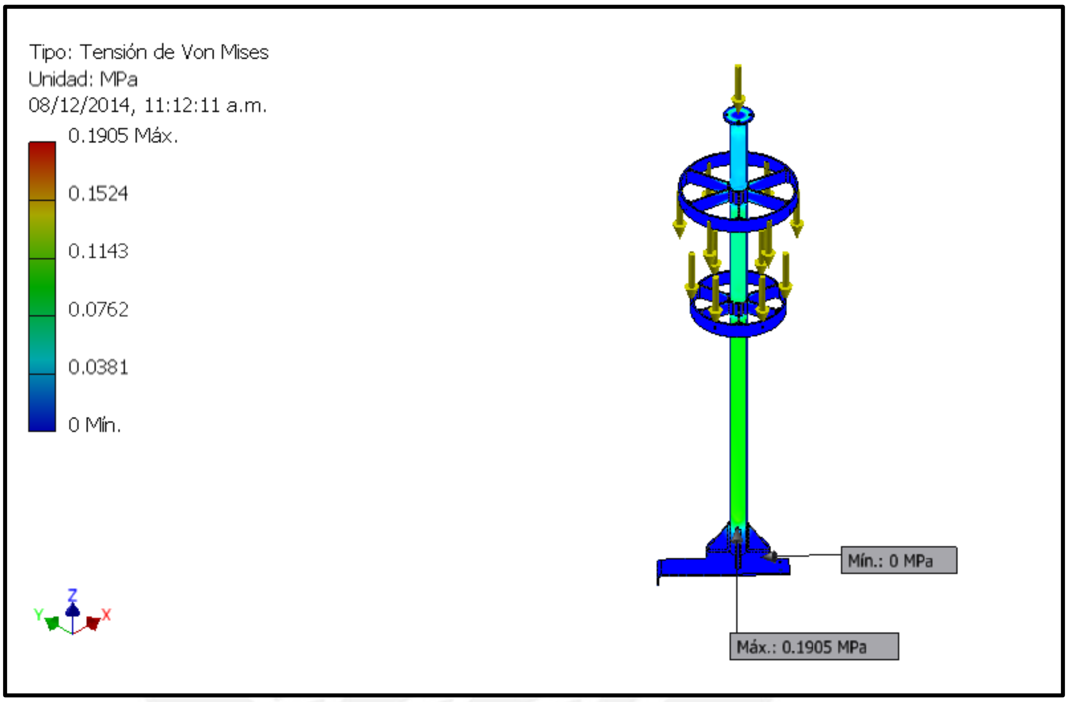
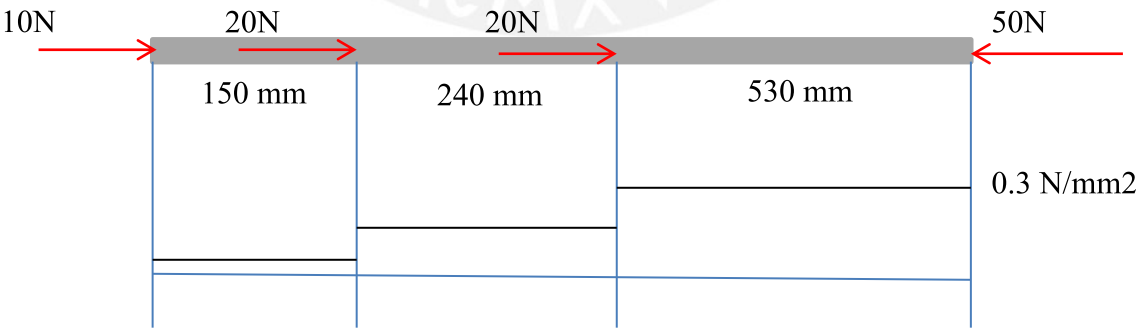


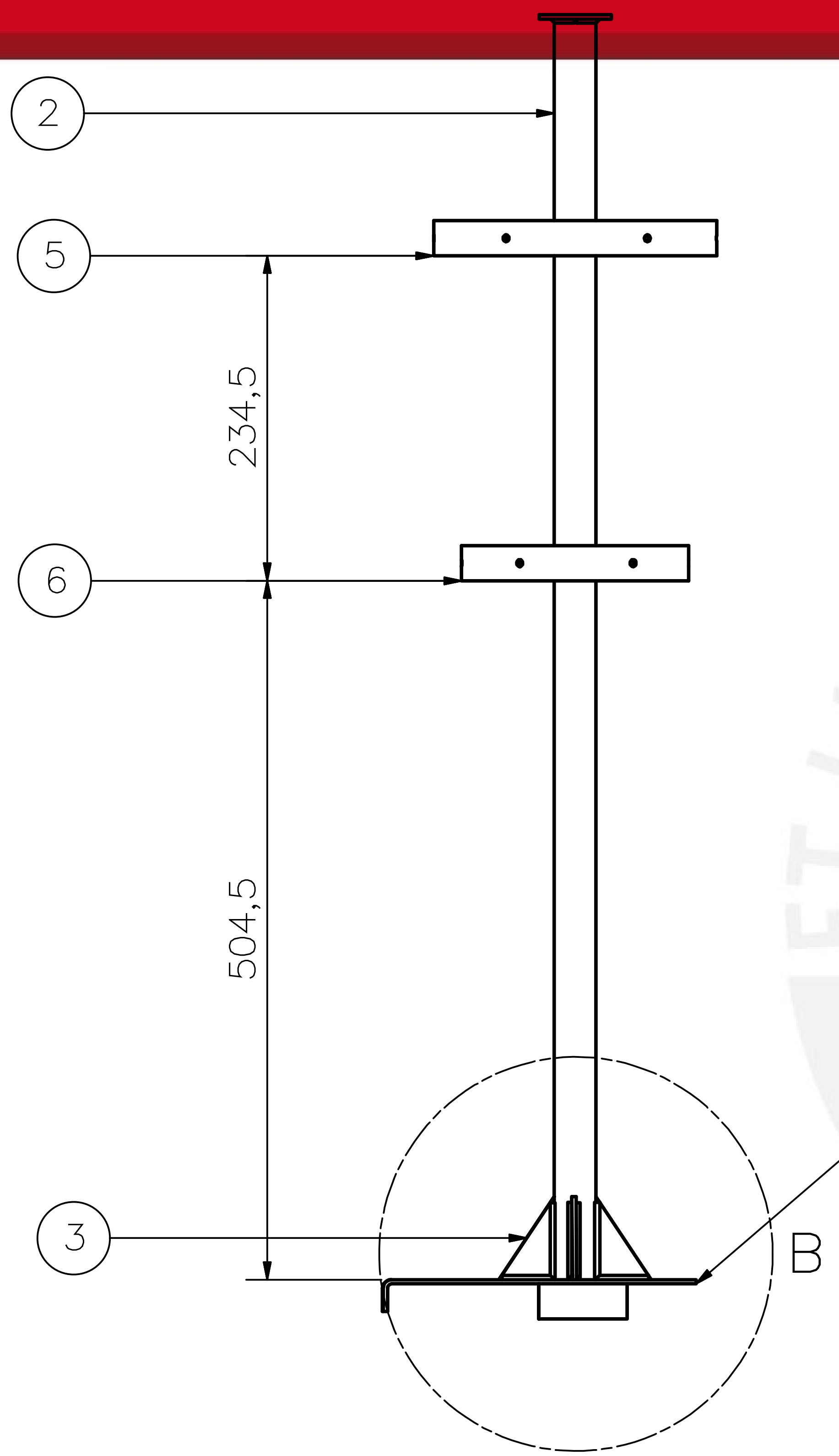
Figura 11-1: Análisis según Von Mises de la estructura interna

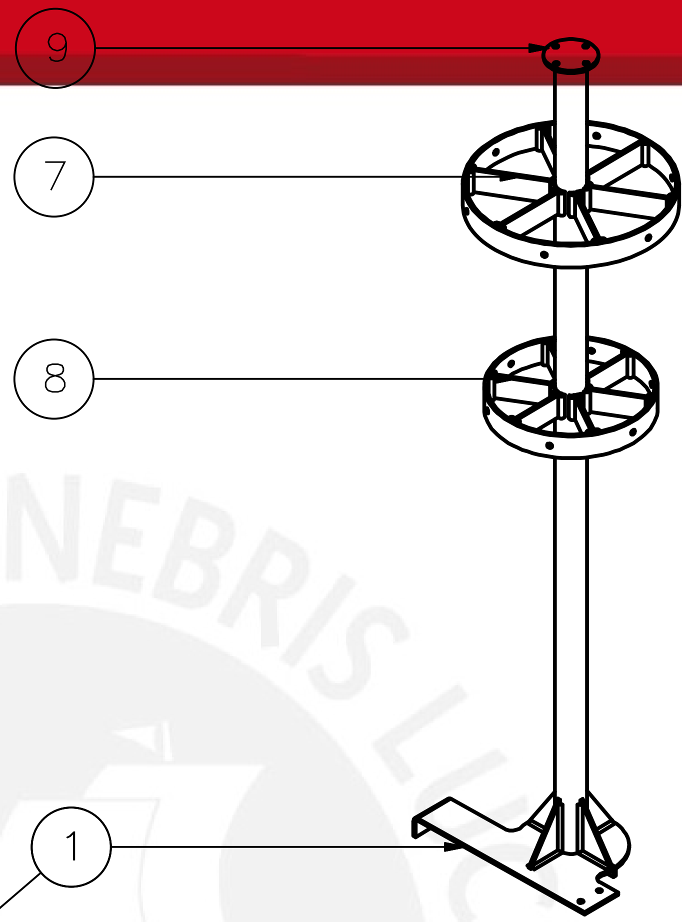


Según Von Mises teórico

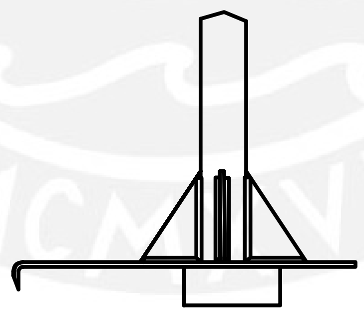
$$\sqrt{(\sigma \quad N) \quad 3(\tau t \quad \tau c)^2} = 0.3 \text{ N/mm2} = 0.3 \text{ MPa}$$

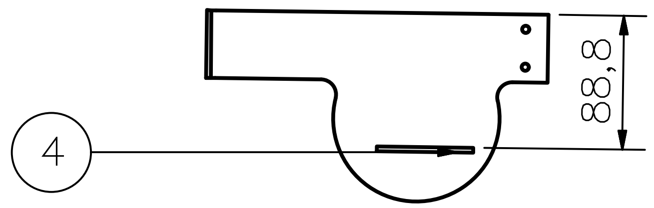
Se tiene que el teórico es mayor que el práctico por lo cual el eje escogido cumple su función.





DETALLE B





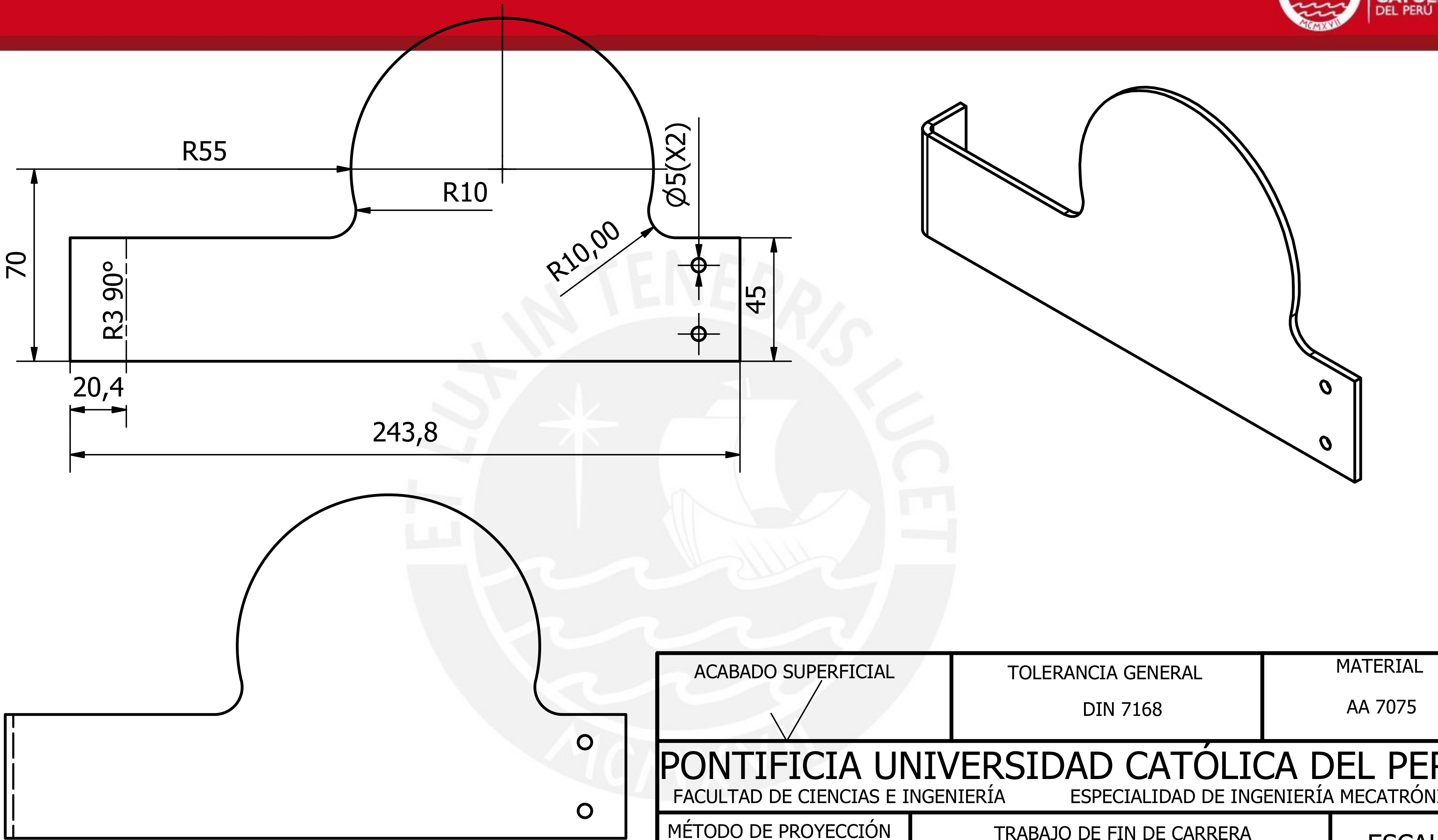
POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES
9	1	Placa Cabeza	DIN7168	AA7075	A4–L10
8	6	Placa Soporte Cintura	DIN7168	AA7075	A4–L09
7	6	Placa Soporte Torso	DIN7168	AA7075	A4–L08
6	1	Soporte Cintura	DIN7168	AA7075	A4–L07
5	1	Soporte Torso	DIN7168	AA7075	A4–L06
4	1	Placa Sujetador Principal	DIN7168	AA7075	A4–L05
3	4	Nervio	DIN7168	AA7075	A4–L04
2	1	Tubo	DIN7168	AA7075	A4–L03
1	1	Sujetador Principal	DIN7168	AA7075	A4–L02

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ TRABAJO FINAL DE CARRERA		
MÉTODO DE PROYECCIÓN	Estructura Central	ESCALA 1:5
20097326	SANCHEZ SIFUENTES, MIDORI	FECHA: 2014.11.10
		LÁMINA: A3–L01

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168

GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5



ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 7168	MATERIAL AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

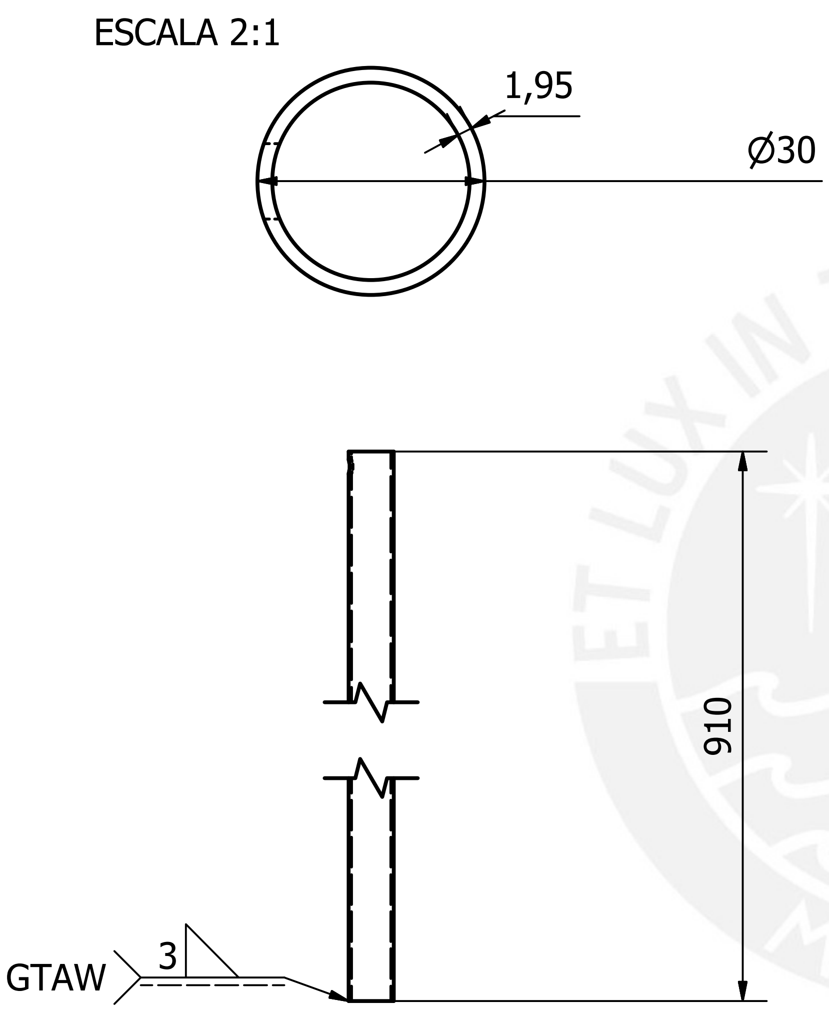
FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO DE FIN DE CARRERA Sujetador Principal	ESCALA 1:2

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L02

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168

GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5



TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
BASTO	±0,15	±0,2	±0,5	±0,8	±1,2

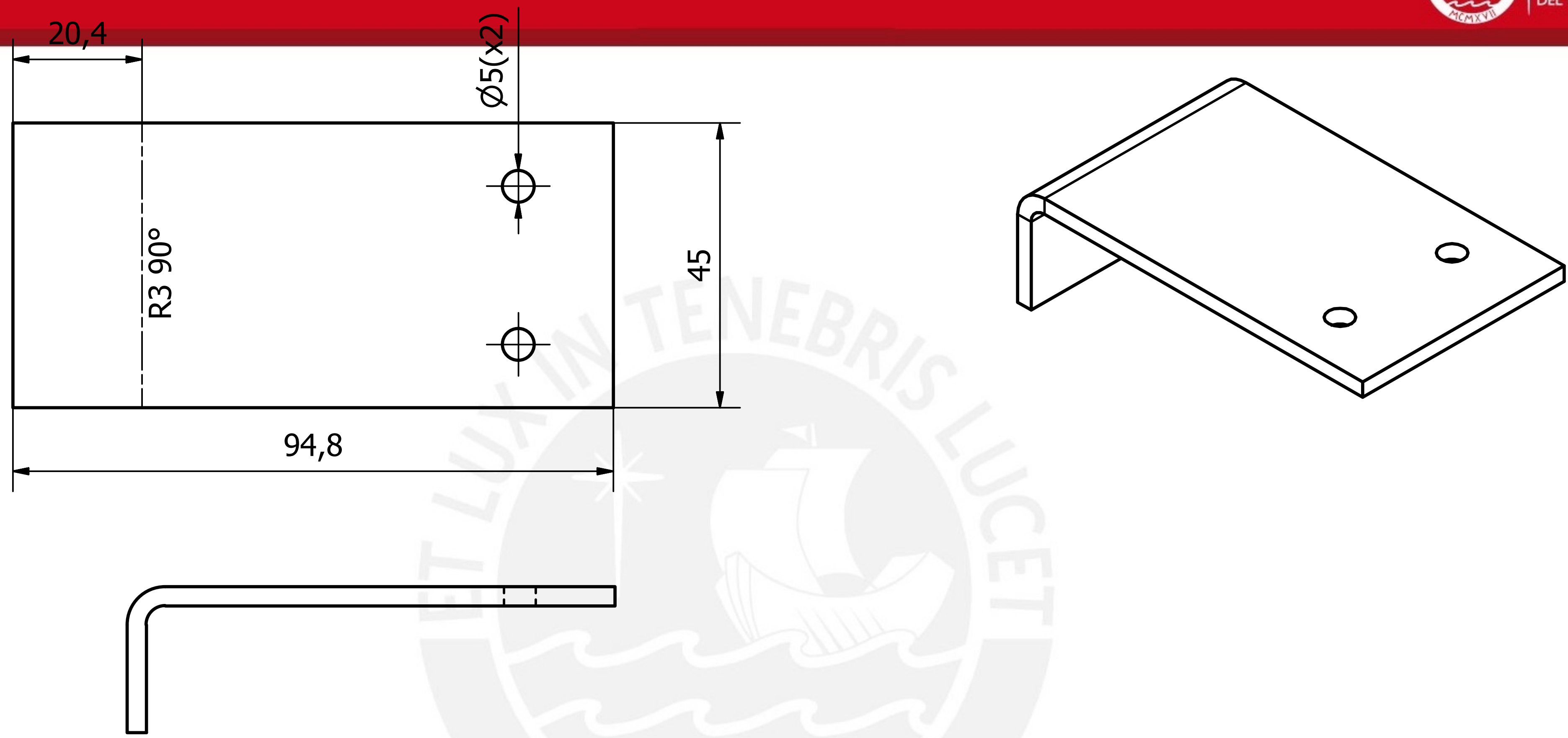
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
	DIN 7168	AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Tubo Guía	1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L03



ESPESOR : 3 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

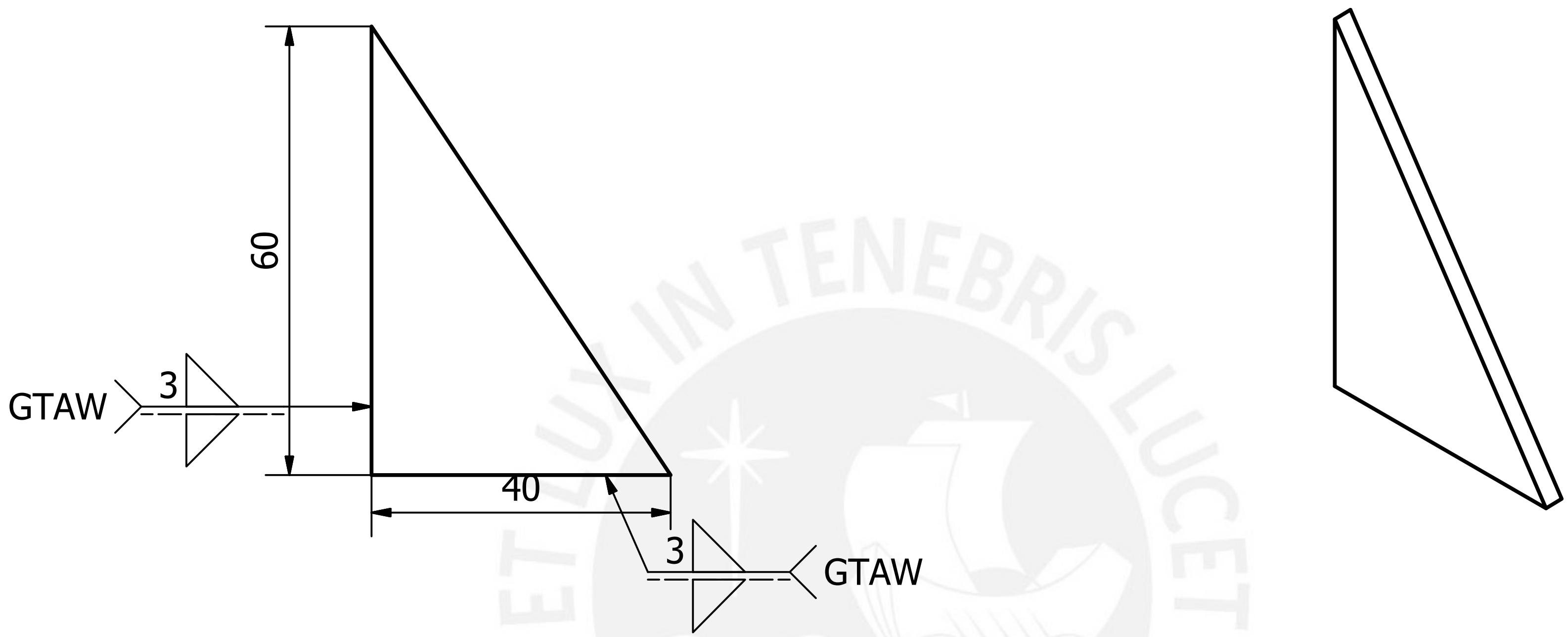
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 7168	MATERIAL AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA Sujetador Derecho	ESCALA 1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L02



ESPESOR : 3 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

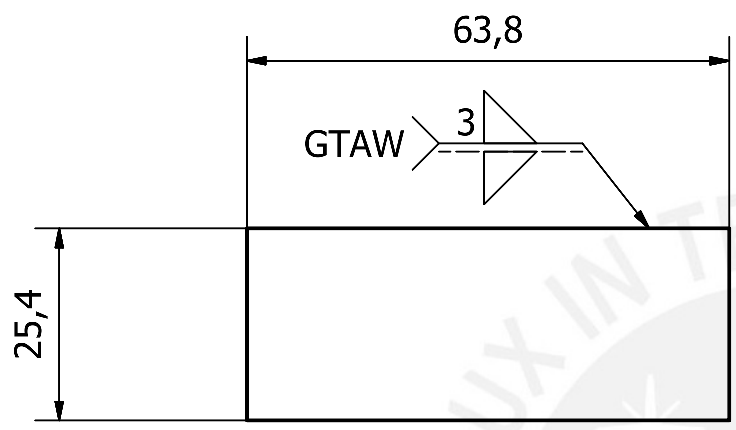
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 7168	MATERIAL AA 7075

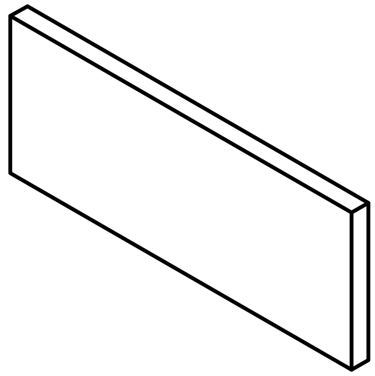
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA Nervio	ESCALA 1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L04





ESPESOR : 3.18 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

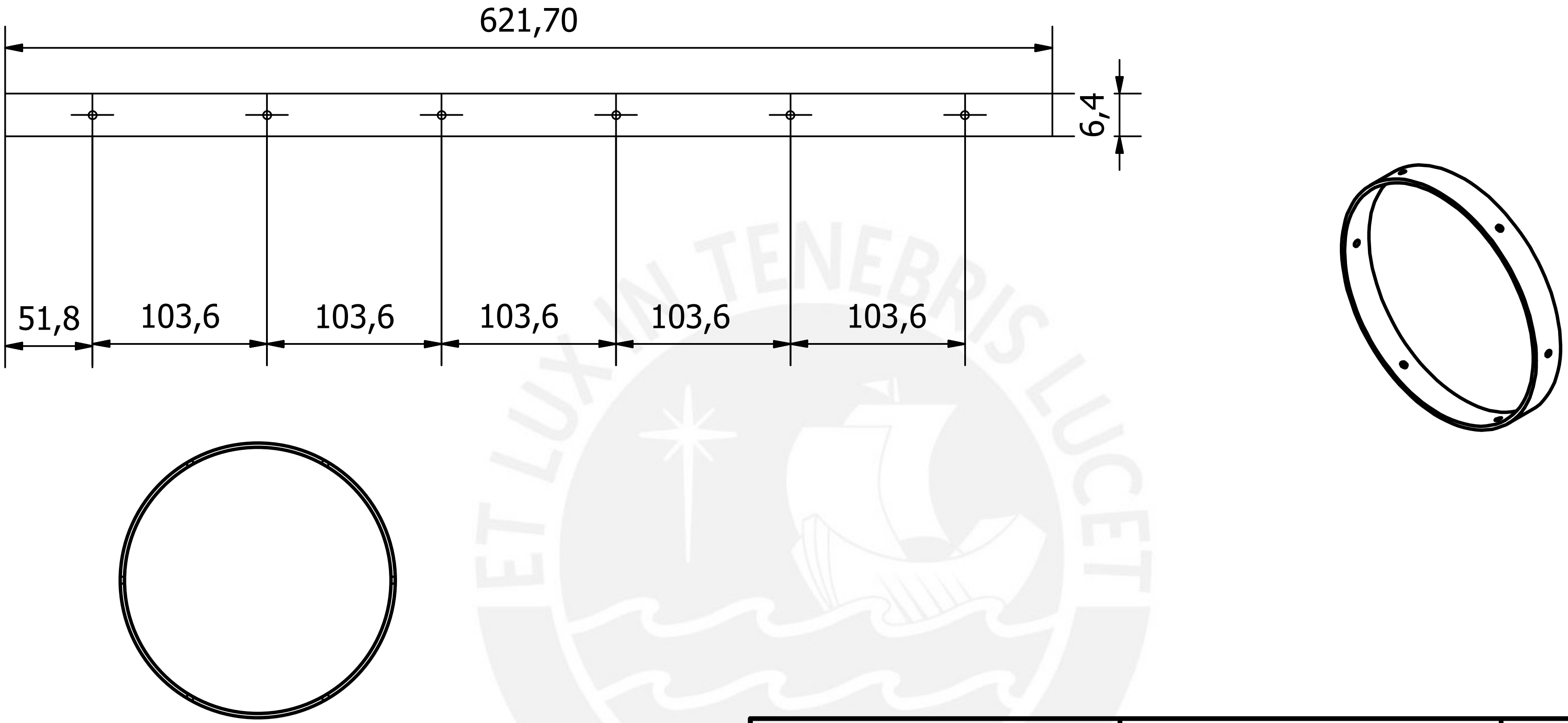
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 7168	MATERIAL AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Placa Soporte Principal	1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L05



ESPESOR : 3.18 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

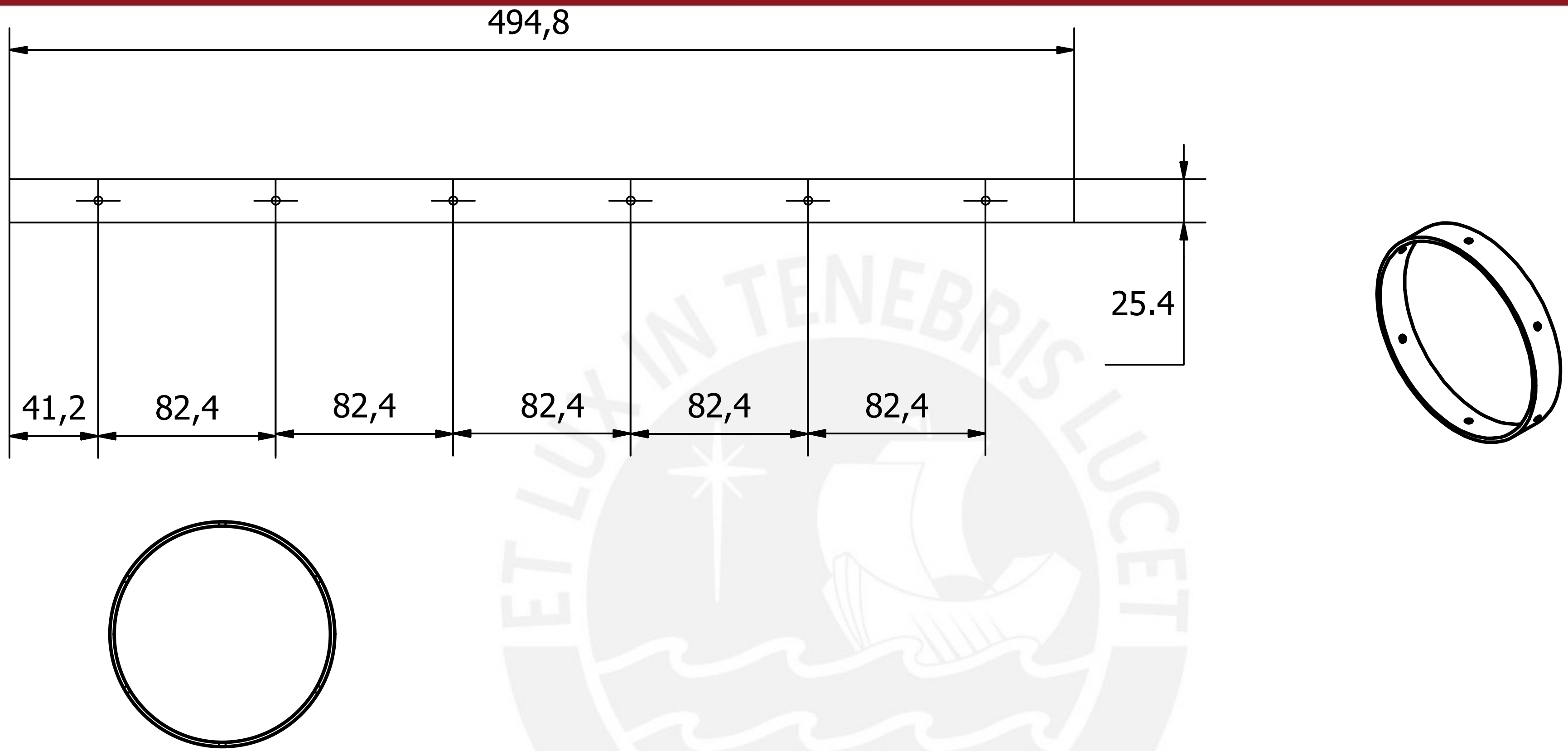
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 7168	MATERIAL AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Soporte Torso	1:5

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L06



ESPESOR : 3.18 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

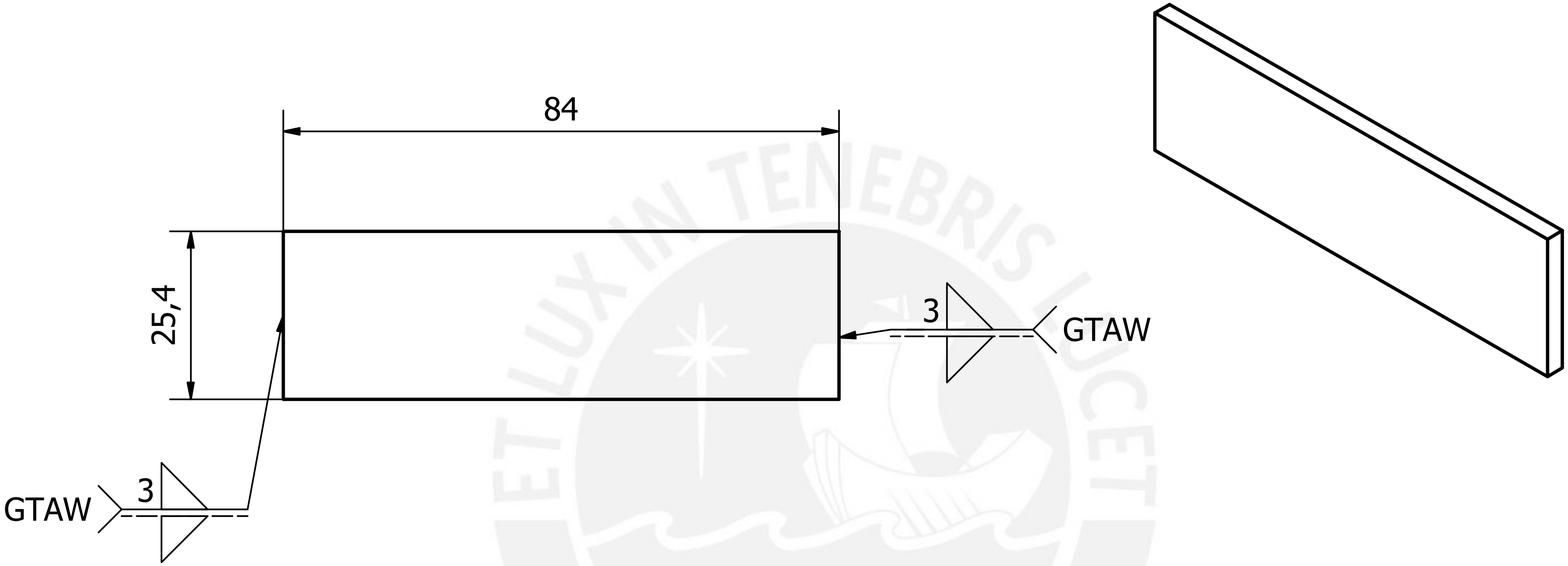
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
√	DIN 7168	AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Soporte Cintura	1:5

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L07



ESPESOR : 3.18 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

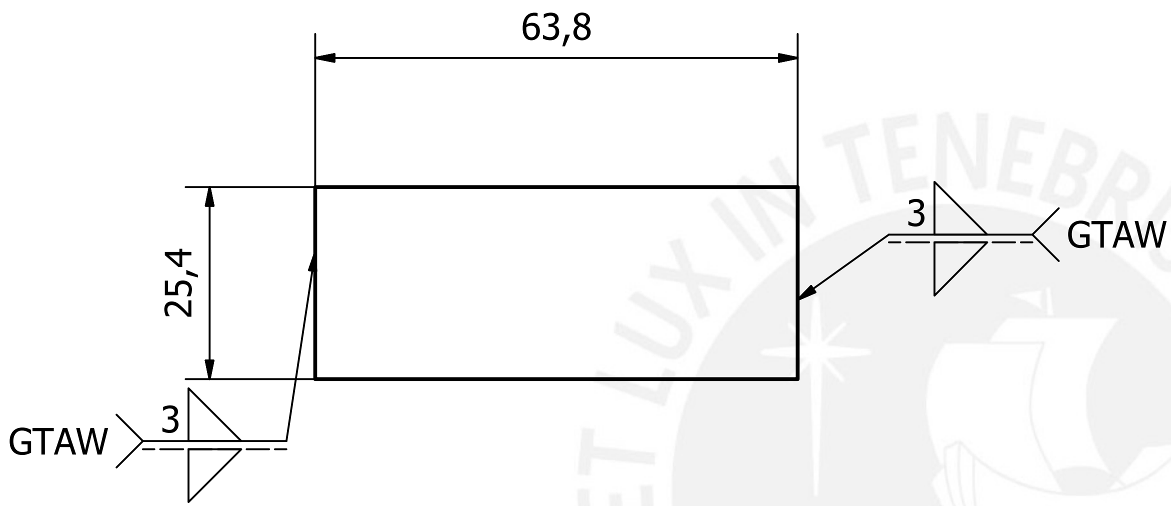
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
	DIN 7168	AA 7075

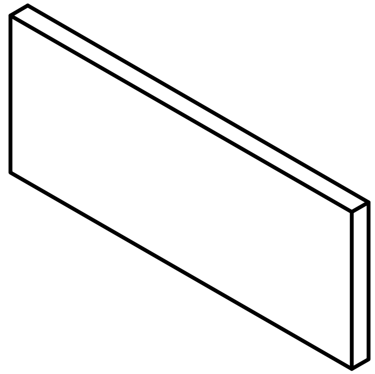
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Placa Soporte Torso	1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L08





ESPESOR : 3.18 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

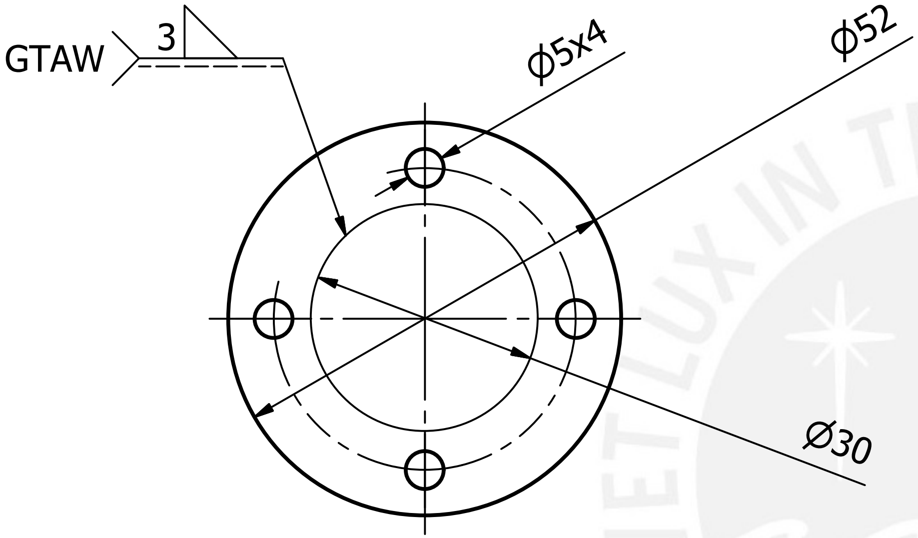
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
	DIN 7168	AA 7075

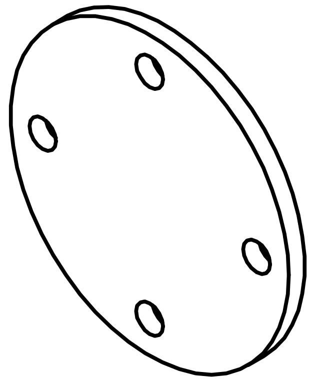
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Placa Soporte Cintura	1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L09





ESPESOR : 3 mm

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

ACABADO SUPERFICIAL

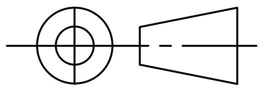
TOLERANCIA GENERAL
DIN 7168

MATERIAL
AA 7075

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN

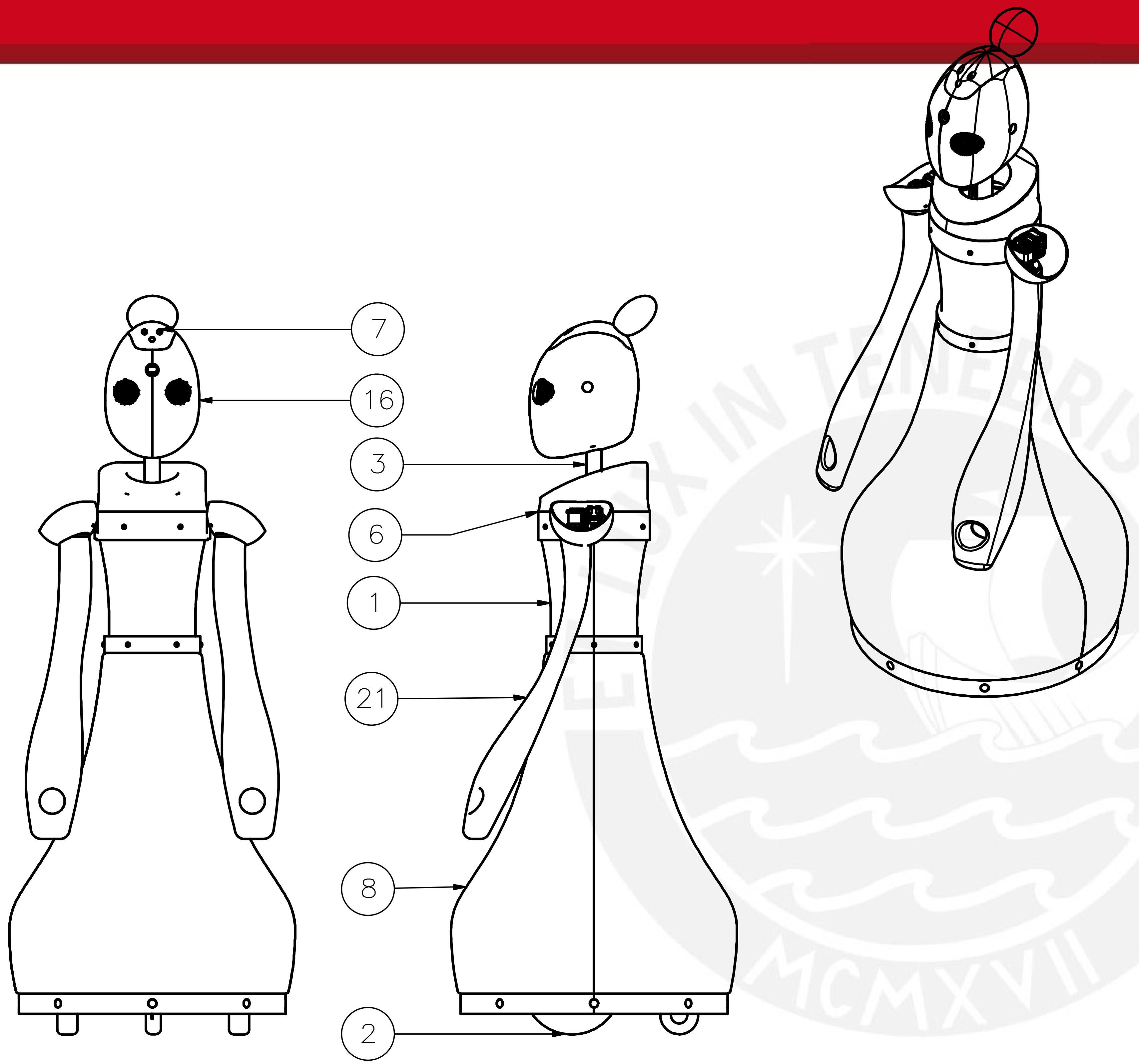


TRABAJO FINAL DE CARRERA

Placa Cabeza

ESCALA
1:1

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: A4-L10



11	2	Tornillo M5x10	DIN 439	Acero	
10	40	Tornillo M2 x 8	DIN 933	Acero	
9	12	Tornillo cabeza redonda M4 x 20	GB/T 818–2000	Acero	
8	1	Falda			
7	1	Cabeza superior		Resina	
6	1	Cuello		Resina	
5	1	Brazo			
4	1	Rostro		Resina	
3	1	Estructura Interna		AA 6061	
2	1	Pioneer P3–dx			
1	1	Torso		Resina	
POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES

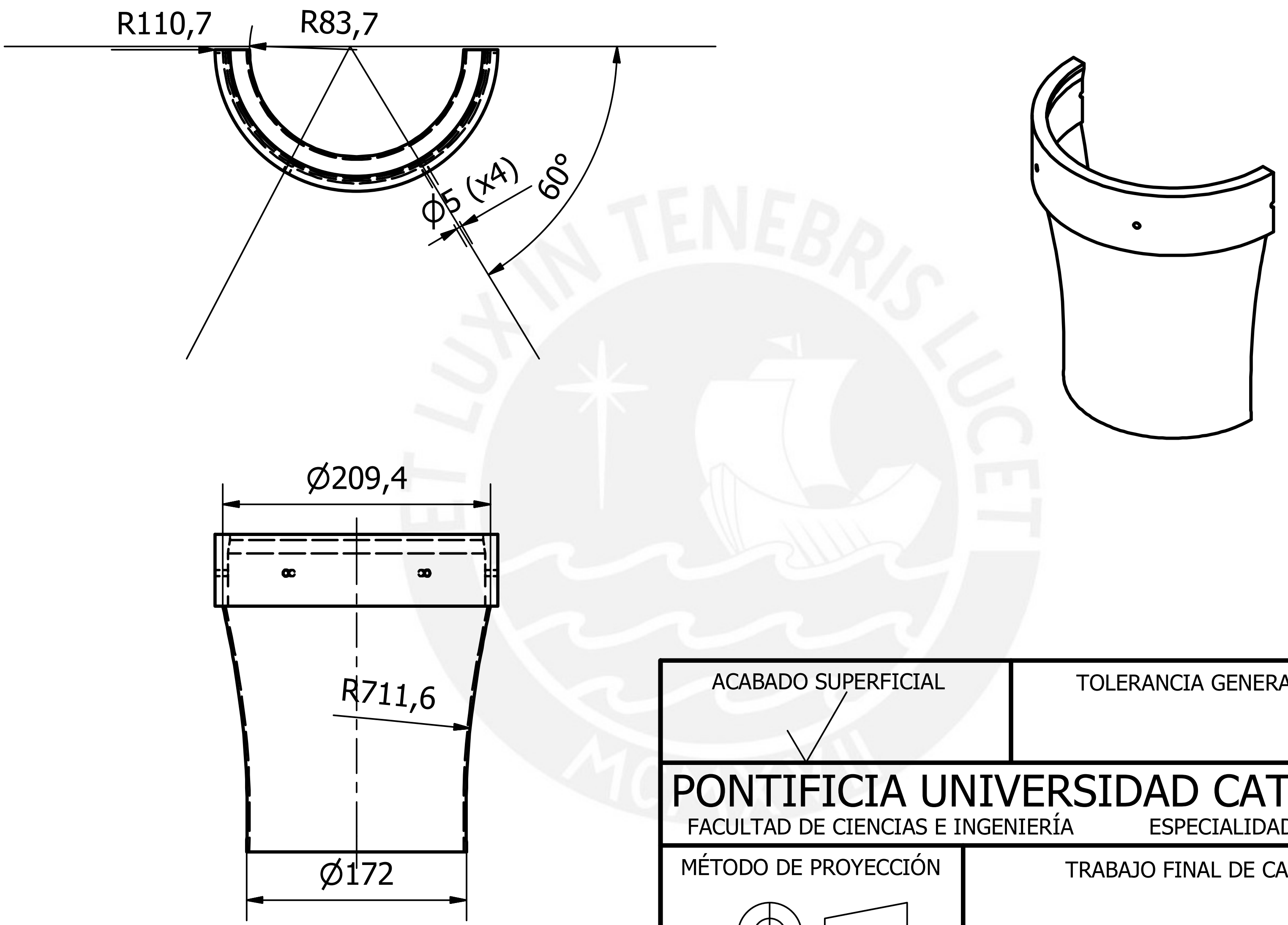
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA 1:5
20097326	SANCHEZ SIFUENTES, MIDORI	FECHA: 2014.11.10
		LÁMINA:

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168

GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA



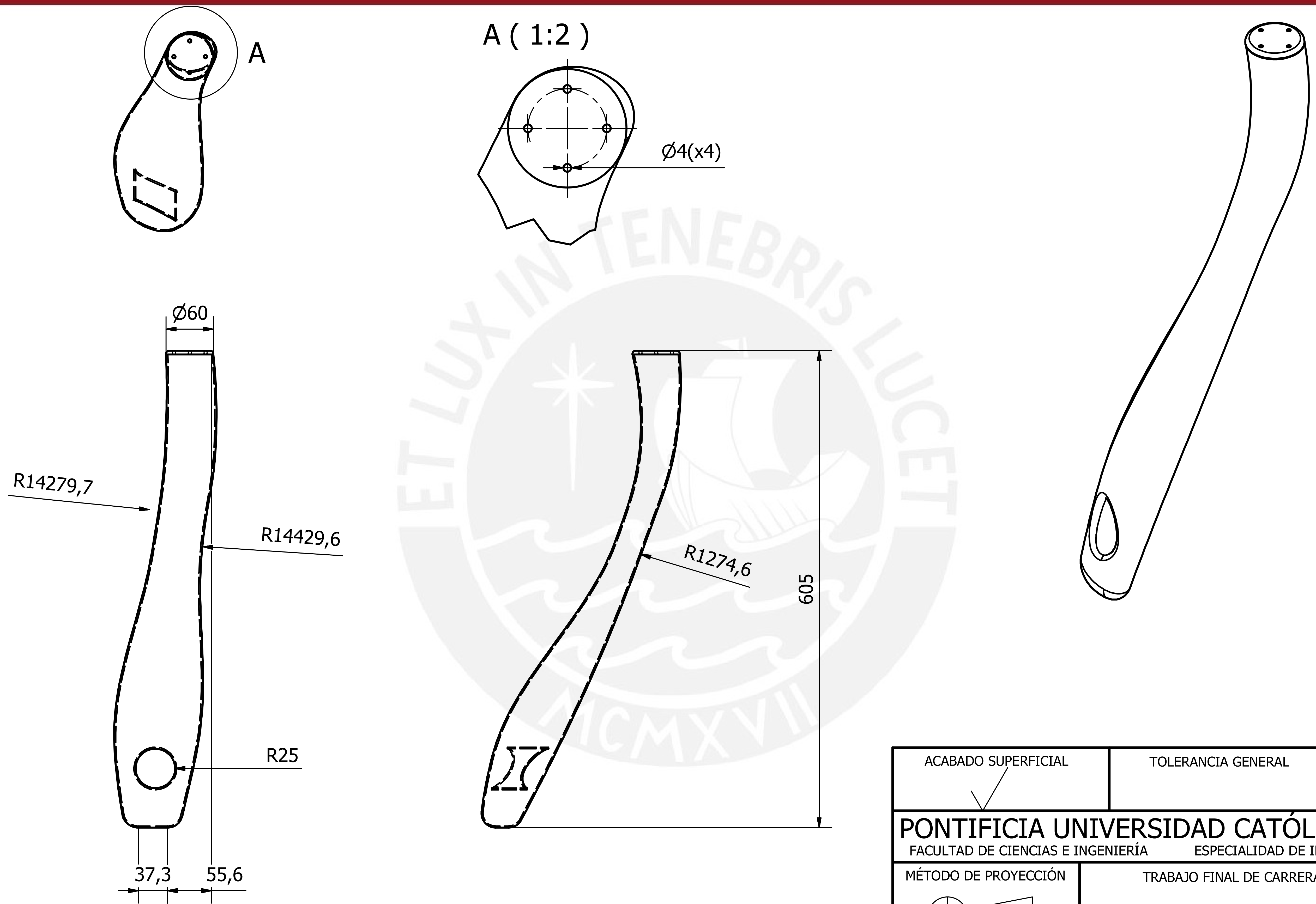
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

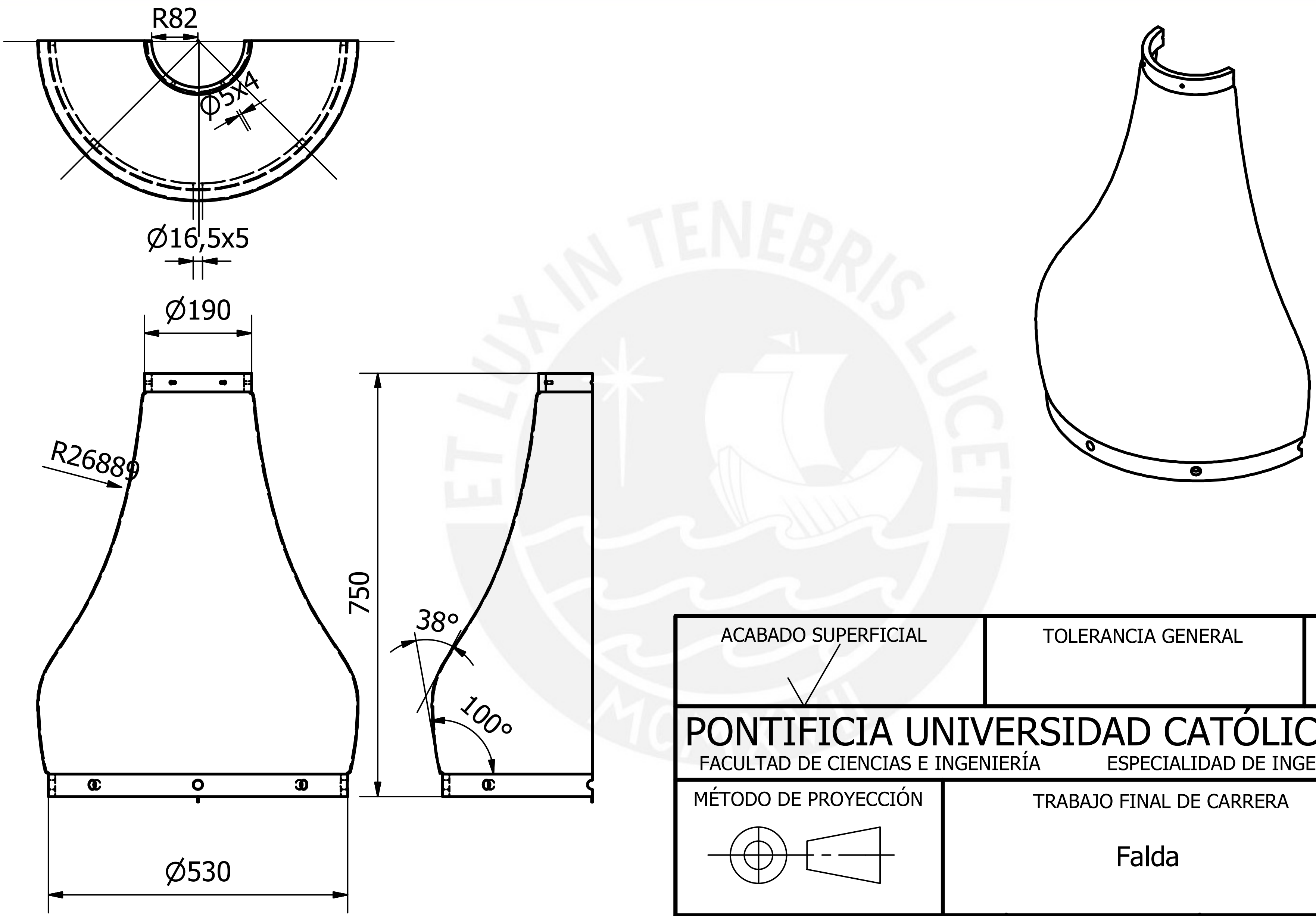
FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Torso	1:5

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: AN-LNN



ACABADO SUPERFICIAL		TOLERANCIA GENERAL		MATERIAL	
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ					
FACULTAD DE CIENCIAS E INGENIERÍA		ESPECIALIDAD DE INGENIERÍA MECATRÓNICA			
MÉTODO DE PROYECCIÓN		TRABAJO FINAL DE CARRERA Brazo Derecho		ESCALA 1:5	
DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07	
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: AN-LNN	

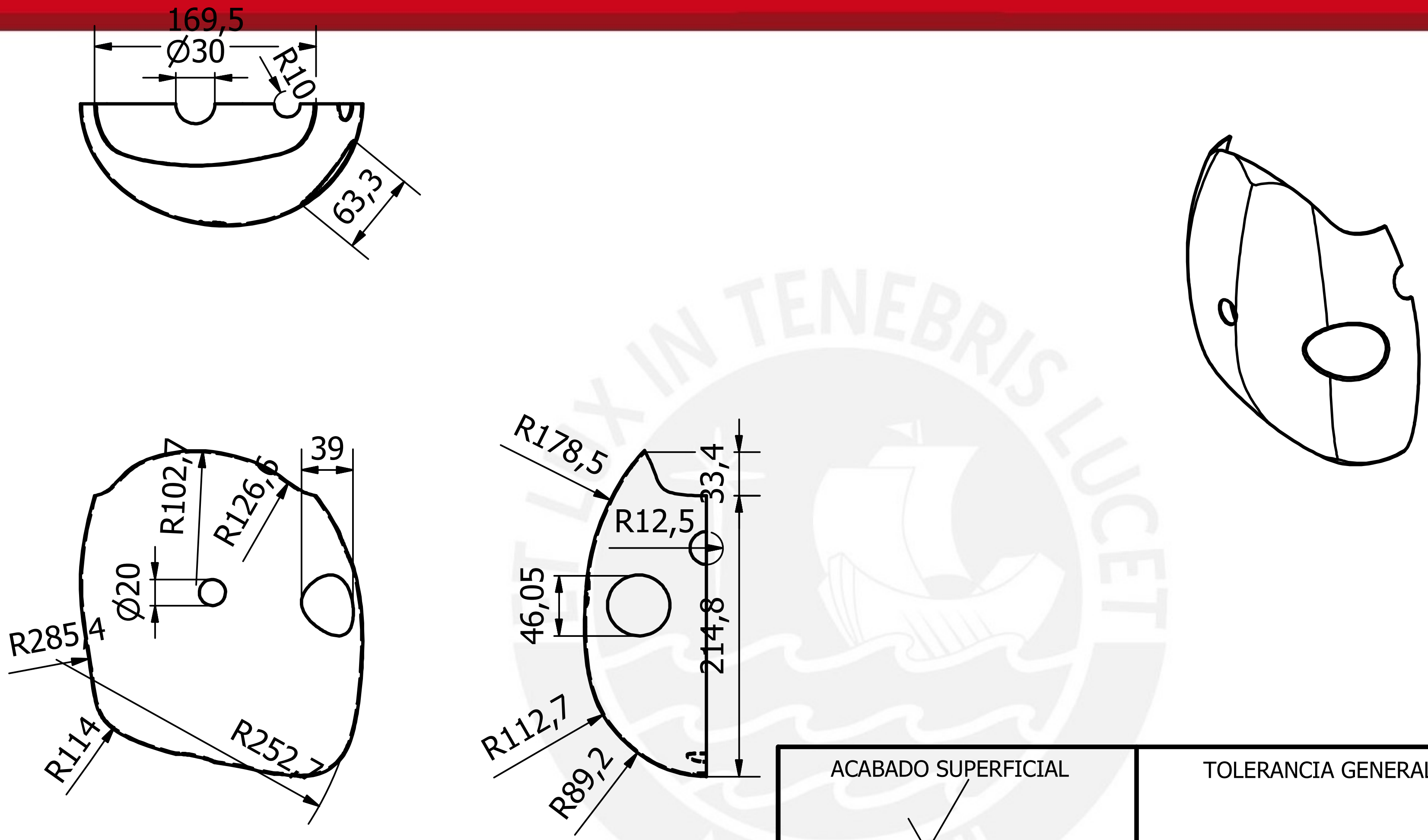


ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA Falda	ESCALA 1:10

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: AN-LNN

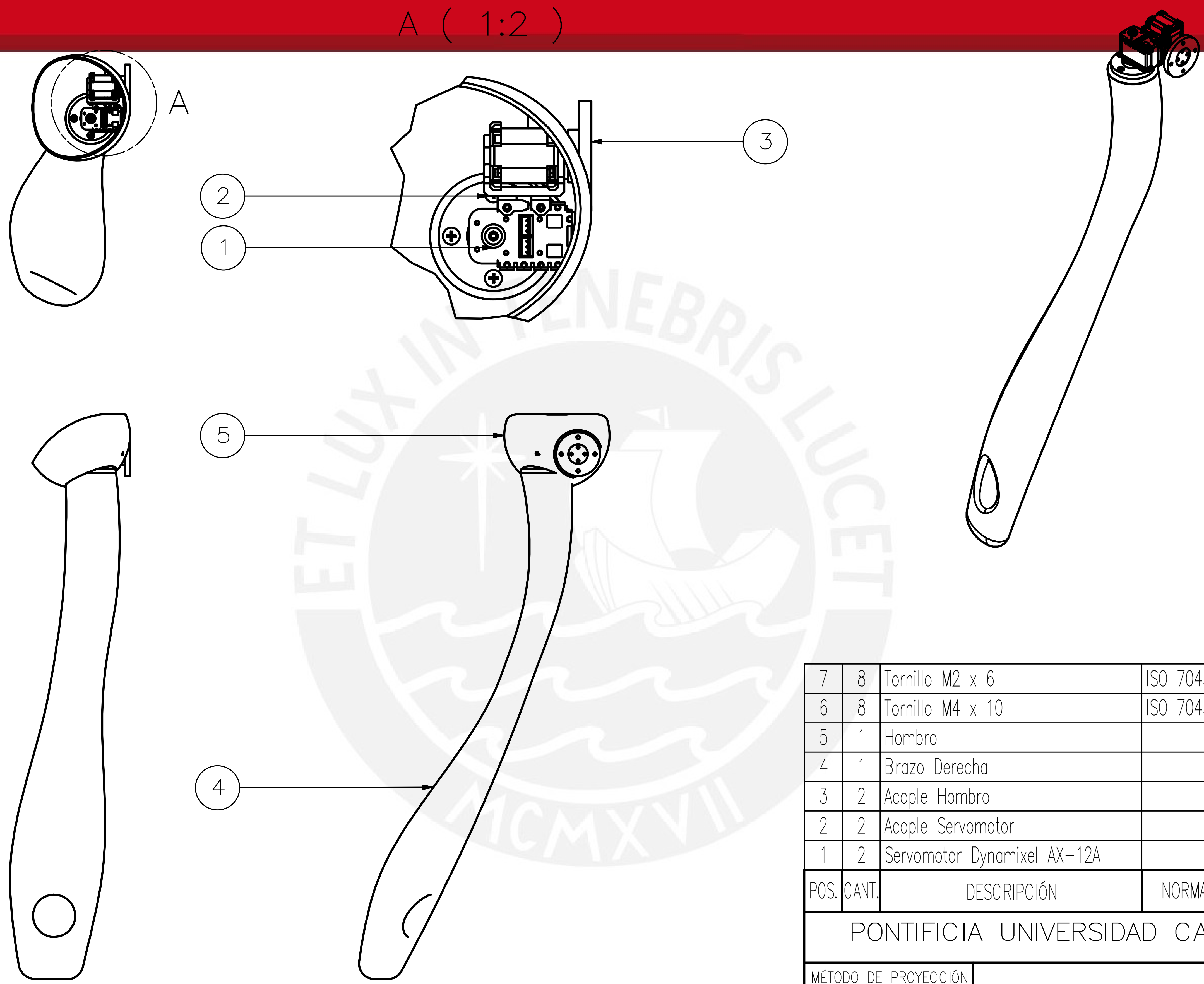


ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Cabeza Mitad	1:5

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: AN-LNN



POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES
7	8	Tornillo M2 x 6	ISO 7045	440C	
6	8	Tornillo M4 x 10	ISO 7045	440C	
5	1	Hombro		Resina	
4	1	Brazo Derecha		Resina	
3	2	Acople Hombro		Plástico ABS	
2	2	Acople Servomotor		Plástico	
1	2	Servomotor Dynamixel AX-12A			

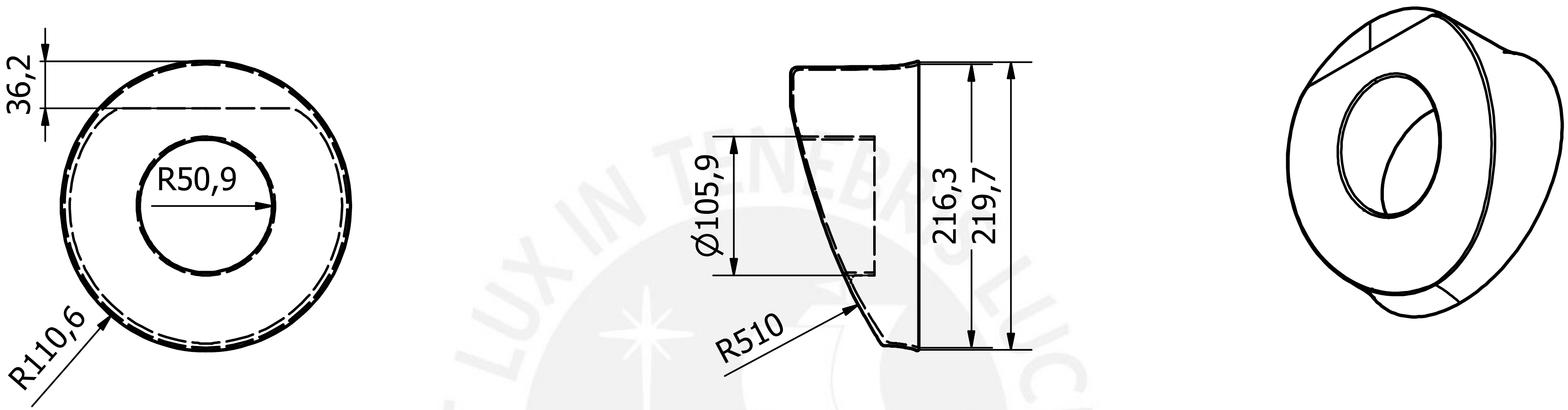
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

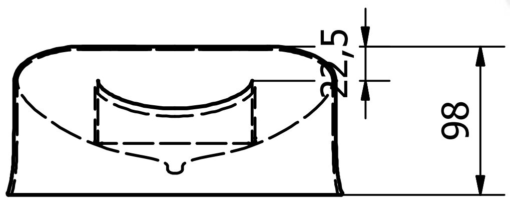
MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA 1:5
20097326	SANCHEZ SIFUENTES, MIDORI	FECHA: 2014.11.10
		LÁMINA:

TOLERANCIAS DIMENSIONALES SEGÚN DIN 7168

GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
MEDIO	±0,1	±0,1	±0,2	±0,3	±0,5

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA





ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL

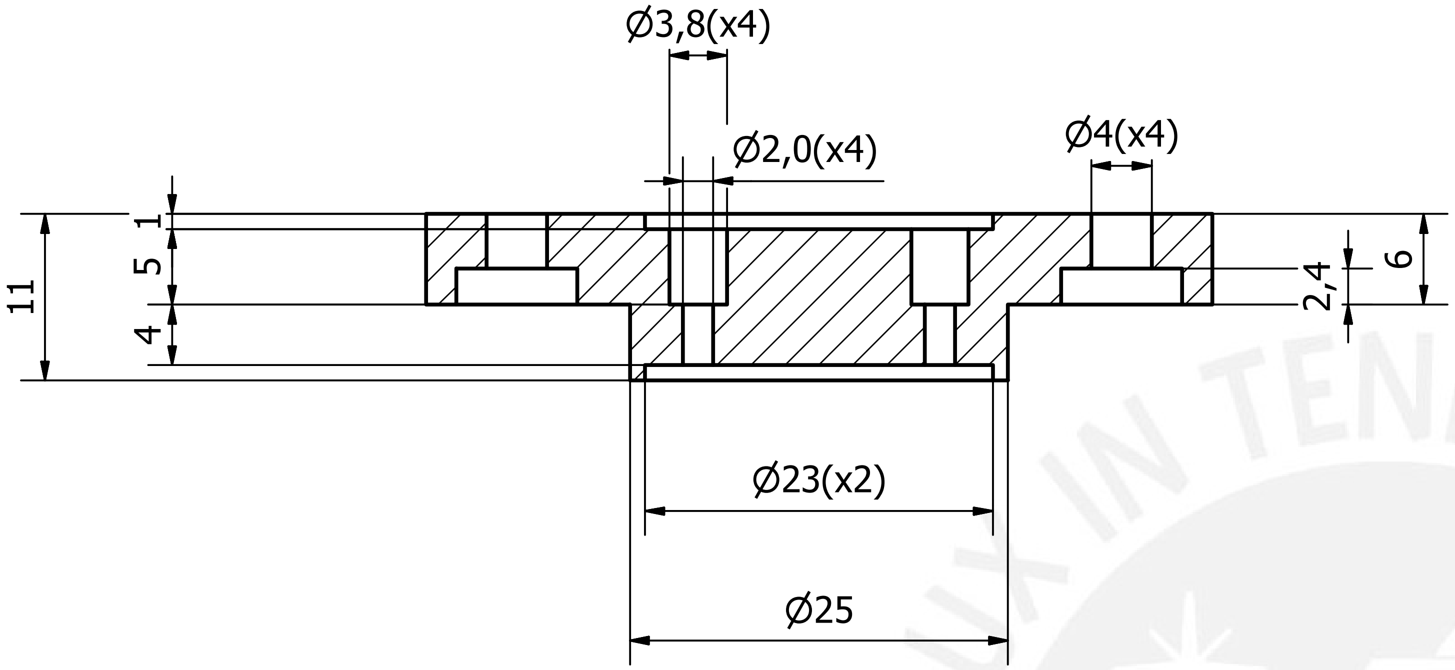
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

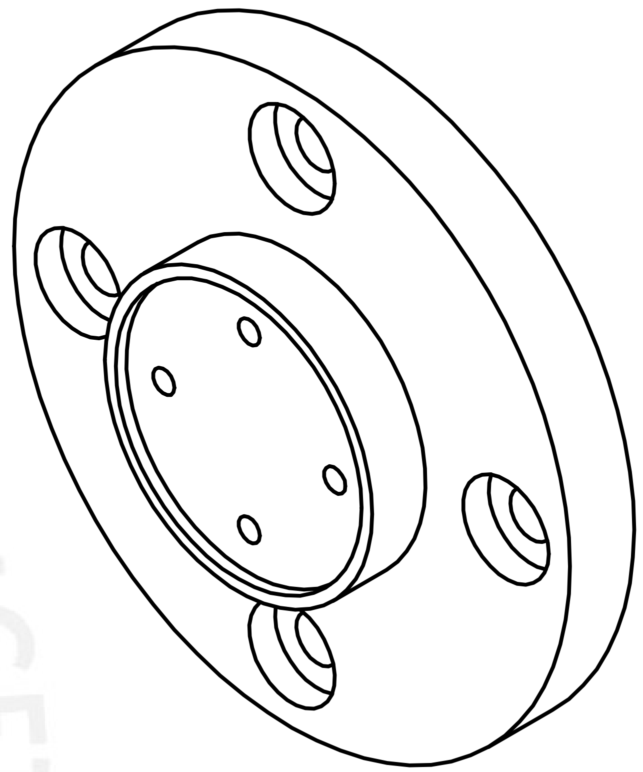
FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

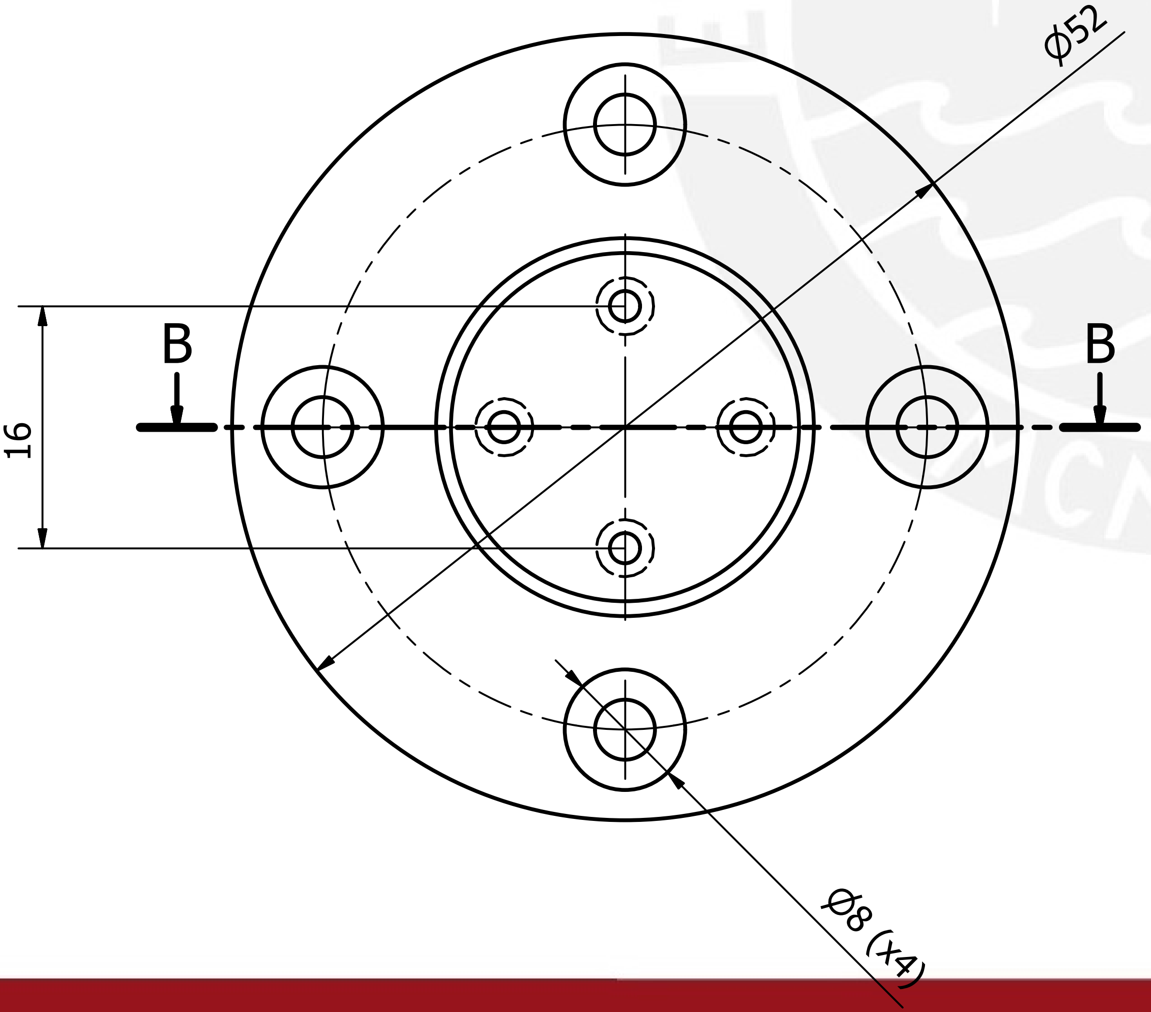
MÉTODO DE PROYECCIÓN	TRABAJO FINAL DE CARRERA	ESCALA
	Cuello	1:5

DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: AN-LNN

B-B (2 : 1)







ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL ABS

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA — ESPECIALIDAD DE INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	PROYECTO DE DISEÑO MECATRÓNICO Acople Hombro	ESCALA 2:1

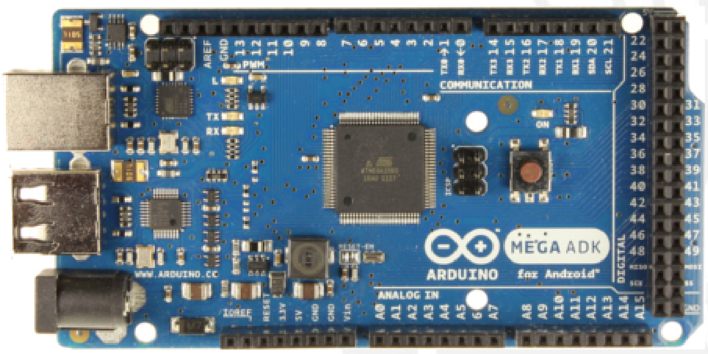
DIBUJADO	SÁNCHEZ MIDORI	REVISADO		FECHA: 2014.10.07
DISEÑADO	SÁNCHEZ MIDORI	APROBADO		LAMINA: AN-LNN

Anexo 12: Arduino MEGA ADK

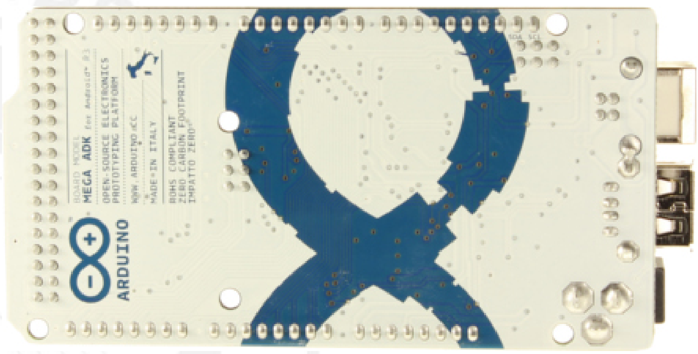
Search the Arduino Website

(redirected from Main.ArduinoBoardADK (http://arduino.cc/en/Main/ArduinoBoardADK))

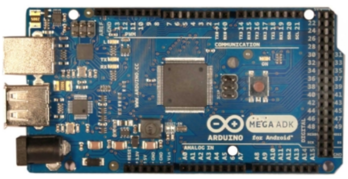
Arduino MEGA ADK



(http://arduino.cc/en/uploads/Main/ArduinoADK_R3_Front.jpg)
Arduino MEGA ADK R3 Front



(http://arduino.cc/en/uploads/Main/ArduinoADK_R3_Back.jpg)
Arduino MEGA ADK R3 Back



(http://arduino.cc/en/uploads/Main/ArduinoADKFront.jpg)
Arduino MEGA ADK Front

(http://arduino.cc/en/uploads/Main/ArduinoADKBack.jpg)
Arduino MEGA ADK Back

Buy From Arduino Store (http://store.arduino.cc/ww/index.php?main_page=product_info&cPath=11_12&products_id=144)

Buy From Distributors (http://arduino.cc/en/Main/Buy)

Overview

The Arduino MEGA ADK is a microcontroller board based on the ATmega2560 (datasheet (http://www.atmel.com/dyn/resources/prod_documents/doc2549.PDF)). It has a USB host interface to connect with Android based phones, based on the MAX3421e IC. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.
The MEGA ADK is based on the Mega 2560 (http://arduino.cc/en/Main/ArduinoBoardMega2560).

Similar to the Mega 2560 and Uno, it features an ATmega8U2 programmed as a USB-to-serial converter.
Revision 2 of the Mega ADK board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode (http://arduino.cc/en/Hacking/DFUProgramming8U2).
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.

For information on using the board with the Android OS, see:
- Google's ADK documentation (http://developer.android.com/guide/topics/usb/adk.html).
- Arduino's ADK documentation (http://arduino.cc/en/Reference/AndroidAccessory).

Arduino Library

The Arduino USB Host Shield can be used with the "USB Host Library for Arduino" hosted by Lauzus from circuits@home (http://www.circuitsathome.com) on GitHub (https://github.com/felis/USB_Host_Shield_2.0) (download (https://github.com/felis/USB_Host_Shield_2.0/archive/master.zip)).

Schematic, Reference Design & Pin Mapping

EAGLE files: Arduino_ADK-Mega_2560-Rev3-reference-design.zip (http://arduino.cc/en/uploads/Main/Arduino_ADK-Mega_2560-Rev3-reference-design.zip)
Schematic: Arduino ADK_Mega_2560-schematic.pdf (http://arduino.cc/en/uploads/Main/ADK_MEGA_2560-Rev2-sch.pdf)
Pin Mapping: PinMap2560 page (http://arduino.cc/en/Hacking/PinMapping2560)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
USB Host Chip	MAX3421E

Power

The Arduino MEGA ADK can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
NB: Because the MEGA ADK is a USB Host, the phone will attempt to draw power from it when it needs to charge. When the ADK is powered over USB, 500mA total is available for the phone and board.The external power regulator can supply up to 1500mA. 750mA is available for the phone and MEGA ADK board. An additional 750mA

is allocated for any actuators and sensors attached to the board. A power supply must be capable of providing 1.5A to use this much current.
The board can operate on an external supply of 5.5 to 16 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The MEGA ADK has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library (http://www.arduino.cc/en/Reference/EEPROM)).

Input and Output

Each of the 50 digital pins on the MEGA ADK can be used as an input or output, using pinMode() (http://arduino.cc/en/Reference/PinMode), digitalWrite() (http://arduino.cc/en/Reference/DigitalWrite), and digitalRead() (http://arduino.cc/en/Reference/DigitalRead) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() (http://arduino.cc/en/Reference/AttachInterrupt) function for details.
- **PWM: 2 to 13 and 44 to 46.** Provide 8-bit PWM output with the analogWrite() (http://arduino.cc/en/Reference/AnalogWrite) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library (http://arduino.cc/en/Reference/SPI). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **USB Host: MAX3421E.** The MAX3421E comunicate with Arduino with the SPI bus. So it uses the following pins:
  - Digital: 7 (RST), 50 (MISO), 51 (MOSI), 52 (SCK).
    NB:Please do not use Digital pin 7 as input or output because is used in the comunication with MAX3421E
  - Non broken out on headers: PJ3 (GP_MAX), PJ6 (INT_MAX), PH7 (SS).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **TWI: 20 (SDA) and 21 (SCL).** Support TWI communication using the Wire library (http://arduino.cc/en/Reference/Wire). Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

The MEGA ADK has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference (http://arduino.cc/en/Reference/AnalogReference)().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino MEGA ADK has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library (http://www.arduino.cc/en/Reference/SoftwareSerial) allows for serial communication on any of the MEGA ADK's digital pins.
The ATmega2560 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus; see the Wire library (http://arduino.cc/en/Reference/Wire) for details. For SPI communication, use the SPI library (http://arduino.cc/en/Reference/SPI).
The USB host interface given by MAX3421E IC allows the Arduino MEGA ADK to connect and interact to any type of device that have a USB port. For example, allows you to interact with many types of phones, controlling Canon cameras, interfacing with keyboard, mouse and games controllers as Wiimote and PS3.

Programming

The Arduino MEGA ADK can be programmed with the Arduino software (download (http://arduino.cc/en/Main/Software)). For details, see the reference (http://arduino.cc/en/Reference/HomePage) and tutorials (http://arduino.cc/en/Tutorial/HomePage).
The ATmega2560 on the Arduino MEGA ADK comes preburned with a bootloader (http://arduino.cc/en/Tutorial/Bootloader) (the same on Mega 2560) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500v2 protocol (reference (http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf), C header files (http://www.atmel.com/dyn/resources/prod_documents/avr061.zip)).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP (http://arduino.cc/en/Main/ArduinoISP) or similar; see these instructions (http://arduino.cc/en/Hacking/Programmer) for details.
The ATmega8U2 firmware source code is available in the Arduino repository (http://github.com/arduino/Arduino/tree/master/hardware/arduino/firmwares/). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3886) (Windows) or the DFU programmer (http://dfu-programmer.sourceforge.net/) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial (http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1285962838) for more information.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino MEGA ADK is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the MEGA ADK is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the MEGA ADK. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The MEGA ADK contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread (http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1213719666/all) for details.

USB Overcurrent Protection

The Arduino MEGA ADK has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the MEGA ADK PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.
The MEGA ADK is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the MEGA ADK and Duemilanove / Diecimila. Please note that I^2C is not located on the same pins on the MEGA ADK (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

Drivers & Setup

With this board you need to change the boards.txt file in your Arduino directory (find it in: "Arduino-00xx > hardware > arduino"):

Arduino 0022 or older	boards.txt (http://arduino.cc/en/uploads/Main/boards.txt.zip)
Arduino 1.0 Beta	boards1.0.txt (http://arduino.cc/en/uploads/Main/boards1.0.txt.zip)

[To be downloaded, this files are compressed into a zip archive, so you need to unzip them into the directory described above.]
Windows users in order to get working the board need a .inf file for this specific product: Arduino_ADK.zip (http://arduino.cc/en/uploads/Main/Arduino_ADK.zip)
For installation follow the same procedure on how install an UNO board on your computer (http://arduino.cc/en/Guide/Windows#toc4).

Share

NEWSLETTER

Enter your email to sign up

 Copyright Notice (http://arduino.cc/en/Main/CopyrightNotice) Contact us (http://arduino.cc/en/Main/ContactUs)

(https://twitter.com/arduino) (http://www.facebook.com/official.arduino)

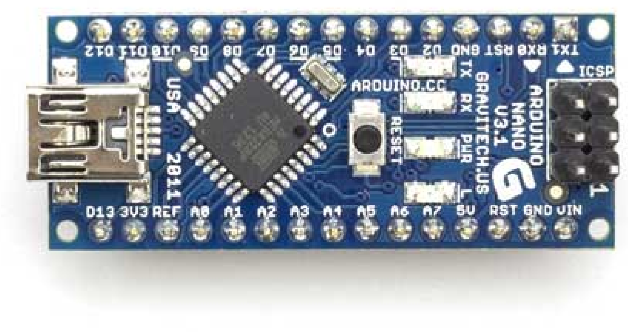
(https://plus.google.com/+Arduino) (http://www.flickr.com/photos/arduino_cc)

(http://youtube.com/arduinoteam)

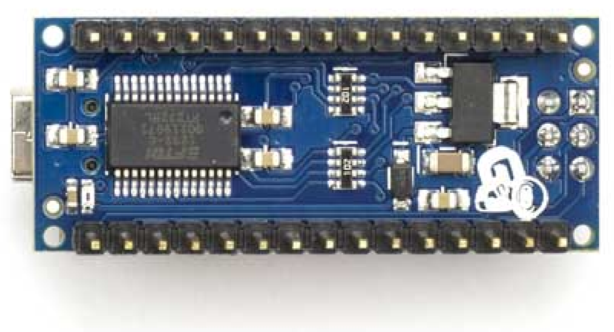
Anexo 13: Arduino Nano

Search the Arduino Website

Arduino Nano



(http://arduino.cc/en/uploads/Main/ArduinoNanoFront_3_lg.jpg)
Arduino Nano Front



(http://arduino.cc/en/uploads/Main/ArduinoNanoBack_3_lg.jpg)
Arduino Nano Rear

Buy From Arduino Store (http://store.arduino.cc/eu/index.php?main_page=product_info&cPath=11&products_id=77)

Buy From Distributors (http://arduino.cc/en/Main/Buy)

Overview

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

Schematic and Design

Arduino Nano 3.0 (ATmega328): schematic (http://arduino.cc/en/uploads/Main/ArduinoNano30Schematic.pdf), Eagle files (http://arduino.cc/en/uploads/Main/ArduinoNano30Eagle.zip).
Arduino Nano 2.3 (ATmega168): manual (http://arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf) (pdf), Eagle files (http://gravitech.us/Arduino/Arduino_Nano_V2_3_Eagle.zip). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Specifications:

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V

Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

Power:

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.
The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library (http://www.arduino.cc/en/Reference/EEPROM)); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode() (http://arduino.cc/en/Reference/PinMode), digitalWrite() (http://arduino.cc/en/Reference/DigitalWrite), and digitalRead() (http://arduino.cc/en/Reference/DigitalRead) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() (http://arduino.cc/en/Reference/AttachInterrupt) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() (http://arduino.cc/en/Reference/AnalogWrite) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() (http://arduino.cc/en/Reference/AnalogReference) function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

- **I^2C: 4 (SDA) and 5 (SCL).** Support I^2C (TWI) communication using the Wire library (http://wiring.org.co/reference/libraries/Wire/index.html) (documentation on the Wiring website).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference (http://arduino.cc/en/Reference/AnalogReference)().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega168 ports (http://arduino.cc/en/Hacking/PinMapping168).

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (http://www.ftdichip.com/Drivers/VCP.htm) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library (http://www.arduino.cc/en/Reference/SoftwareSerial) allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation (http://arduino.cc/en/Reference/Wire) for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software (download (http://arduino.cc/en/Main/Software)). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference (http://arduino.cc/en/Reference/HomePage) and tutorials (http://arduino.cc/en/Tutorial/HomePage).

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a bootloader (http://arduino.cc/en/Tutorial/Bootloader) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference (http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf), C header files (http://www.atmel.com/dyn/resources/prod_documents/avr061.zip)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP (http://arduino.cc/en/Main/ArduinoISP) or similar; see these instructions (http://arduino.cc/en/Hacking/Programmer) for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

(http://youtube.com/arduinoteam)

Anexo 14: BeagleBone Black, tarjeta Level Shifter y Módulo WiFi

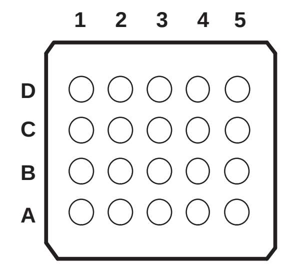
8-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR WITH AUTO-DIRECTION SENSING AND ±15-kV ESD PROTECTION

Check for Samples: TXB0108

FEATURES

- **1.2 V to 3.6 V on A Port and 1.65 to 5.5 V on B Port ($V_{CCA} \le V_{CCB}$)**
- **V_{CC} Isolation Feature – If Either V_{CC} Input Is at GND, All Outputs Are in the High-Impedance State**
- **OE Input Circuit Referenced to V_{CCA}**
- **Low Power Consumption, 4-µA Max I_{CC}**
- **I_{off} Supports Partial-Power-Down Mode Operation**
- **Latch-Up Performance Exceeds 100 mA Per JESD 78, Class II**
- **ESD Protection Exceeds JESD 22**
  - **A Port**
    - **2000-V Human-Body Model (A114-B)**
    - **1000-V Charged-Device Model (C101)**
  - **B Port**
    - **±15-kV Human-Body Model (A114-B)**
    - **1000-V Charged-Device Model (C101)**

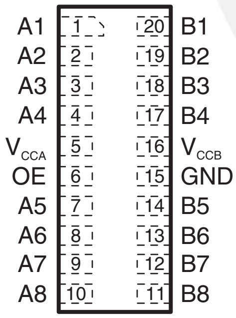
GXY OR ZXY PACKAGE (BOTTOM VIEW)



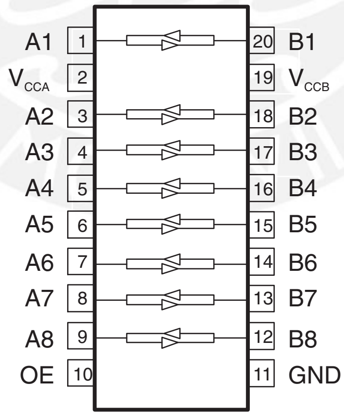
TERMINAL ASSIGNMENTS (20-Ball GXY/ZXY Package)

	1	2	3	4	5
D	V_{CCB}	B2	B4	B6	B8
C	B1	B3	B5	B7	GND
B	A1	A3	A5	A7	OE
A	V_{CCA}	A2	A4	A6	A8

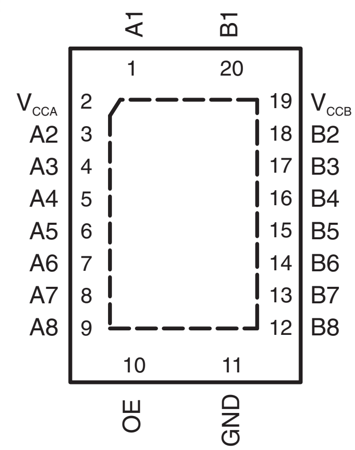
DQS PACKAGE (TOP VIEW)



PW PACKAGE (TOP VIEW)



RGY PACKAGE (TOP VIEW)



DESCRIPTION/ORDERING INFORMATION

This 8-bit noninverting translator uses two separate configurable power-supply rails. The A port is designed to track V_{CCA}. V_{CCA} accepts any supply voltage from 1.2 V to 3.6 V. The B port is designed to track V_{CCB}. V_{CCB} accepts any supply voltage from 1.65 V to 5.5 V. This allows for universal low-voltage bidirectional translation between any of the 1.2-V, 1.5-V, 1.8-V, 2.5-V, 3.3-V, and 5-V voltage nodes. V_{CCA} should not exceed V_{CCB}.

When the output-enable (OE) input is low, all outputs are placed in the high-impedance state.

The TXB0101 is designed so that the OE input circuit is supplied by V_{CCA}.

This device is fully specified for partial-power-down applications using I_{off}. The I_{off} circuitry disables the outputs, preventing damaging current backflow through the device when it is powered down.

To ensure the high-impedance state during power up or power down, OE should be tied to GND through a pulldown resistor; the minimum value of the resistor is determined by the current-sourcing capability of the driver.

Table 1. ORDERING INFORMATION(1)

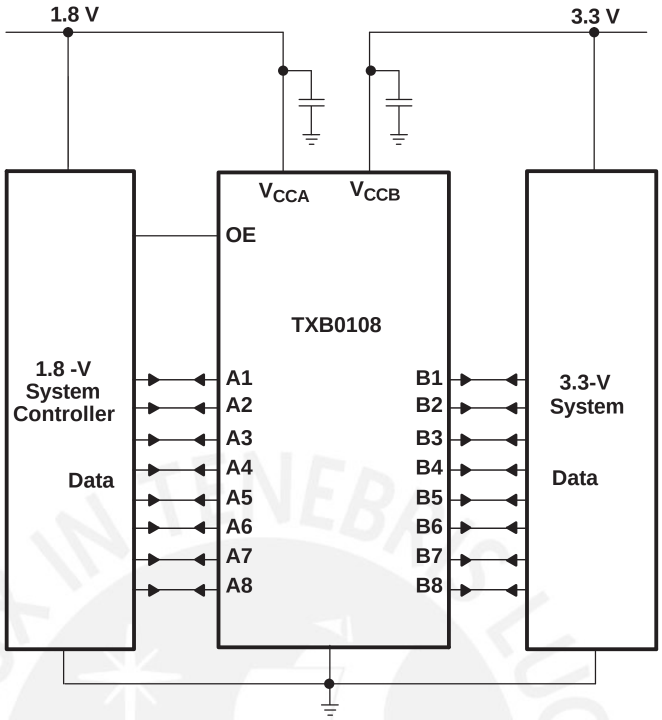
T_A	PACKAGE(2)		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	QFN – RGY	Reel of 1000	TXB0108RGYR	YE08
	SON – DQS	Reel of 2000	TXB0108DQSR	5MR
	TSSOP – PW	Reel of 2000	TXB0108PWR	YE08
	VFBGA – GXY	Reel of 2500	TXB0108GXYR	YE08
	VFBGA – ZXY (Pb-free)	Reel of 2500	TXB0108ZXYR	YE08

(1) For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.
(2) Package drawings, thermal data, and symbolization are available at www.ti.com/packaging.

PIN DESCRIPTION

NO. (DQS, PW, RGY)	NAME	FUNCTION
1	A1	Input/output 1. Referenced to V_{CCA}.
2	V_{CCA}	A-port supply voltage. 1.1 V ≤ V_{CCA} ≤ 3.6 V, V_{CCA} ≤ V_{CCB}.
3	A2	Input/output 2. Referenced to V_{CCA}.
4	A3	Input/output 3. Referenced to V_{CCA}.
5	A4	Input/output 4. Referenced to V_{CCA}.
6	A5	Input/output 5. Referenced to V_{CCA}.
7	A6	Input/output 6. Referenced to V_{CCA}.
8	A7	Input/output 7. Referenced to V_{CCA}.
9	A8	Input/output 8. Referenced to V_{CCA}.
10	OE	Output enable. Pull OE low to place all outputs in 3-state mode. Referenced to V_{CCA}.
11	GND	Ground
12	B8	Input/output 8. Referenced to V_{CCB}.
13	B7	Input/output 7. Referenced to V_{CCB}.
14	B6	Input/output 6. Referenced to V_{CCB}.
15	B5	Input/output 5. Referenced to V_{CCB}.
16	B4	Input/output 4. Referenced to V_{CCB}.
17	B3	Input/output 3. Referenced to V_{CCB}.
18	B2	Input/output 2. Referenced to V_{CCB}.
19	V_{CCB}	B-port supply voltage. 1.65 V ≤ V_{CCB} ≤ 5.5 V.
20	B1	Input/output 1. Referenced to V_{CCB}.

TYPICAL OPERATING CIRCUIT



Absolute Maximum Ratings[1]

over operating free-air temperature range (unless otherwise noted)

			MIN	MAX	UNIT
V_{CCA}	Supply voltage range		–0.5	4.6	V
V_{CCB}	Supply voltage range		–0.5	6.5	V
V_I	Input voltage range[2]		–0.5	6.5	V
V_O	Voltage range applied to any output in the high-impedance or power-off state[2]		–0.5	6.5	V
V_O	Voltage range applied to any output in the high or low state[2] [3]	A inputs	–0.5	V_{CCA} + 0.5	V
		B inputs	–0.5	V_{CCB} + 0.5	
I_{IK}	Input clamp current	V_I < 0		–50	mA
I_{OK}	Output clamp current	V_O < 0		–50	mA
I_O	Continuous output current			±50	mA
	Continuous current through V_{CCA}, V_{CCB}, or GND			±100	mA
θ_{JA}	Package thermal impedance	DQS package		TBD	°C/W
		GXY/ZXY package[4]		78	
		PW package[4]		83	
		RGY package[5]		37	
T_{stg}	Storage temperature range		–65	150	°C

(1) Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) The input and output negative-voltage ratings may be exceeded if the input and output current ratings are observed.
(3) The value of V_{CCA} and V_{CCB} are provided in the recommended operating conditions table.
(4) The package thermal impedance is calculated in accordance with JESD 51-7.
(5) The package thermal impedance is calculated in accordance with JESD 51-5.

Recommended Operating Conditions[1] [2]

			V_{CCA}	V_{CCB}	MIN	MAX	UNIT
V_{CCA}	Supply voltage				1.2	3.6	V
V_{CCB}					1.65	5.5	
V_{IH}	High-level input voltage	Data inputs	1.2 V to 3.6 V	1.65 V to 5.5 V	V_{CCI} x 0.65[3]	V_{CCI}	V
		OE			V_{CCA} x 0.65	5.5	
V_{IL}	Low-level input voltage	Data inputs	1.2 V to 5.5 V	1.65 V to 5.5 V	0	V_{CCI} x 0.35[3]	V
		OE	1.2 V to 3.6 V		0	V_{CCA} x 0.35	
Δt/Δv	Input transition rise or fall rate	A-port inputs	1.2 V to 3.6 V	1.65 V to 5.5 V		40	ns/V
		B-port inputs	1.2 V to 3.6 V	1.65 V to 3.6 V		40	
				4.5 V to 5.5 V		30	
T_A	Operating free-air temperature				–40	85	°C

(1) The A and B sides of an unused data I/O pair must be held in the same state, i.e., both at V_{CCI} or both at GND.
(2) V_{CCA} must be less than or equal to V_{CCB} and must not exceed 3.6 V.
(3) V_{CCI} is the supply voltage associated with the input port.

Electrical Characteristics[(1)] [(2)]

over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	V_{CCA}	V_{CCB}	T_A = 25°C MIN	T_A = 25°C TYP	T_A = 25°C MAX	–40°C to 85°C MIN	–40°C to 85°C MAX	UNIT
V_{OHA}		I_{OH} = –20 µA	1.2 V			1.1				V
			1.4 V to 3.6 V					V_{CCA} – 0.4		
V_{OLA}		I_{OL} = 20 µA	1.2 V			0.9				V
			1.4 V to 3.6 V						0.4	
V_{OHB}		I_{OH} = –20 µA		1.65 V to 5.5 V				V_{CCB} – 0.4		V
V_{OLB}		I_{OL} = 20 µA		1.65 V to 5.5 V					0.4	V
I_I	OE		1.2 V to 3.6 V	1.65 V to 5.5 V			±1		±2	µA
I_{off}	A port		0 V	0 V to 5.5 V			±1		±2	µA
	B port		0 V to 3.6 V	0 V			±1		±2	
I_{OZ}	A or B port	OE = GND	1.2 V to 3.6 V	1.65 V to 5.5 V			±1		±2	µA
I_{CCA}		$V_I = V_{CCI}$ or GND, I_O = 0	1.2 V	1.65 V to 5.5 V		0.06				µA
			1.4 V to 3.6 V						5	
			3.6 V	0 V					2	
			0 V	5.5 V					–2	
I_{CCB}		$V_I = V_{CCI}$ or GND, I_O = 0	1.2 V	1.65 V to 5.5 V		3.4				µA
			1.4 V to 3.6 V						5	
			3.6 V	0 V					–2	
			0 V	5.5 V					2	
I_{CCA} + I_{CCB}		$V_I = V_{CCI}$ or GND, I_O = 0	1.2 V	1.65 V to 5.5 V		3.5				µA
			1.4 V to 3.6 V						10	
I_{CCZA}		$V_I = V_{CCI}$ or GND, I_O = 0, OE = GND	1.2 V	1.65 V to 5.5 V		0.05				µA
			1.4 V to 3.6 V						5	
I_{CCZB}		$V_I = V_{CCI}$ or GND, I_O = 0, OE = GND	1.2 V	1.65 V to 5.5 V		3.3				µA
			1.4 V to 3.6 V						5	
C_I	OE		1.2 V to 3.6 V	1.65 V to 5.5 V		5			5.5	pF
C_{io}	A port		1.2 V to 3.6 V	1.65 V to 5.5 V		5			6.5	pF
	B port					8			10	

(1) V_{CCI} is the supply voltage associated with the input port.
(2) V_{CCO} is the supply voltage associated with the output port.

Timing Requirements

T_A = 25°C, V_{CCA} = 1.2 V

			V_{CCB} = 1.8 V TYP	V_{CCB} = 2.5 V TYP	V_{CCB} = 3.3 V TYP	V_{CCB} = 5 V TYP	UNIT
	Data rate		20	20	20	20	Mbps
t_w	Pulse duration	Data inputs	50	50	50	50	ns

Timing Requirements

over recommended operating free-air temperature range, V_{CCA} = 1.5 V ± 0.1 V (unless otherwise noted)

			V_{CCB} = 1.8 V ± 0.15 V MIN	V_{CCB} = 1.8 V ± 0.15 V MAX	V_{CCB} = 2.5 V ± 0.2 V MIN	V_{CCB} = 2.5 V ± 0.2 V MAX	V_{CCB} = 3.3 V ± 0.3 V MIN	V_{CCB} = 3.3 V ± 0.3 V MAX	V_{CCB} = 5 V ± 0.5 V MIN	V_{CCB} = 5 V ± 0.5 V MAX	UNIT
	Data rate			50		50		50		50	Mbps
t_w	Pulse duration	Data inputs	20		20		20		20		ns

Timing Requirements

over recommended operating free-air temperature range, V_{CCA} = 1.8 V ± 0.15 V (unless otherwise noted)

			V_{CCB} = 1.8 V ± 0.15 V		V_{CCB} = 2.5 V ± 0.2 V		V_{CCB} = 3.3 V ± 0.3 V		V_{CCB} = 5 V ± 0.5 V		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
	Data rate			52		60		60		60	Mbps
t_w	Pulse duration	Data inputs	19		17		17		17		ns

Timing Requirements

over recommended operating free-air temperature range, V_{CCA} = 2.5 V ± 0.2 V (unless otherwise noted)

			V_{CCB} = 2.5 V ± 0.2 V		V_{CCB} = 3.3 V ± 0.3 V		V_{CCB} = 5 V ± 0.5 V		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
	Data rate			70		100		100	Mbps
t_w	Pulse duration	Data inputs	14		10		10		ns

Timing Requirements

over recommended operating free-air temperature range, V_{CCA} = 3.3 V ± 0.3 V (unless otherwise noted)

			V_{CCB} = 3.3 V ± 0.3 V		V_{CCB} = 5 V ± 0.5 V		UNIT
			MIN	MAX	MIN	MAX	
	Data rate			100		100	Mbps
t_w	Pulse duration	Data inputs	10		10		ns

Switching Characteristics

T_A = 25°C, V_{CCA} = 1.2 V

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CCB} = 1.8 V	V_{CCB} = 2.5 V	V_{CCB} = 3.3 V	V_{CCB} = 5 V	UNIT
			TYP	TYP	TYP	TYP	
t_{pd}	A	B	9.5	7.9	7.6	8.5	ns
	B	A	9.2	8.8	8.4	8	
t_{en}	OE	A	1	1	1	1	µs
		B	1	1	1	1	
t_{dis}	OE	A	20	17	17	18	ns
		B	20	16	15	15	
t_{rA}, t_{fA}	A-port rise and fall times		4.1	4.4	4.1	3.9	ns
t_{rB}, t_{fB}	B-port rise and fall times		5	5	5.1	5.1	ns
$t_{SK(O)}$	Channel-to-channel skew		2.4	1.7	1.9	7	ns
Max data rate			20	20	20	20	Mbps

Switching Characteristics

over recommended operating free-air temperature range, V_{CCA} = 1.5 V ± 0.1 V (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CCB} = 1.8 V ± 0.15 V		V_{CCB} = 2.5 V ± 0.2 V		V_{CCB} = 3.3 V ± 0.3 V		V_{CCB} = 5 V ± 0.5 V		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A	B	1.4	12.9	1.2	10.1	1.1	10	0.8	9.9	ns
	B	A	0.9	14.2	0.7	12	0.4	11.7	0.3	13.7	
t_{en}	OE	A		1		1		1		1	μs
		B		1		1		1		1	
t_{dis}	OE	A	6.6	33	6.4	25.3	6.1	23.1	5.9	24.6	ns
		B	6.6	35.6	5.8	25.6	5.5	22.1	5.6	20.6	
t_{rA}, t_{fA}	A-port rise and fall times		0.8	6.5	0.8	6.3	0.8	6.3	0.8	6.3	ns
t_{rB}, t_{fB}	B-port rise and fall times		1	7.3	0.7	4.9	0.7	4.6	0.6	4.6	ns
$t_{SK(O)}$	Channel-to-channel skew			2.6		1.9		1.6		1.3	ns
Max data rate			50		50		50		50		Mbps

Switching Characteristics

over recommended operating free-air temperature range, V_{CCA} = 1.8 V ± 0.15 V (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CCB} = 1.8 V ± 0.15 V		V_{CCB} = 2.5 V ± 0.2 V		V_{CCB} = 3.3 V ± 0.3 V		V_{CCB} = 5 V ± 0.5 V		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A	B	1.6	11	1.4	7.7	1.3	6.8	1.2	6.5	ns
	B	A	1.5	12	1.2	8.4	0.8	7.6	0.5	7.1	
t_{en}	OE	A		1		1		1		1	μs
		B		1		1		1		1	
t_{dis}	OE	A	5.9	26.7	5.6	21.6	5.4	18.9	4.8	18.7	ns
		B	6.1	33.9	5.2	23.7	5	19.9	5	17.6	
t_{rA}, t_{fA}	A-port rise and fall times		0.7	5.1	0.7	5	1	5	0.7	5	ns
t_{rB}, t_{fB}	B-port rise and fall times		1	7.3	0.7	5	0.7	3.9	0.6	3.8	ns
$t_{SK(O)}$	Channel-to-channel skew			0.8		0.7		0.6		0.6	ns
Max data rate			52		60		60		60		Mbps

Switching Characteristics

over recommended operating free-air temperature range, V_{CCA} = 2.5 V ± 0.2 V (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CCB} = 2.5 V ± 0.2 V MIN	MAX	V_{CCB} = 3.3 V ± 0.3 V MIN	MAX	V_{CCB} = 5 V ± 0.5 V MIN	MAX	UNIT
t_{pd}	A	B	1.1	6.4	1	5.3	0.9	4.7	ns
	B	A	1	7	0.6	5.6	0.3	4.4	
t_{en}	OE	A		1		1		1	µs
		B		1		1		1	
t_{dis}	OE	A	5	16.9	4.9	15	4.5	13.8	ns
		B	4.8	21.8	4.5	17.9	4.4	15.2	
t_{rA}, t_{fA}	A-port rise and fall times		0.8	3.6	0.6	3.6	0.5	3.5	ns
t_{rB}, t_{fB}	B-port rise and fall times		0.6	4.9	0.7	3.9	0.6	3.2	ns
$t_{SK(O)}$	Channel-to-channel skew			0.4		0.3		0.3	ns
Max data rate			70		100		100		Mbps

Switching Characteristics

over recommended operating free-air temperature range, V_{CCA} = 3.3 V ± 0.3 V (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CCB} = 3.3 V ± 0.3 V MIN	MAX	V_{CCB} = 5 V ± 0.5 V MIN	MAX	UNIT
t_{pd}	A	B	0.9	4.9	0.8	4	ns
	B	A	0.5	5.4	0.2	4	
t_{en}	OE	A		1		1	µs
		B		1		1	
t_{dis}	OE	A	4.5	13.9	4.1	12.4	ns
		B	4.1	17.3	4	14.4	
t_{rA}, t_{fA}	A-port rise and fall times		0.5	3	0.5	3	ns
t_{rB}, t_{fB}	B-port rise and fall times		0.7	3.9	0.6	3.2	ns
$t_{SK(O)}$	Channel-to-channel skew			0.4		0.3	ns
Max data rate			100		100		Mbps

Operating Characteristics

T_A = 25°C

PARAMETER		TEST CONDITIONS	V_{CCA} 1.2 V / V_{CCB} 5 V	V_{CCA} 1.2 V / V_{CCB} 1.8 V	V_{CCA} 1.5 V / V_{CCB} 1.8 V	V_{CCA} 1.8 V / V_{CCB} 1.8 V	V_{CCA} 2.5 V / V_{CCB} 2.5 V	V_{CCA} 2.5 V / V_{CCB} 5 V	V_{CCA} 3.3 V / V_{CCB} 3.3 V to 5 V	UNIT
			TYP	TYP	TYP	TYP	TYP	TYP	TYP	
C_{pdA}	A-port input, B-port output	C_L = 0, f = 10 MHz, $t_r = t_f$ = 1 ns, OE = V_{CCA} (outputs enabled)	9	8	7	7	7	7	8	pF
	B-port input, A-port output		12	11	11	11	11	11	11	
C_{pdB}	A-port input, B-port output		35	26	27	27	27	27	28	
	B-port input, A-port output		26	19	18	18	18	20	21	
C_{pdA}	A-port input, B-port output	C_L = 0, f = 10 MHz, $t_r = t_f$ = 1 ns, OE = GND (outputs disabled)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	pF
	B-port input, A-port output		0.01	0.01	0.01	0.01	0.01	0.01	0.01	
C_{pdB}	A-port input, B-port output		0.01	0.01	0.01	0.01	0.01	0.01	0.03	
	B-port input, A-port output		0.01	0.01	0.01	0.01	0.01	0.01	0.03	

PRINCIPLES OF OPERATION

Applications

The TXB0108 can be used in level-translation applications for interfacing devices or systems operating at different interface voltages with one another.

Architecture

The TXB0108 architecture (see Figure 1) does not require a direction-control signal to control the direction of data flow from A to B or from B to A. In a dc state, the output drivers of the TXB0108 can maintain a high or low, but are designed to be weak, so that they can be overdriven by an external driver when data on the bus starts flowing the opposite direction.

The output one shots detect rising or falling edges on the A or B ports. During a rising edge, the one shot turns on the PMOS transistors (T1, T3) for a short duration, which speeds up the low-to-high transition. Similarly, during a falling edge, the one shot turns on the NMOS transistors (T2, T4) for a short duration, which speeds up the high-to-low transition. The typical output impedance during output transition is 70Ω at V_{CCO} = 1.2 V to 1.8 V, 50Ω at V_{CCO} = 1.8 V to 3.3 V and 40Ω at V_{CCO} = 3.3 V to 5 V.

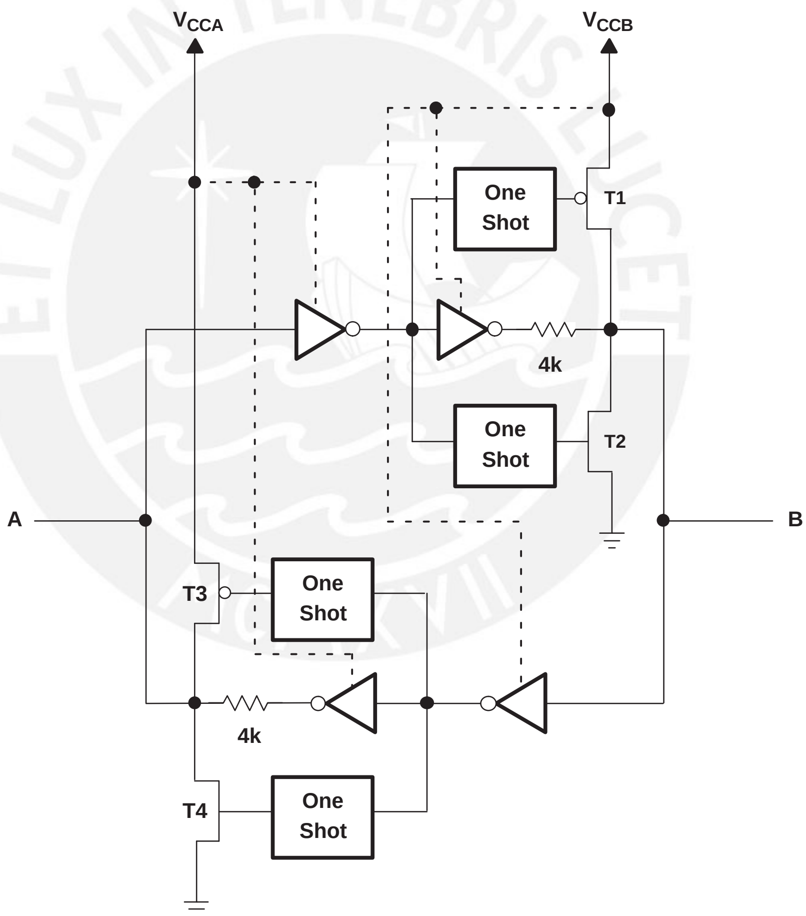
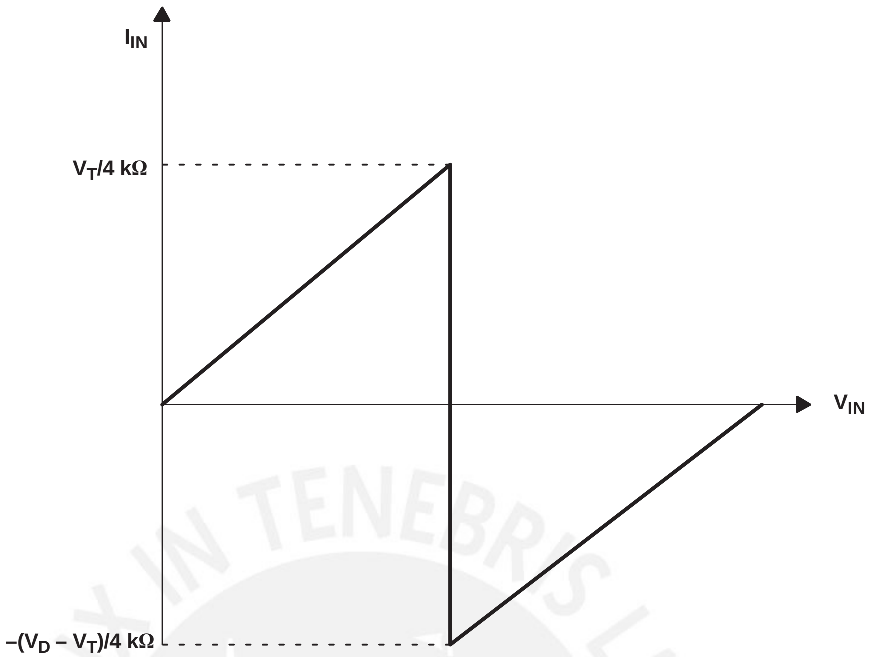


Figure 1. Architecture of TXB0108 I/O Cell

Input Driver Requirements

Typical I_{IN} vs V_{IN} characteristics of the TXB0108 are shown in Figure 2. For proper operation, the device driving the data I/Os of the TXB0108 must have drive strength of at least ±2 mA.



A. V_T is the input threshold voltage of the TXB0108 (typically $V_{CCI}/2$).

B. V_D is the supply voltage of the external driver.

Figure 2. Typical I_{IN} vs V_{IN} Curve

Power Up

During operation, ensure that $V_{CCA} \le V_{CCB}$ at all times. During power-up sequencing, $V_{CCA} \ge V_{CCB}$ does not damage the device, so any power supply can be ramped up first. The TXB0108 has circuitry that disables all output ports when either V_{CC} is switched off ($V_{CCA/B}$ = 0 V).

Enable and Disable

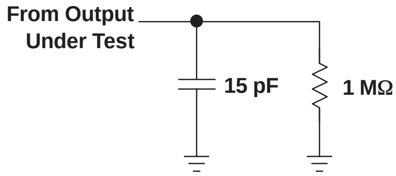
The TXB0108 has an OE input that is used to disable the device by setting OE = low, which places all I/Os in the high-impedance (Hi-Z) state. The disable time (t_{dis}) indicates the delay between when OE goes low and when the outputs actually get disabled (Hi-Z). The enable time (t_{en}) indicates the amount of time the user must allow for the one-shot circuitry to become operational after OE is taken high.

Pullup or Pulldown Resistors on I/O Lines

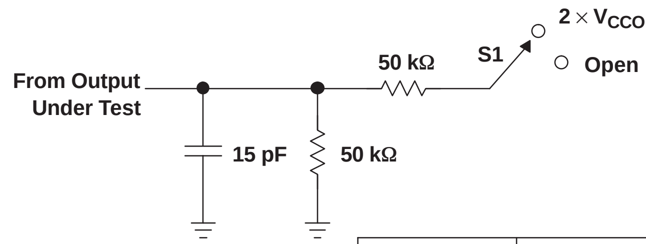
The TXB0108 is designed to drive capacitive loads of up to 70 pF. The output drivers of the TXB0108 have low dc drive strength. If pullup or pulldown resistors are connected externally to the data I/Os, their values must be kept higher than 50 kΩ to ensure that they do not contend with the output drivers of the TXB0108.

For the same reason, the TXB0108 should not be used in applications such as I^2C or 1-Wire where an open-drain driver is connected on the bidirectional data I/O. For these applications, use a device from the TI TXS01xx series of level translators.

PARAMETER MEASUREMENT INFORMATION

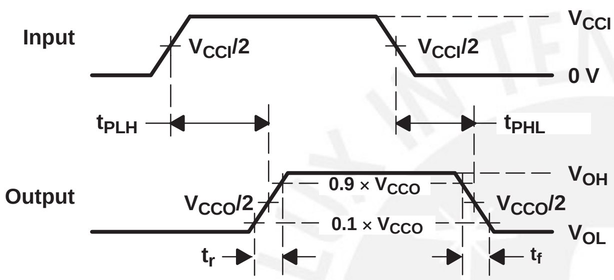


LOAD CIRCUIT FOR MAX DATA RATE, PULSE DURATION PROPAGATION DELAY OUTPUT RISE AND FALL TIME MEASUREMENT

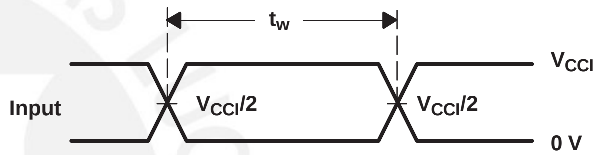


LOAD CIRCUIT FOR ENABLE/DISABLE TIME MEASUREMENT

TEST	S1
t_{PZL}/t_{PLZ}	$2 \times V_{CCO}$
t_{PHZ}/t_{PZH}	Open



VOLTAGE WAVEFORMS PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS PULSE DURATION

A. C_L includes probe and jig capacitance.
B. All input pulses are supplied by generators having the following characteristics: PRR ≤ 10 MHz, Z_O = 50 Ω, dv/dt ≥ 1 V/ns.
C. The outputs are measured one at a time, with one transition per measurement.
D. t_{PLH} and t_{PHL} are the same as t_{pd}.
E. V_{CCI} is the V_{CC} associated with the input port.
F. V_{CCO} is the V_{CC} associated with the output port.
G. All parameters and waveforms are not applicable to all devices.

Figure 3. Load Circuits and Voltage Waveforms

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/ Ball Finish	MSL Peak Temp (3)	Samples (Requires Login)
TXB0108DQSR	ACTIVE	USON	DQS	20	3000	Green (RoHS & no Sb/Br)	Call TI	Level-1-260C-UNLIM	Request Free Samples
TXB0108PWR	ACTIVE	TSSOP	PW	20	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	Request Free Samples
TXB0108PWRG4	ACTIVE	TSSOP	PW	20	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	Request Free Samples
TXB0108RGYR	ACTIVE	VQFN	RGY	20	3000	Green (RoHS & no Sb/Br)	Call TI	Level-2-260C-1 YEAR	Request Free Samples
TXB0108RGYRG4	ACTIVE	VQFN	RGY	20	3000	Green (RoHS & no Sb/Br)	Call TI	Level-2-260C-1 YEAR	Request Free Samples
TXB0108ZXYR	ACTIVE	BGA MICROSTAR JUNIOR	ZXY	20	2500	Green (RoHS & no Sb/Br)	SNAGCU	Level-1-260C-UNLIM	Request Free Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

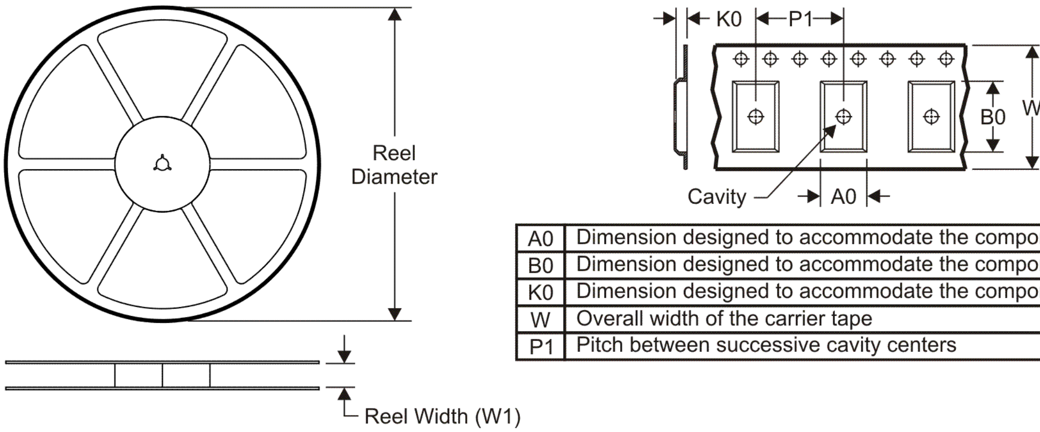
(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

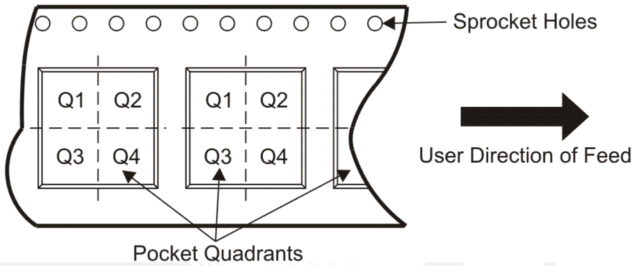
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

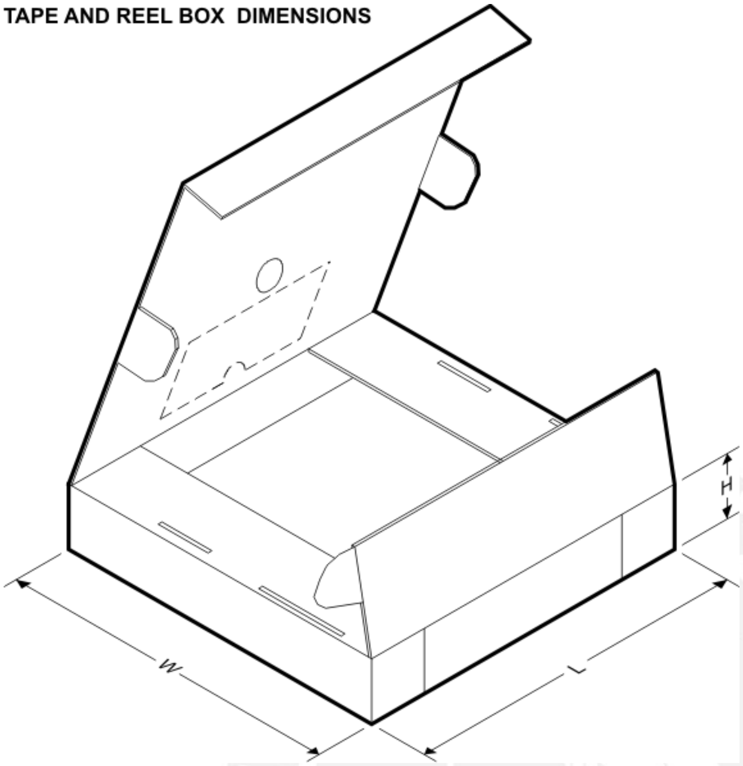
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
TXB0108DQSR	USON	DQS	20	3000	177.8	12.4	2.21	4.22	0.81	4.0	12.0	Q1
TXB0108PWR	TSSOP	PW	20	2000	330.0	16.4	6.95	7.1	1.6	8.0	16.0	Q1
TXB0108RGYR	VQFN	RGY	20	3000	330.0	12.4	3.8	4.8	1.6	8.0	12.0	Q1
TXB0108ZXYR	BGA MI CROSTA R JUNI OR	ZXY	20	2500	330.0	12.4	2.8	3.3	1.0	4.0	12.0	Q2

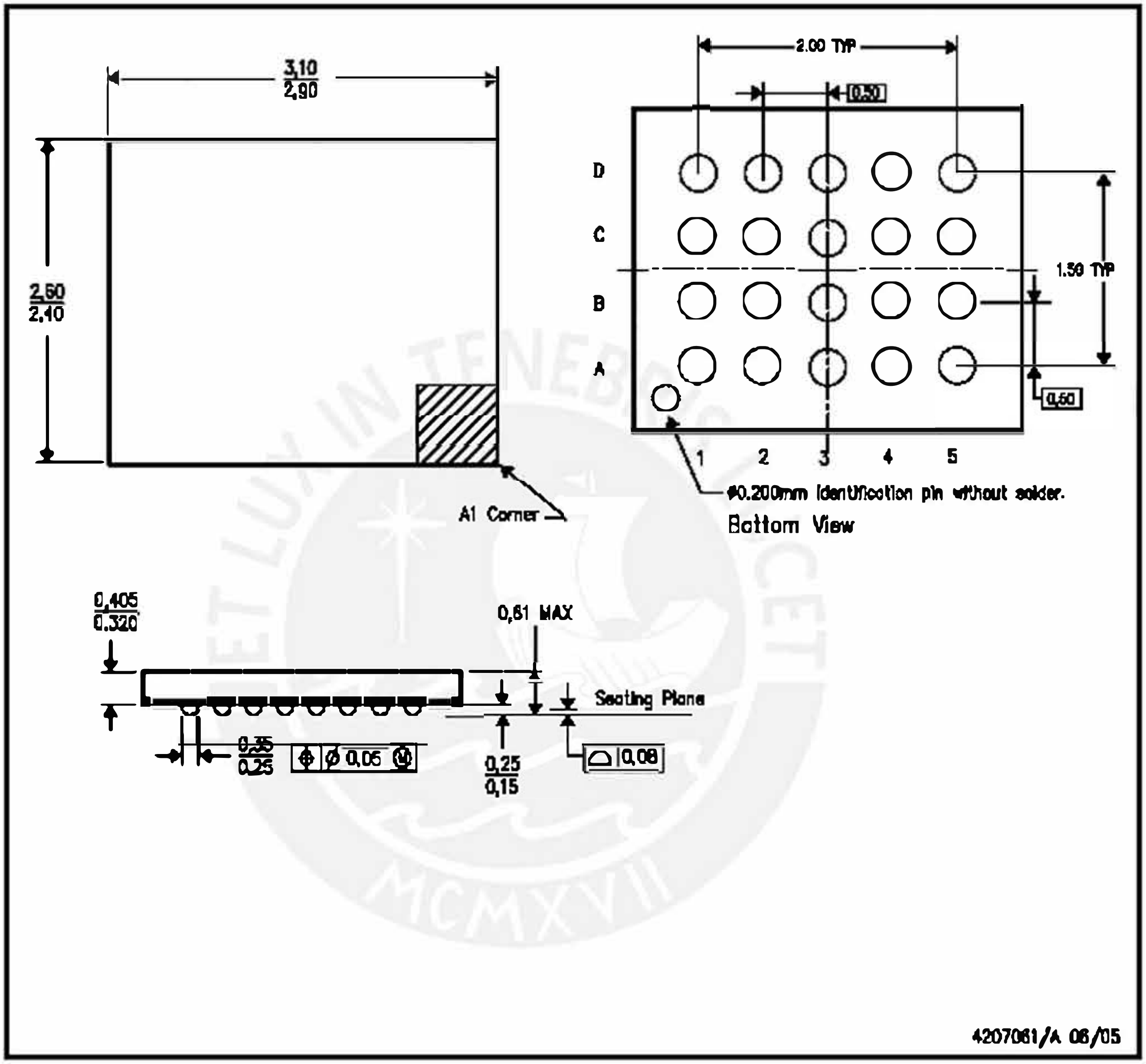
TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

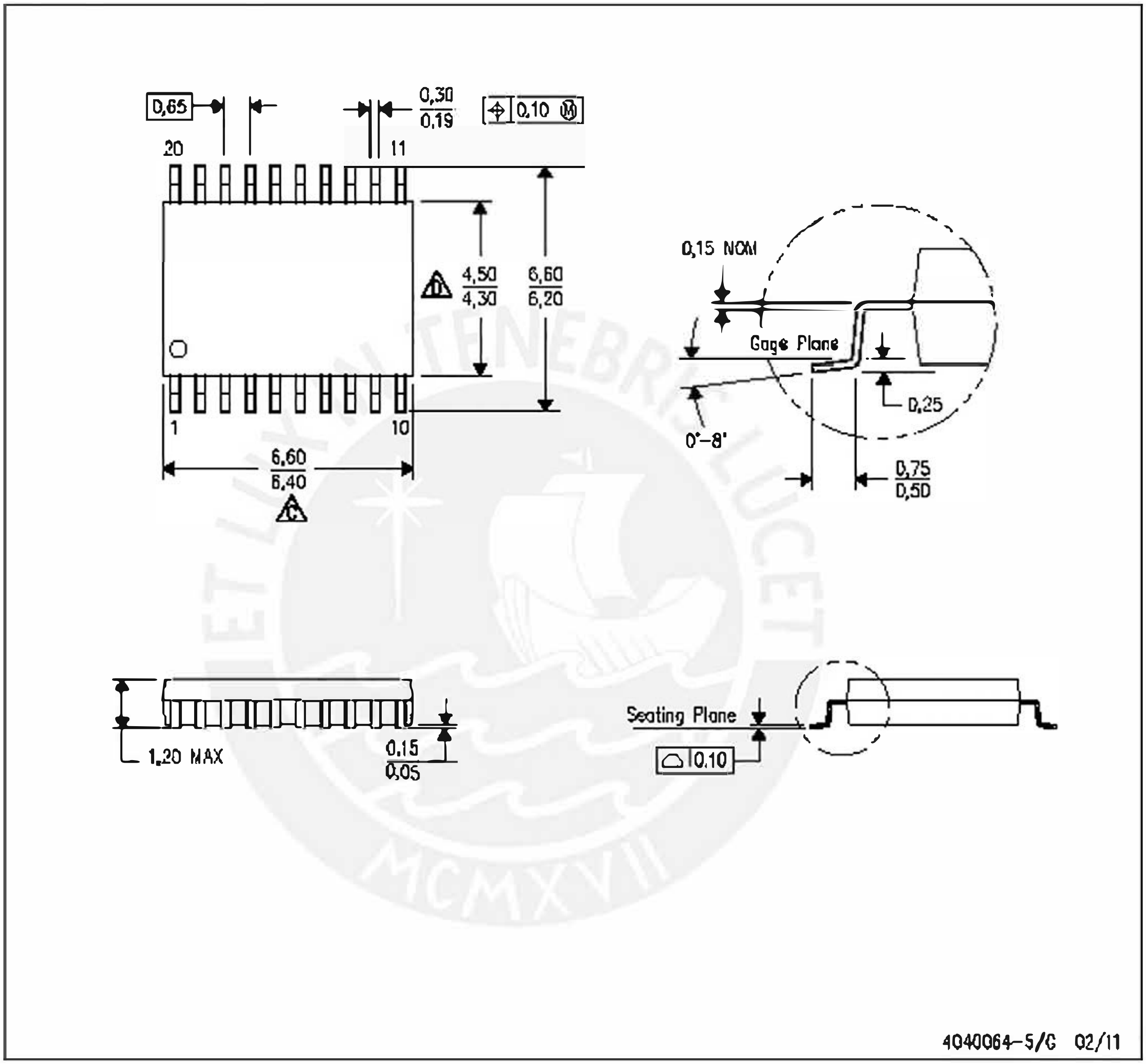
Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
TXB0108DQSR	USON	DQS	20	3000	202.0	201.0	28.0
TXB0108PWR	TSSOP	PW	20	2000	346.0	346.0	33.0
TXB0108RGYR	VQFN	RGY	20	3000	355.0	350.0	50.0
TXB0108ZXYR	BGA MICROSTAR JUNIOR	ZXY	20	2500	340.5	338.1	20.6

ZXY (S-PBGA-N20) PLASTIC BALL GRID ARRAY



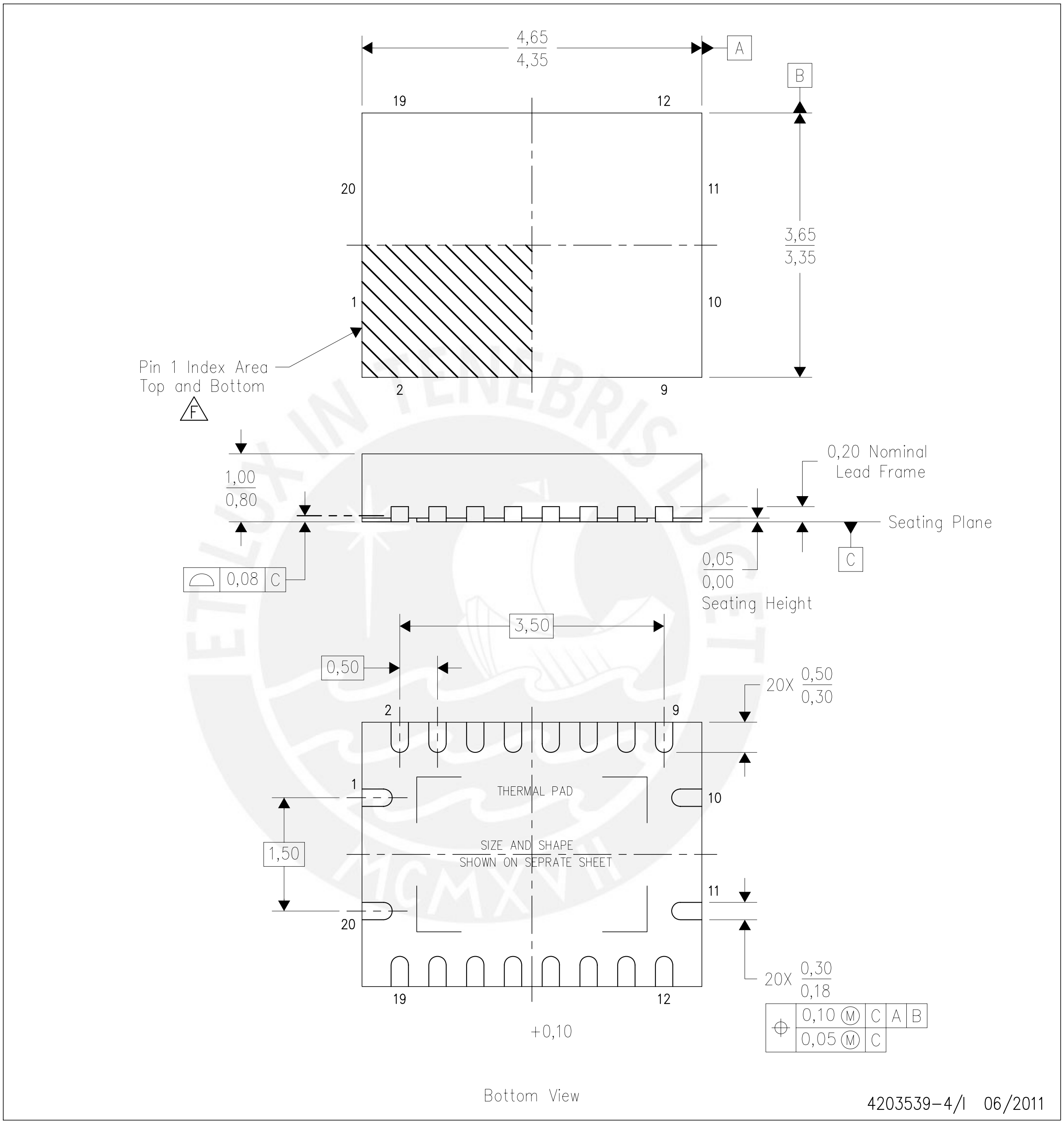
NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. This package is a lead-free solder ball design.

PW (R-PDSO-G20) PLASTIC SMALL OUTLINE



NOTES: A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.
B. This drawing is subject to change without notice.
C. Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0,15 each side.
D. Body width does not include interlead flash. Interlead flash shall not exceed 0,25 each side.
E. Falls within JEDEC MO-153

RGY (R–PVQFN–N20) PLASTIC QUAD FLATPACK NO–LEAD



NOTES:
- A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
- B. This drawing is subject to change without notice.
- C. QFN (Quad Flatpack No–Lead) package configuration.
- D. The package thermal pad must be soldered to the board for thermal and mechanical performance.
- E. See the additional figure in the Product Data Sheet for details regarding the exposed thermal pad features and dimensions.
- F. Pin 1 identifiers are located on both top and bottom of the package and within the zone indicated. The Pin 1 identifiers are either a molded, marked, or metal feature.
- G. Package complies to JEDEC MO–241 variation BA.

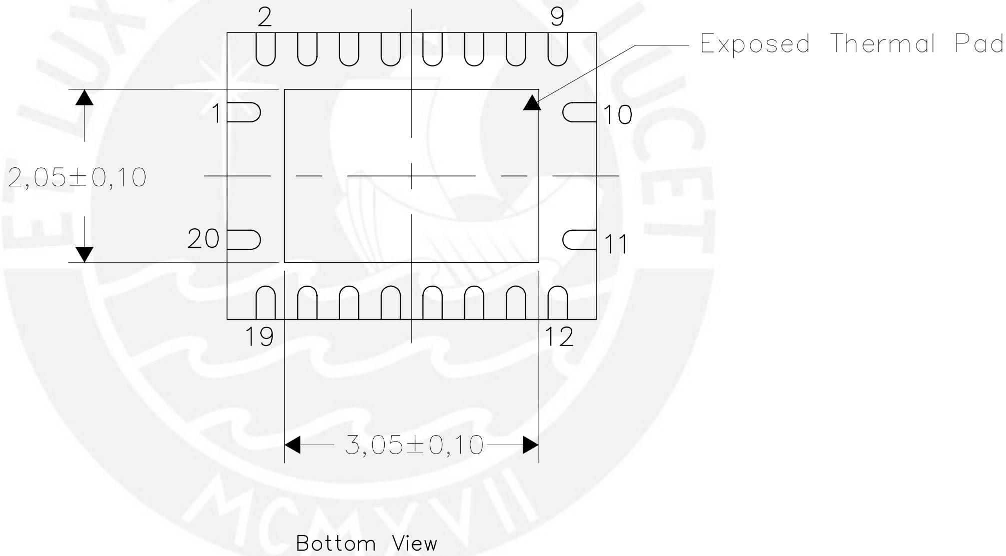
RGY (R–PVQFN–N20) PLASTIC QUAD FLATPACK NO–LEAD

THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No–Lead (QFN) package and its advantages, refer to Application Report, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271. This document is available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.

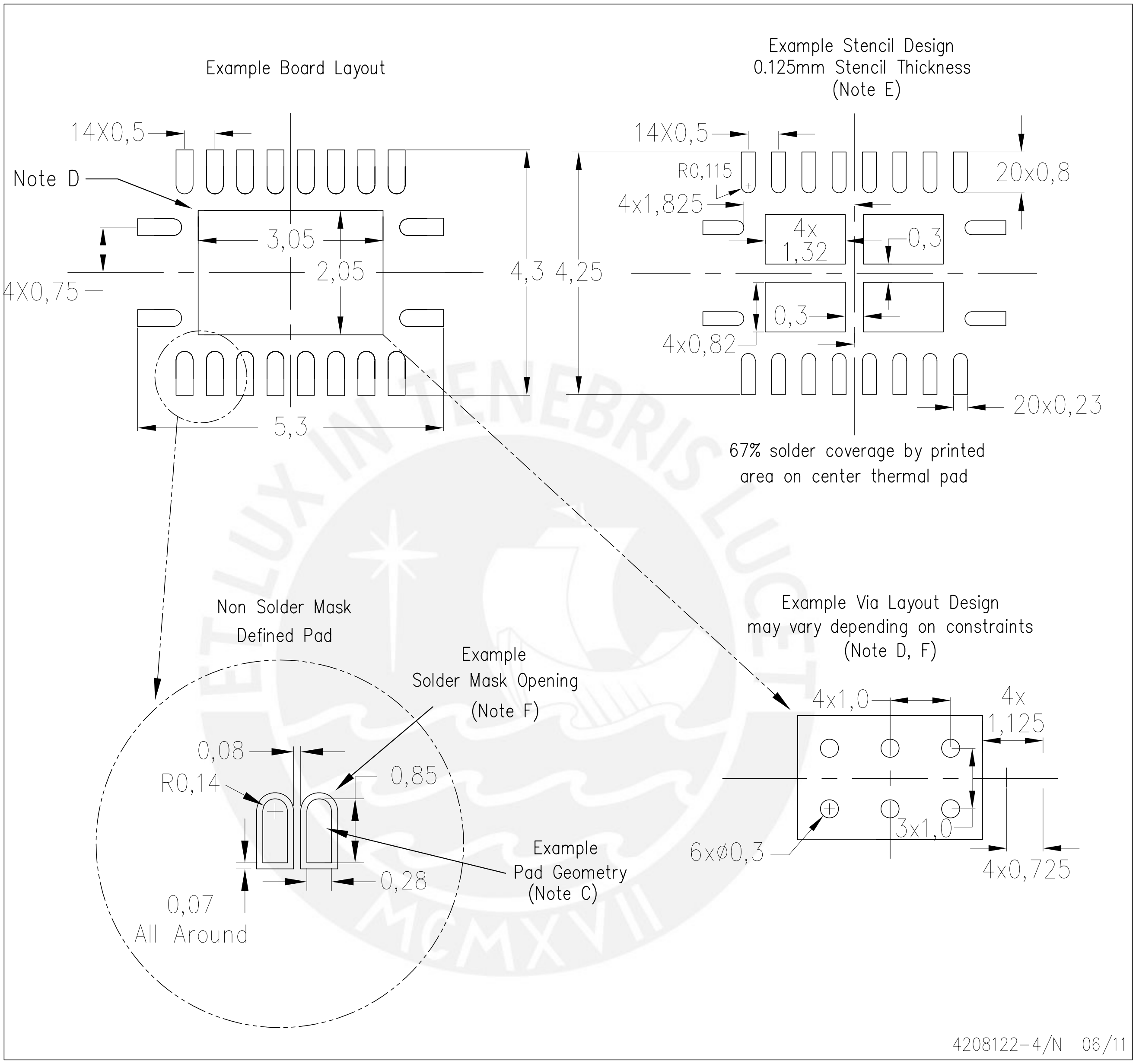


Exposed Thermal Pad Dimensions

4206353–4/N 06/11

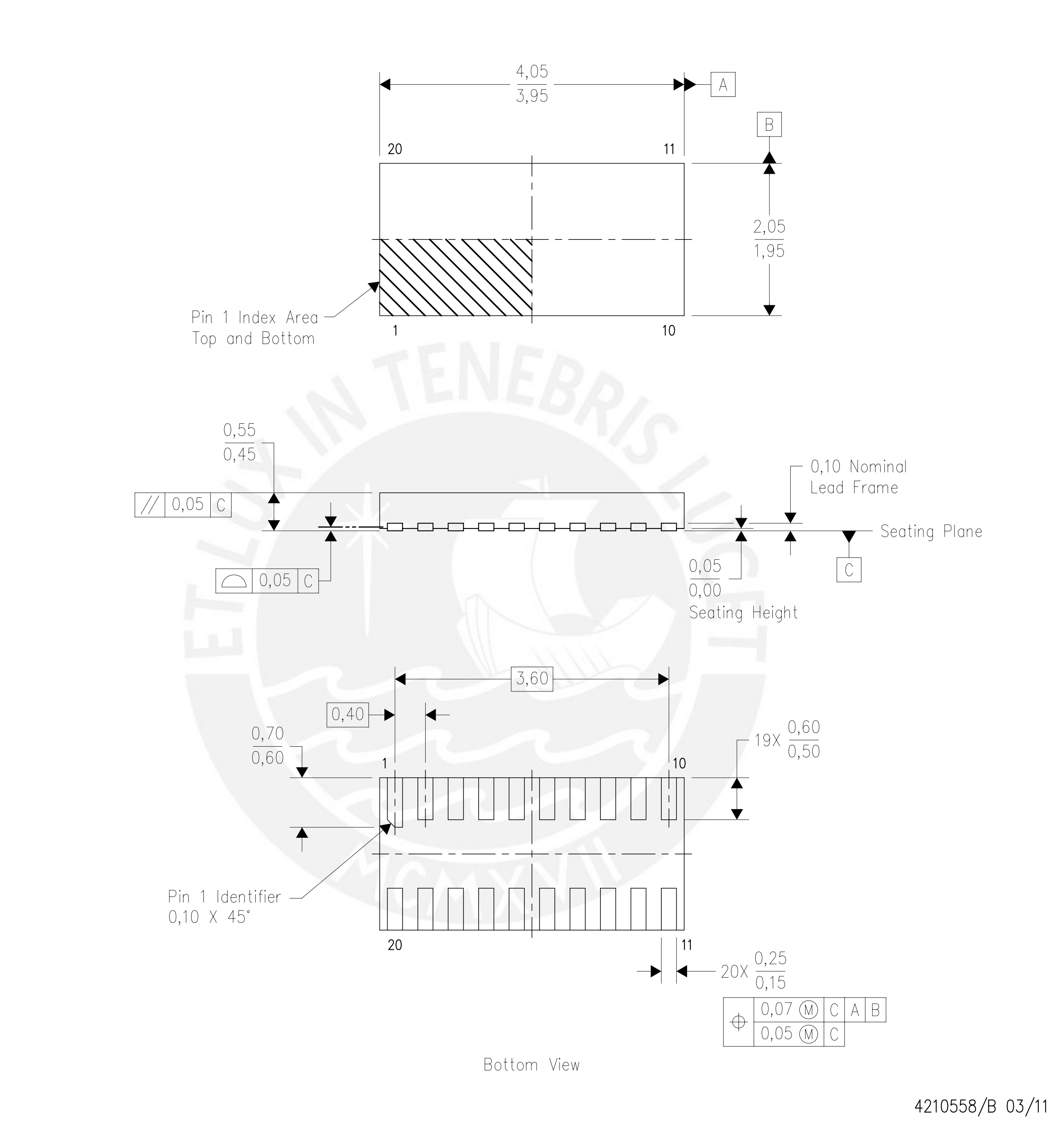
NOTE: A. All linear dimensions are in millimeters

RGY (R–PVQFN–N20) PLASTIC QUAD FLATPACK NO–LEAD



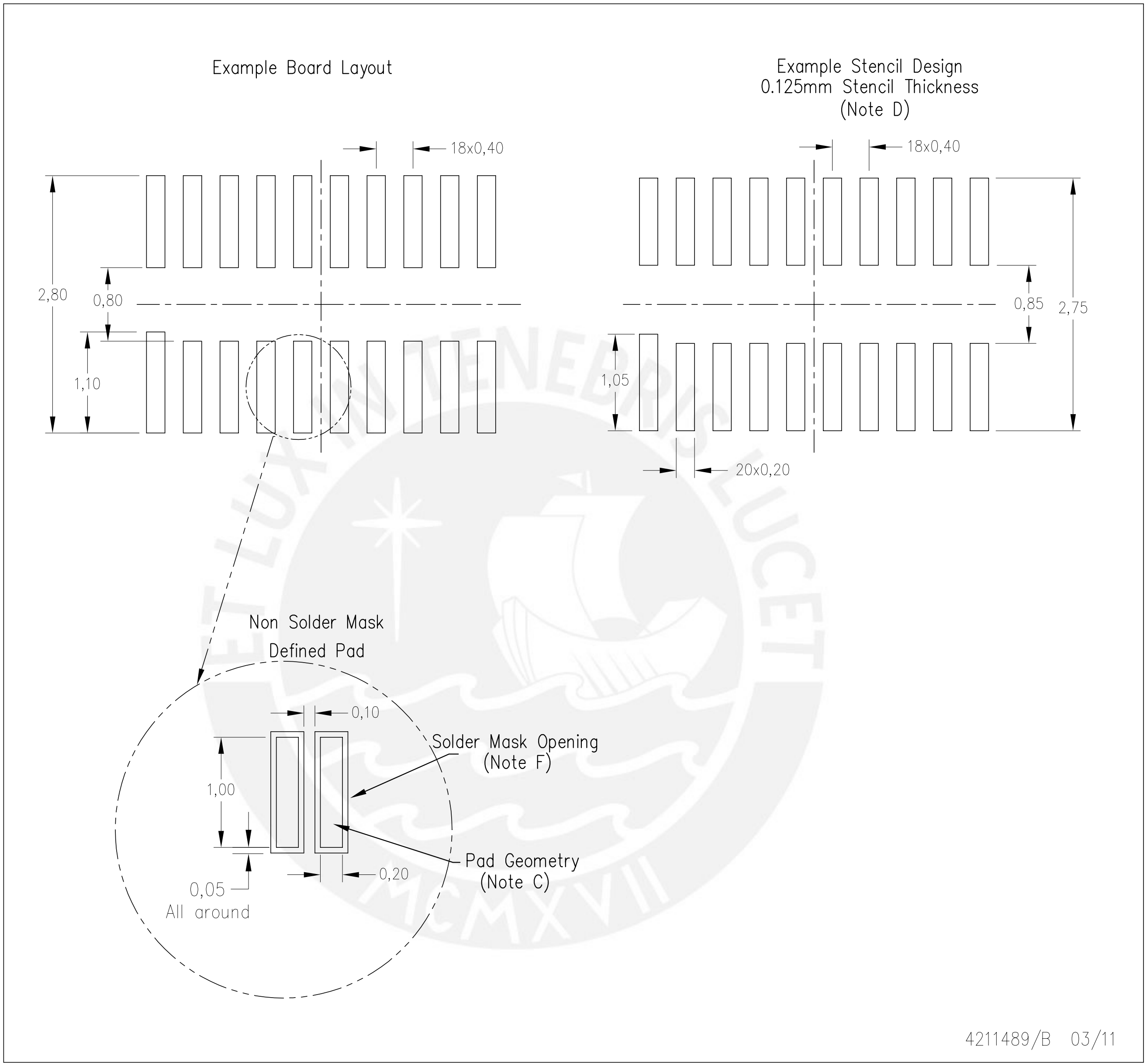
NOTES:
- A. All linear dimensions are in millimeters.
- B. This drawing is subject to change without notice.
- C. Publication IPC–7351 is recommended for alternate designs.
- D. This package is designed to be soldered to a thermal pad on the board. Refer to Application Note, Quad Flat–Pack QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at www.ti.com <http://www.ti.com>.
- E. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
- F. Customers should contact their board fabrication site for minimum solder mask web tolerances between signal pads.

DQS (R–PUSON–N20) PLASTIC SMALL OUTLINE NO–LEAD



NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. SON (Small Outline No–Lead) package configuration.

DQS (R-PUSON-N20) PLASTIC SMALL OUTLINE NO-LEAD



NOTES:
- A. All linear dimensions are in millimeters.
- B. This drawing is subject to change without notice.
- C. Publication IPC-7351 is recommended for alternate designs.
- D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
- E. Customers should contact their board fabrication site for minimum solder mask web tolerances between signal pads.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com	Wireless	www.ti.com/wireless-apps
RF/IF and ZigBee® Solutions	www.ti.com/lprf		
		TI E2E Community Home Page	e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Módulo Wi-Fi

DESCRIPTION

Make your Internet of Things device cable-free by adding WiFi. Take advantage of the Raspberry Pi and Beagle Bone's USB port to add a low cost, but high-reliability wireless link. We tried half a dozen modules to find one that works well with the Pi and Bone without the need of recompiling any kernels: its supported by the Bone's Angstrom/Debian installation that comes with each Bone as well as Raspbian and just about any other modern Pi operating system. You'll have wireless Internet in 10 minutes! Works great with 802.11b/g/n networks.

If using with a Beagle Bone: Because of the high power required by WiFi, a 5V 2A power adapter is required to power both the Bone and WiFi. Flaky behavior and crashes may result if this is not followed! We have a tutorial for using this module with the Beagle Bone!

If using with a Raspberry Pi: The latest Raspbian distributions support this module out-of-the-box. Check out our detailed tutorial for how to set up WiFi networking on the Pi If you have a model B you may find that you need to have a powered hub to use this adapter, so if you're having power flakiness with your Pi, try a hub!

Please note: These are good for home/office usage with a Raspberry Pi when the router is nearby. For installation projects or large-scale distribution, we strongly recommend upgrading to this stick with a larger antenna - especially if you plan on putting the Pi behind something or inside an enclosure/wall/box/sign/etc!

The WiFi module may look slightly different than above, but all modules shipped contain the same chipset and have equivalent performance, the only difference is the plastic shell and any printing. We have requested to minimize packaging and now ship without a CD since all modern computers have support built in. If you need drivers, see the Technical Details tab

---

TECHNICAL DETAILS

RoHS

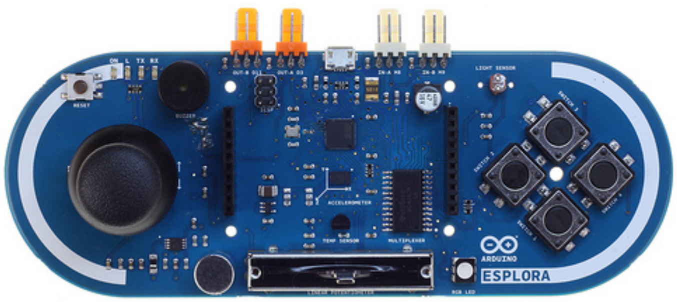
- RTl8192/8188CUS Chipset
- Sticks out 8mm (0.3") beyond USB port
- Weight: 2.17g

- Wireless Standards: IEEE 802.11n (draft), IEEE 802.11g, IEEE 802.11b
- Host Interface: High speed USB2.0/1.1 interface
- Data Rate: 802.11n: up to 150Mbps (downlink) and up to 150Mbps (uplink) , 802.11g: 54 / 48 / 36 / 24 / 18/ 12 / 9 / 6 Mbps auto fallback, 802.11b: 11 / 5.5 / 2 / 1 Mbps auto fallback
- Frequency Band: 2.4GHz ISM (Industrial Scientific Medical) Band
- Chipset: Realtek
- RF Frequency: 2412 ~ 2462 MHz (North America), 2412 ~ 2472 MHz (Europe), 2412 ~ 2484 MHz (Japan)
- Radio Channel: 1 ~ 14 channels (Universal Domain Selection)
- Range Coverage: Up to 3 times farther range than 802.11g
- Antenna Type: Integrated Antenna
- Roaming: Full mobility and seamless roaming from cell to cell
- RF Output Power: 13 ~17 dBm (Typical)
- Modulation: 11n: BPSK, QPSK, 16QAM, 64QAM with OFDM, 11g: BPSK, QPSK, 16QAM, 64QAM, OFDM, 11b: DQPSK, DBPSK, DSSS, CCK
- Data Security: 64/128-bit WEP Encryption
- WPA, WPA-PSK, WPA2, WPA2-PSK. TKIP/AES
- Network: Auto-switch to use 802.11n or 802.11g or 802.11b mode
- Supports Ad-Hoc, Infrastructure WLAN network, Wireless roaming, Data rate auto fall-back under noisy environment or longer range distance, Site Survey with Profile function
- Configuration & Management: Plug-and-Play setup and installation, Management Utility supports 2000 / XP/ Vista
- Media Access Control CSMA/CA with ACK
- LED Indicator Link/Active (Blue)
- Operating Temperature 0°C to 40°C
- Storage Temperature -20°C to 75°C
- Operating Humidity 10% ~ 90% (Non Condensing)
- Storage Humidity 5% ~ 95% (Non Condensing)
- Requirements: Available USB 2.0 port

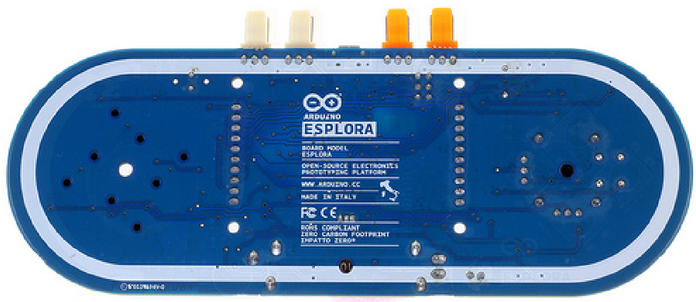
Anexo 15: Arduino Esplora

Search the Arduino Website

Arduino Esplora



(http://farm9.staticflickr.com/8044/8134573901_63952a4f17_b.jpg)
Arduino Esplora Front



(http://farm9.staticflickr.com/8471/8134600572_35b9d2e416_b.jpg)
Arduino Esplora Rear

Buy From Arduino Store (http://store.arduino.cc/it/index.php?main_page=product_info&cPath=11_12&products_id=250)

Buy From Distributors (http://arduino.cc/en/Main/Buy)

Overview

The Arduino Esplora is a microcontroller board derived from the Arduino Leonardo (http://arduino.cc/en/Guide/ArduinoLeonardo). The Esplora differs from all preceding Arduino boards in that it provides a number of built-in, ready-to-use set of onboard sensors for interaction. It's designed for people who want to get up and running with Arduino without having to learn about the electronics first. For a step-by-step introduction to the Esplora, check out the Getting Started with Esplora (http://arduino.cc/en/Guide/ArduinoEsplora) guide.
The Esplora has onboard sound and light outputs, and several input sensors, including a joystick, a slider, a temperature sensor, an accelerometer, a microphone, and a light sensor. It also has the potential to expand its capabilities with two Tinkerkit input and output connectors, and a socket for a color TFT LCD screen.
Like the Leonardo board, the Esplora uses an Atmega32U4 AVR microcontroller with 16 MHz crystal oscillator and a micro USB connection capable of acting as a USB client device, like a mouse or a keyboard.
In the upper left corner of the board there is a reset pushbutton, that you can use to restart the board. There are four status LEDS :

- ON [green] indicates whether the board is receiving power supply
- L [yellow] connected directly to the microcontroller, accessible through pin 13
- RX and TX [yellow] indicates the data being transmitted or received over the USB communication

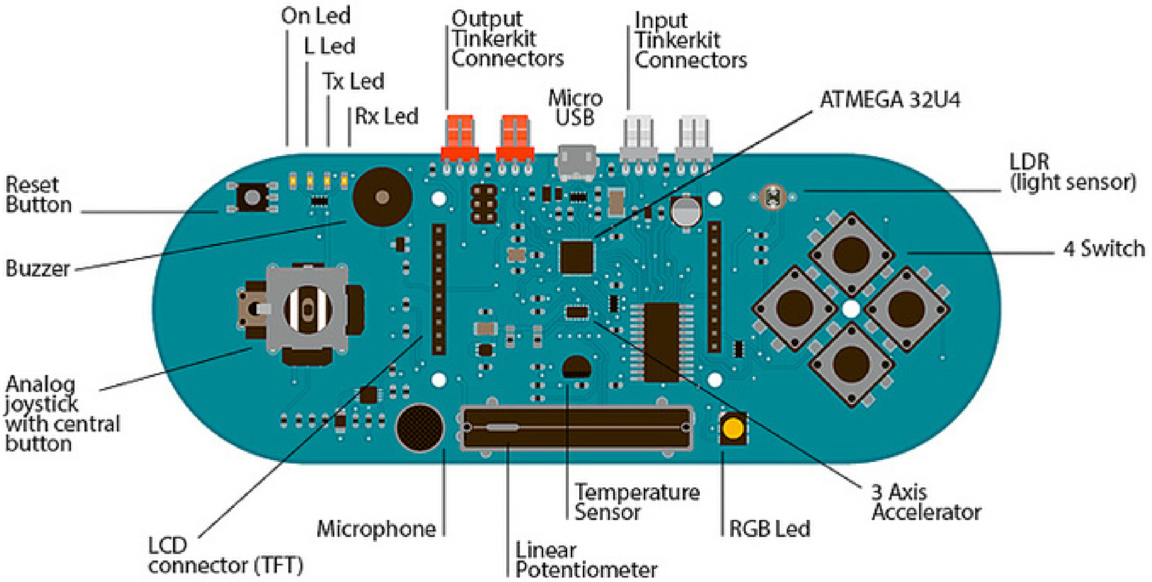
The board contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable to get started.
The Esplora has built-in USB communication; it can appear to a connected computer as a mouse or keyboard, in addition to a virtual (CDC) serial / COM port. This has other implications for the behavior of the board; these are detailed on the getting started page (http://arduino.cc/en/Guide/ArduinoEsplora).

Summary

Microcontroller	ATmega32u4
Operating Voltage	5V
Flash Memory	32 KB of which 4 KB used by bootloader
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: Attach:arduino-esplora-reference-design.zip (http://arduino.cc/en/uploads/Main/arduino-esplora-reference-design.zip)
Schematic: Attach:arduino-esplora-schematic.pdf (http://arduino.cc/en/uploads/Main/arduino-esplora-schematic.pdf)



Memory

The ATmega32u4 has 32 KB (with 4 KB used for the bootloader). It also has 2.5 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output:

The design of the Esplora board recalls traditional gamepad design with an analog joystick on the left and four pushbuttons on the right.
The Esplora has the following on-board inputs and outputs :

- **Analog joystick with central push-button** two axis (X and Y) and a center pushbutton.
- **4 push-buttons** laid out in a diamond pattern.
- **Linear potentiometer** slider near the bottom of the board.
- **Microphone** for getting the loudness (amplitude) of the surrounding environment.
- **Light sensor** for getting the brightness.
- **Temperature sensor** reads the ambient temperature
- **Three-axis accelerometer** measures the board's relation to gravity on three axes (X, Y, and Z)
- **Buzzer** can produce square-waves.
- **RGB led** bright LED with Red Green and Blue elements for color mixing.
- **2 TinkerKit Inputs** to connect the TinkerKit sensor modules with the 3-pin connectors.
- **2 TinkerKit Outputs** to connect the TinkerKit actuator modules with the 3-pin connectors.
- **TFT display connector** connector for an optional color LCD screen, SD card, or other devices that use the SPI protocol.

In order to utilize the total number of available sensors, the board uses an analog multiplexer. This means a single analog input of the microcontroller is shared among all the input channels (except the 3-axis accelerometer). Four additional microcontroller pins choose which channel to read.

Communication

The Leonardo the Esplora has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega32U4 provides serial (CDC) communication over USB and appears as a virtual com port to software on the computer. The chip also acts as a full speed USB 2.0 device, using standard USB COM drivers. On Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB connection to the computer.
The ATmega32U4 also supports SPI communication, that can be accessed through the SPI library.
The Esplora can appear as a generic keyboard and mouse, and can be programmed to control these input devices using the Keyboard and Mouse (http://arduino.cc/en/Reference/MouseKeyboard) libraries.

Programming

The Esplora can be programmed with the Arduino software (download). Select "Arduino Esplora" from the Tools > Board menu. For details, see the getting started page (http://arduino.cc/en/Guide/ArduinoEsplora).
The ATmega32U4 on the Arduino Esplora comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the AVR109 protocol.
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions (http://arduino.cc/en/Hacking/Programmer) for details.

Esplora Library

To facilitate writing sketches for the Esplora, there is a dedicated library that contains methods for reading the sensors and writing to the outputs on-board.
The library offers high level methods which provide pre-processed data, like degrees Fahrenheit or Celsius from the temperature sensor. It also enables easy access to the outputs, like writing values to the RGB LED.
Visit the Esplora library (http://arduino.cc/en/Reference/EsploraLibrary) reference page to see the complete documentation of the library and examples.

Automatic (Software) Reset and Bootloader Initiation

Rather than requiring a physical press of the reset button before an upload, the Esplora is designed in a way that allows it to be reset by software running on a connected computer. The reset is triggered when the Esplora's virtual (CDC) serial / COM port is opened at 1200 baud and then closed. When this happens, the processor will reset, breaking the USB connection to the computer (meaning that the virtual serial / COM port will disappear). After the processor resets, the bootloader starts, remaining active for about 8 seconds. The bootloader can also be initiated by pressing the reset button on the Esplora. Note that when the board first powers up, it will jump straight to the user sketch, if present, rather than initiating the bootloader.
Because of the way the Esplora handles reset it's best to let the Arduino software try to initiate the reset before uploading, especially if you are in the habit of pressing the reset button before uploading on other boards. If the software can't reset the board you can always start the bootloader by pressing the reset button on the board.

USB Overcurrent Protection

The Esplora has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Esplora PCB are 6.5 and 2.4 inches respectively, with the USB and TinkerKit connectors extending beyond the latter dimension. Four screw holes allow the board to be attached to a surface or case.

NEWSLETTER

Enter your email to sign up →

(https://twitter.com/arduino) (http://www.facebook.com/official.arduino)

(https://plus.google.com/+Arduino) (http://www.flickr.com/photos/arduino_cc)

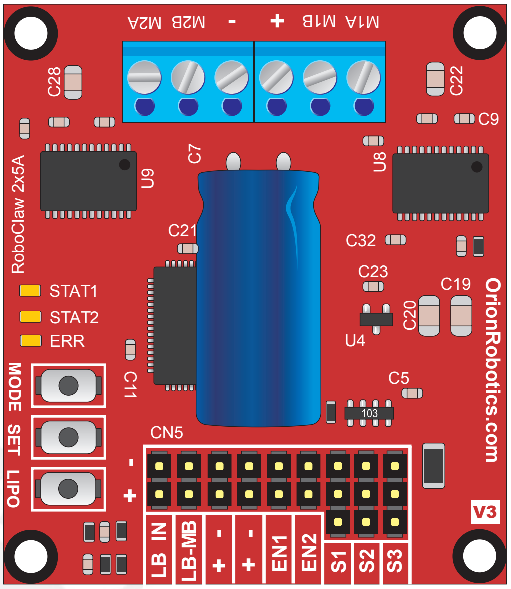
(http://youtube.com/arduinoteam)

Anexo 16: Roboclaw

RoboClaw 2x5A Motor Controller
Data Sheet
Firmware Version 3.1.3+

Feature Overview:

- 2 Channels at 5Amp each, Peak 10Amp
- 3.3V Compliant Outputs
- 5V Tolerant Inputs
- Battery Elimination Circuit (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 2X5 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 5 amps per channel continuous and 10 amp peak.

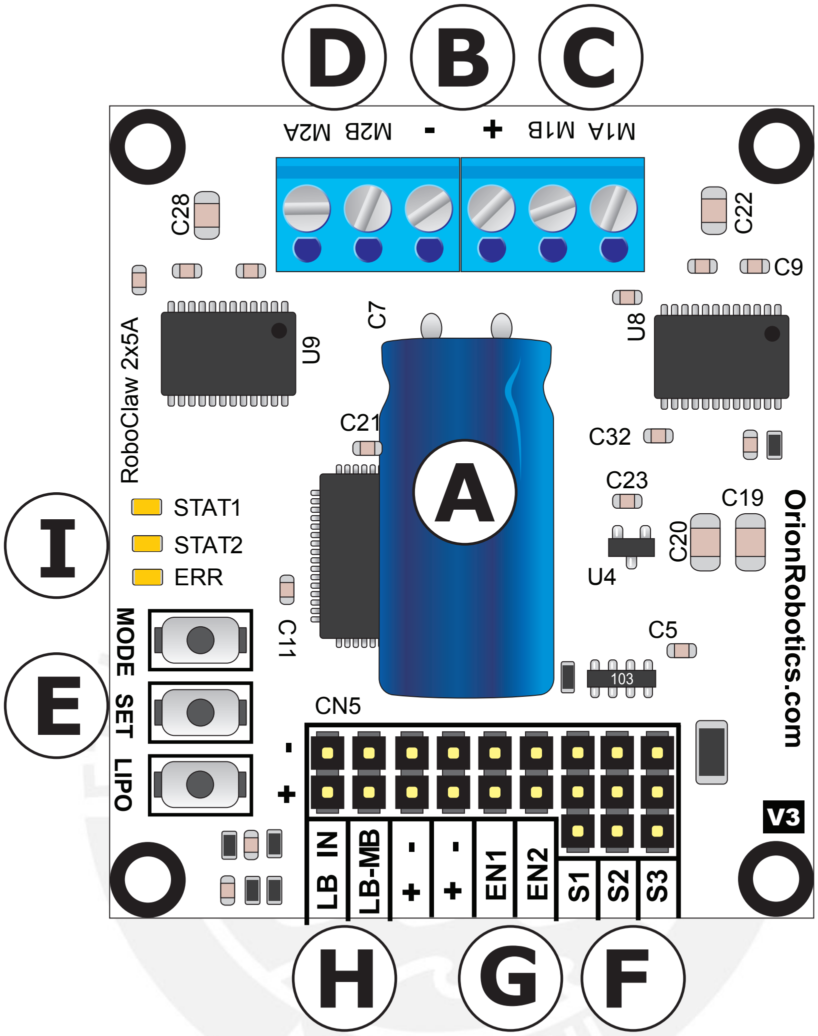
With support for dual quadrature decoding you get greater control over speed and velocity. Automatically maintain a speed even if load increases. RoboClaw uses PID calculations with feed forward in combination with external quadrature encoders to make an accurate control solution.

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.

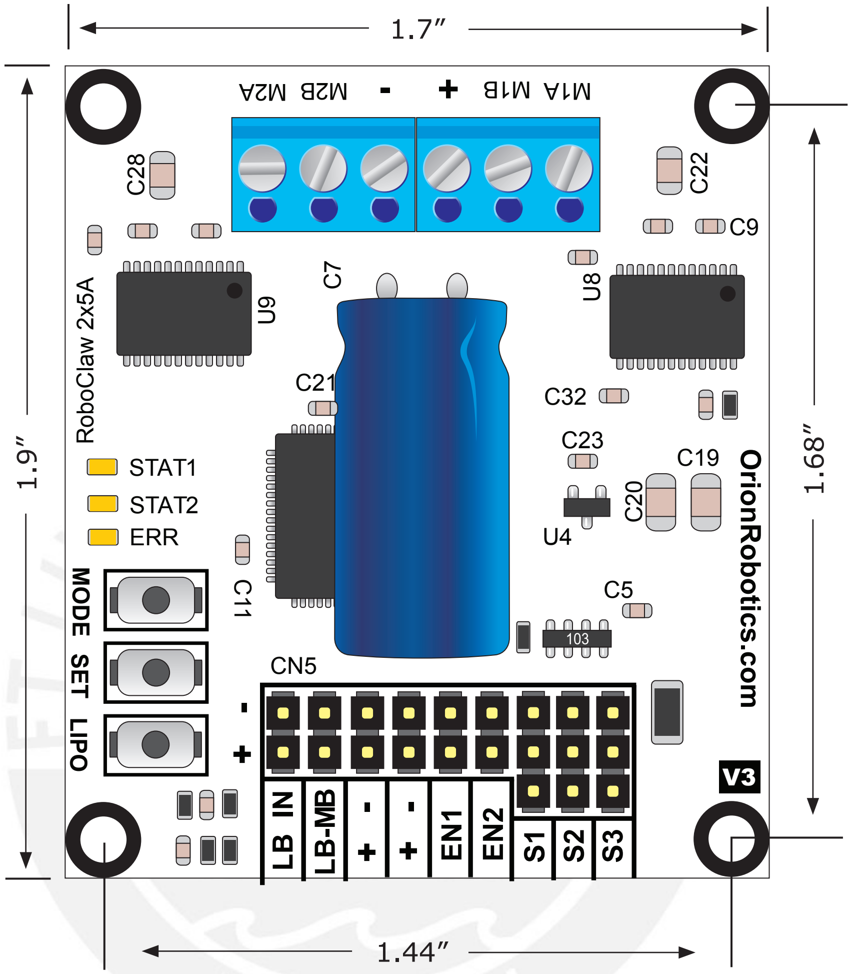
Power System
The RoboClaw is equipped with synchronous regenerative motor drivers. This means your battery is recharged when slowing down, braking or reversing. In addition a BEC circuit is included. It can supply a useful current of up to 150mA at 5v. The BEC is meant to provide power to a microcontroller or RC receiver.

Hardware Overview:



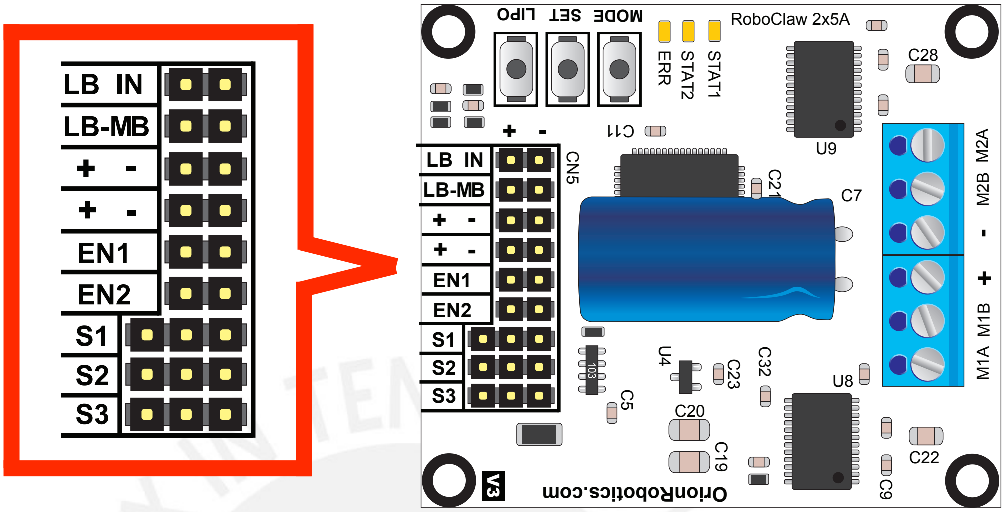
A: Power Stabilizer
B: Main Battery Input
C: Motor Channel 1
D: Motor Channel 2
E: Setup Buttons
F: Control Inputs
G: Encoder Inputs
H: Logic Voltage Source Selection Header
I: Status and Error LED Indicators

Dimensions:



Board Edge: 1.7"W X 1.9"L
Hole Pattern: 0.125D, 1.44"W x 1.68"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.

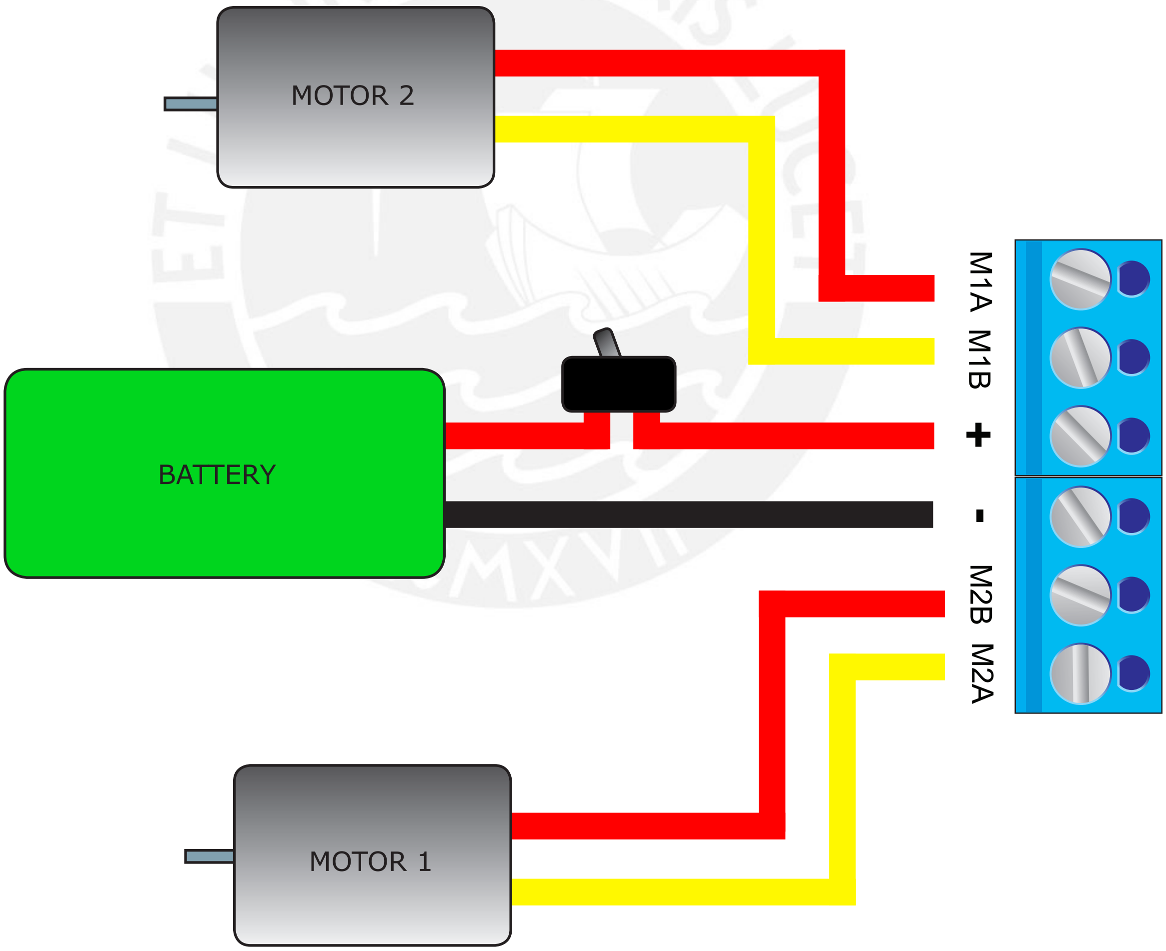
Control Inputs (S1 / S2 / S3)
S1, S2 and S3 are setup for standard servo style headers I/O, +5V and GND. S1 and S2 are the control inputs for serial, analog and RC modes. S3 can be used as a flip switch input when in RC or Analog modes. In serial mode S3 becomes an emergency stop. S3 is active when pulled 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.

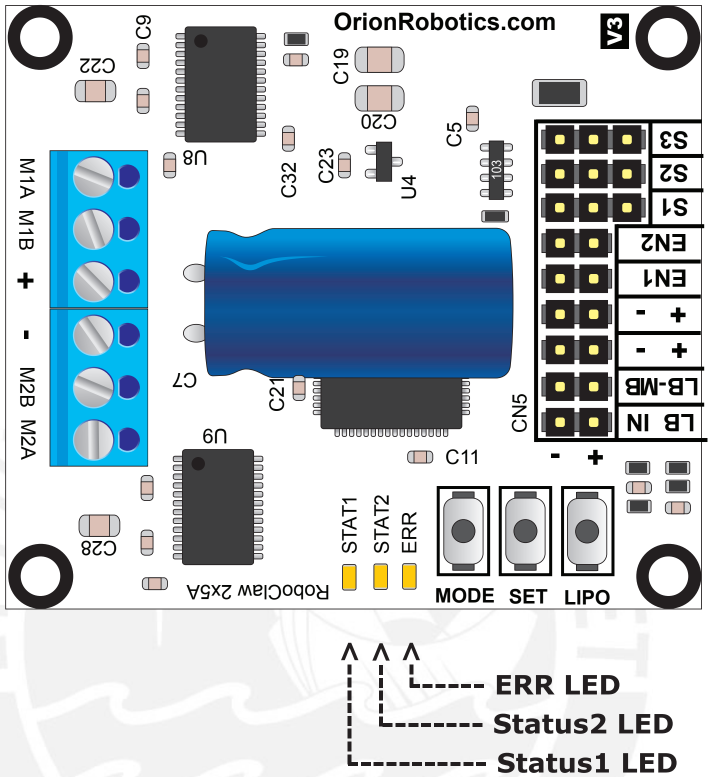
Motor Screw Terminals

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.



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.

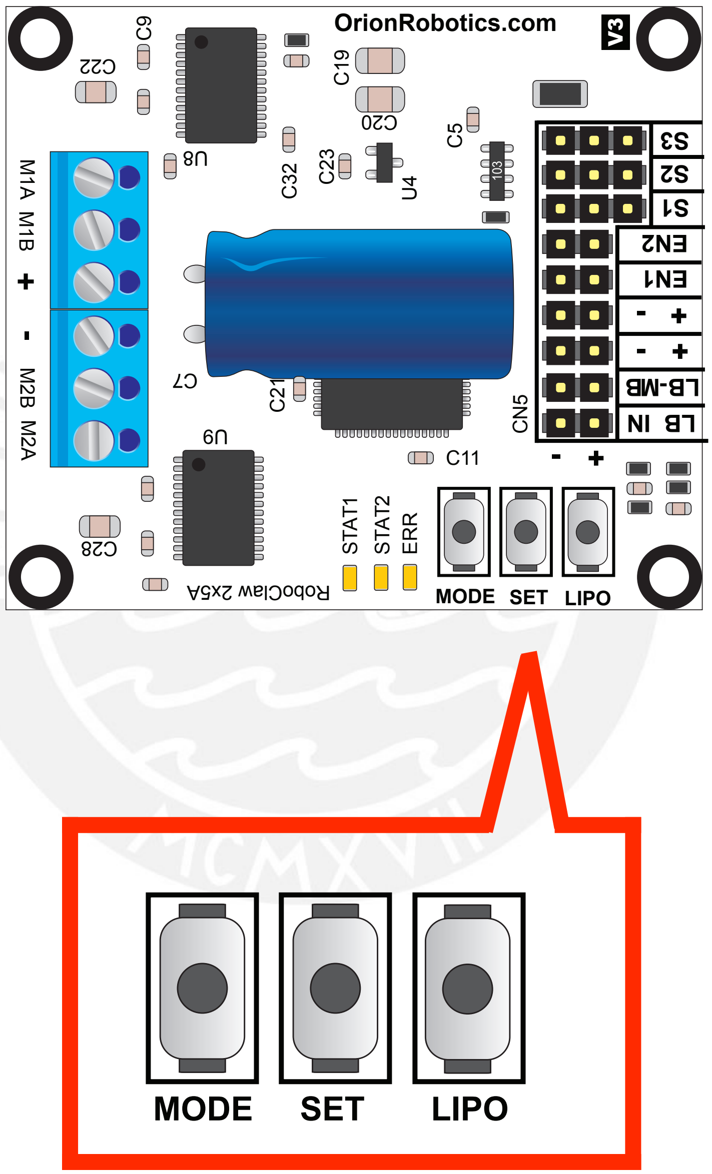
Serial Modes
Status 1 LED = On continuous, blink on serial receive.
Status 2 LED = On when motor(s) active.

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)

RC Mode

RC Mode
RC mode is typically used when controlling RoboClaw from a hobby RC radio. This mode can also be used to simplify driving RoboClaw from a microcontroller using servo pulses. In this mode S1 controls the direction and speed of motor 1 and S2 controls the speed and direction of motor 2. This drive method is similar to how a tank is controlled.

Using RC Mode with feedback for velocity/position control
RC Mode can be used with encoders. Packet Serial commands must be used to enable this option. Velocity and/or Position PID constants must be calibrated for proper operation using Packet Serial commands. Once calibrated values have been set and saved into Roboclaws eeprom, encoder support using velocity or position PID control can be enabled using commands 92 for motor 1 or 93 for motor 2. See the Packet Serial section for more details.

RC Mode With Mixing
This mode is the same as RC mode with the exception of how S1 and S2 control the attached motors. S1 controls speed and direction of both motors 1 and 2. S2 controls steering by slowing one of the motors. This drive method is similar to how a car would be controlled.

RC Mode Options

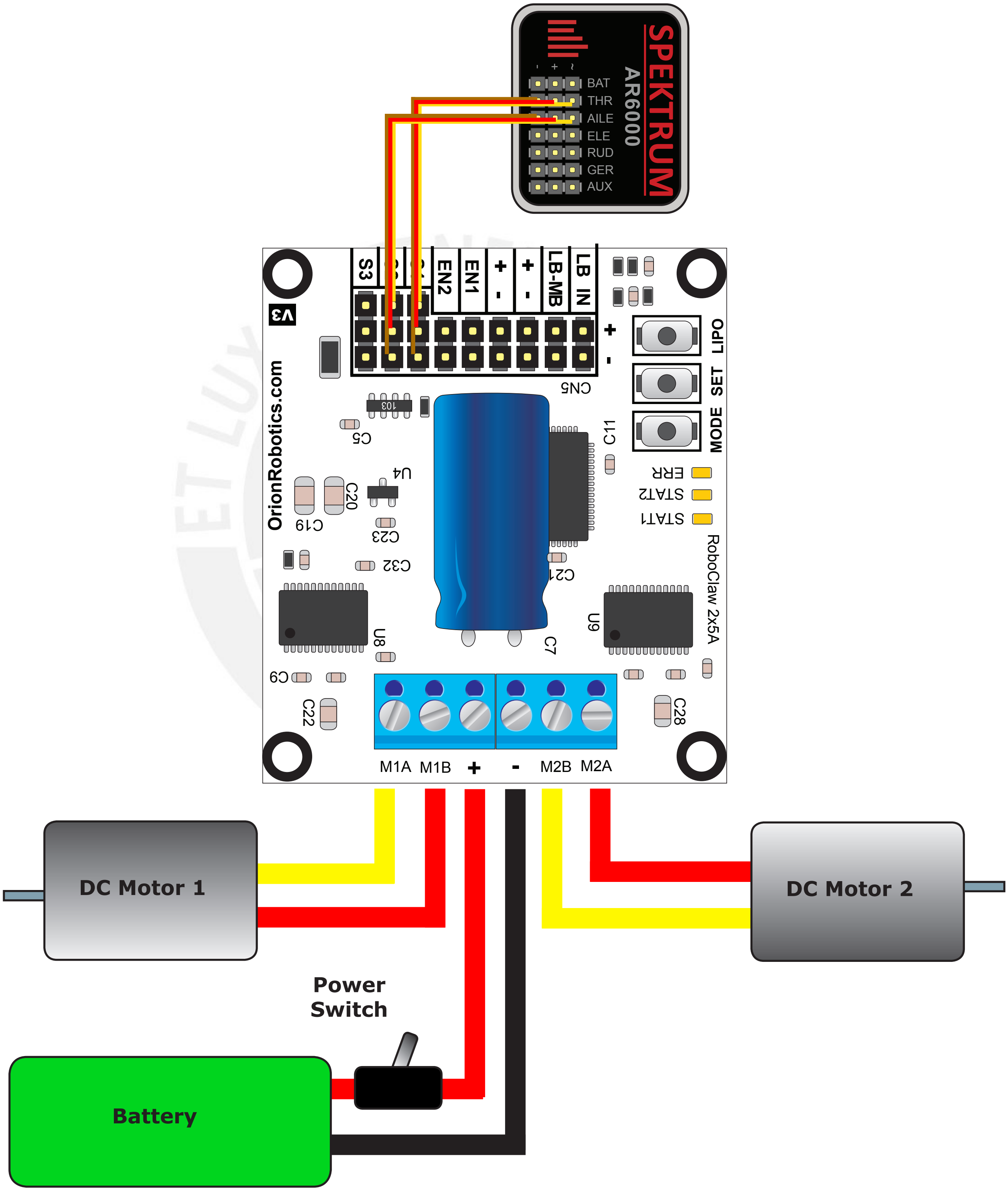
Option	Function	Description
1	TTL Flip Switch	Flip switch triggered by low signal.
2	TTL Flip and Exponential Enabled	Softens the center control position. This mode is ideal with tank style robots. Making it easier to control from an RC radio. Flip switch triggered by low signal.
3	TTL Flip and MCU Enabled	Continues to execute last pulse received until new pulse received. Disables Signal loss fail safe and auto calibration. Flip switch triggered by low signal.
4	TTL Flip and Exponential and MCU Enabled	Enables both options. Flip switch triggered by low signal.
5	RC Flip Switch Enabled	Same as mode 1 with flip switch triggered by RC signal.
6	RC Flip and Exponential Enabled	Same as mode 2 with flip switch triggered by RC signal.
7	RC Flip and MCU Enabled	Same as mode 3 with flip switch triggered by RC signal.
8	RC Flip and Exponential and MCU Enabled	Same as mode 4 with flip switch triggered by RC signal.

Servo Pulse Ranges
The RoboClaw expects RC servo pulses on S1 and S2 to drive the motors when the mode is set to RC mode. The center points are calibrated at start up. 1000us is the default for full reverse and 2000us is the default for full forward. The RoboClaw will auto calibrate these ranges on the fly unless auto-calibration is disabled. If a pulse smaller than 1000us or larger than 2000us is detected the new pulses will be set as the new ranges.

Pulse	Function
1000us	Full Reverse
2000us	Full Forward

RC Wiring Example

Connect the RoboClaw as shown below. Set mode 1 with option 1. The configuration below uses a separate logic battery so remove the MB-LB jumper. Before powering up anything center the control sticks then turn the radio on first, then the receiver, then RoboClaw. It will take RoboClaw about 1 second to calibrate the neutral position.



RC Control - Arduino Example
The example will drive a 2 motor 4 wheel robot in reverse, stop, forward, left turn and then right turn. The program was written and tested with a Arduino Uno and P5 connected to S1, P6 connected to S2. Set mode 2 with option 4.

```
//Basic Micro RoboClaw RC Mode. Control RoboClaw
//with servo pulses from a microcontroller.
//Mode settings: Mode 2 with Option 4.


#include <Servo.h>

Servo myservo1;  // create servo object to control a RoboClaw channel
Servo myservo2;  // create servo object to control a RoboClaw channel

int pos = 0;    // variable to store the servo position

void setup()
{
  myservo1.attach(5);  // attaches the RC signal on pin 5 to the servo object
  myservo2.attach(6);  // attaches the RC signal on pin 6 to the servo object
}


void loop()
{
  myservo1.writeMicroseconds(1500);  //Stop
  myservo2.writeMicroseconds(1500);  //Stop
  delay(2000);

  myservo1.writeMicroseconds(1250);  //full forward
  delay(1000);

  myservo1.writeMicroseconds(1500);  //stop
  delay(2000);

  myservo1.writeMicroseconds(1750);  //full reverse
  delay(1000);

  myservo1.writeMicroseconds(1500);  //Stop
  delay(2000);

  myservo2.writeMicroseconds(1250);  //full forward
  delay(1000);

  myservo2.writeMicroseconds(1500);  //Stop
  delay(2000);

  myservo2.writeMicroseconds(1750);  //full reverse
  delay(1000);
}
```

Analog Mode

Analog Mode
Analog mode is used when controlling RoboClaw from a potentiometer or a filtered PWM signal. In this mode S1 and S2 are set as analog inputs. Voltage range is 0V = Full reverse, 1V = Stop and 2V = Full forward.

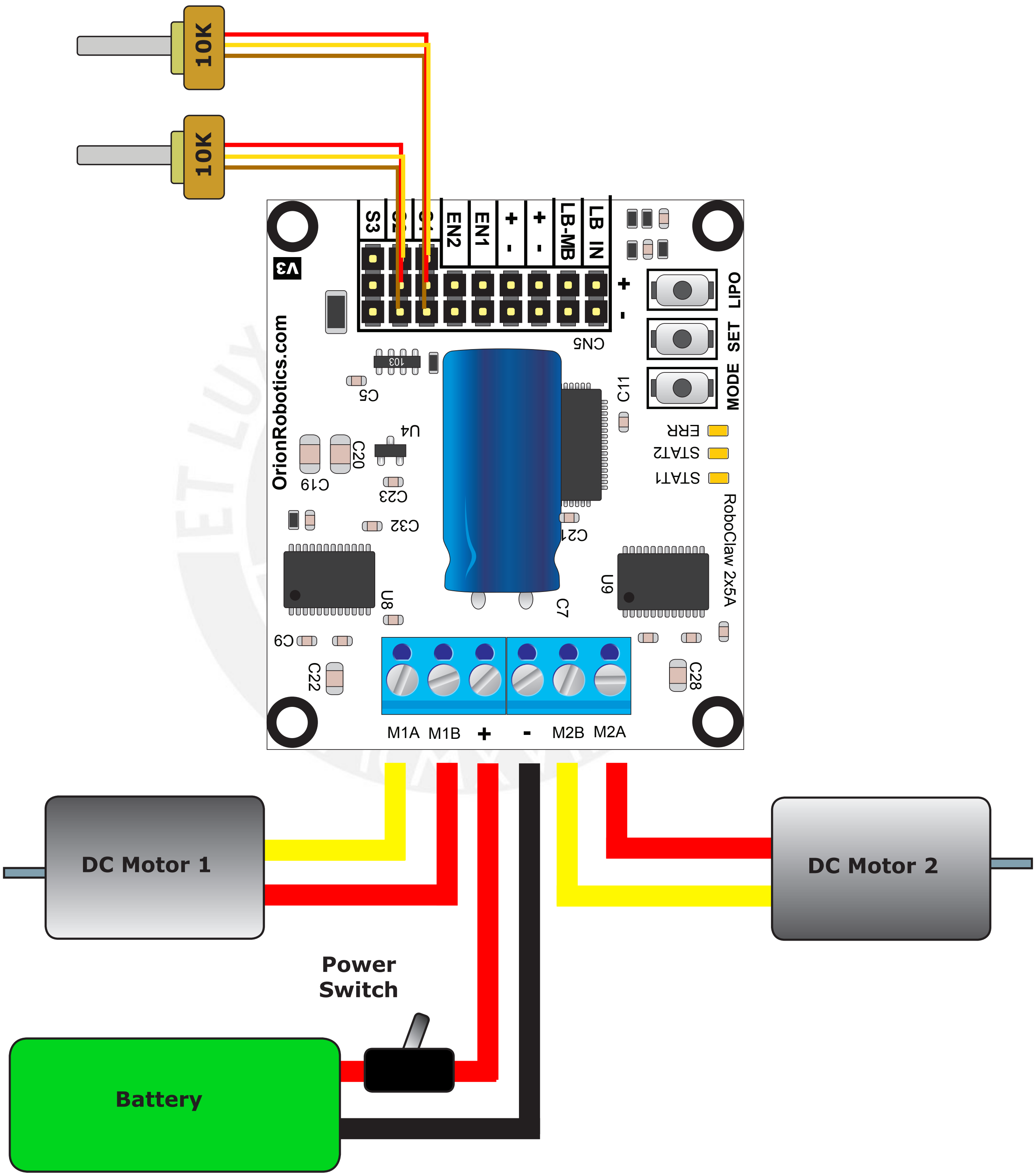
Using Analog Mode with feedback for velocity/position control
Analog Mode can be used with encoders. Packet Serial commands must be used to enable this option. Velocity and/or Position PID constants must be calibrated for proper operation using Packet Serial commands. Once calibrated values have been set and saved into Roboclaws eeprom, encoder support using velocity or position PID control can be enabled using commands 92 for motor 1 or 93 for motor 2. See the Packet Serial section for more details.

Analog Mode With Mixing
This mode is the same as Analog mode with the exception of how S1 and S2 control the attached motors. S1 controls speed and direction of both motors 1 and 2. S2 controls steering by slowing one of the motors. This drive method is similar to how a car would be controlled.

Analog Mode Options

Option	Function	Description
1	TTL Flip Switch	Flip switch triggered by low signal.
2	TTL Flip and Exponential Enabled	Softens the center control position. This mode is ideal with tank style robots. Making it easier to control from an RC radio. Flip switch triggered by low signal.
3	TTL FLip and MCU Enabled	Continues to execute last pulse received until new pulse received. Disables Signal loss fail safe and auto calibration. Flip switch triggered by low signal.
4	TTL FLip and Exponential and MCU Enabled	Enables both options. Flip switch triggered by low signal.
5	RC Flip Switch Enabled	Same as mode 1 with flip switch triggered by RC signal.
6	RC Flip and Exponential Enabled	Same as mode 2 with flip switch triggered by RC signal.
7	RC Flip and MCU Enabled	Same as mode 3 with flip switch triggered by RC signal.
8	RC Flip and Exponential and MCU Enabled	Same as mode 4 with flip switch triggered by RC signal.

Analog Wiring Example
Connect the RoboClaw as shown below using two potentiometers. Set mode 3 with option 1. Center the potentiometers before applying power or the attached motors will start moving. S1 potentiometer will control motor 1 direction and speed. S2 potentiometer will control motor 2 direction and speed.



Simple Serial

Simple Serial Mode

In this mode S1 accepts TTL level byte commands. Simple serial mode is one way serial data. RoboClaw can receive only. A standard 8N1 format is used. Which is 8 bits, no parity bits and 1 stop bit. If you are using a microcontroller you can interface directly to RoboClaw. If you are using a PC a level shifting circuit (See Max232) is required. The baud rate can be changed using the SET button once a serial mode has been selected.

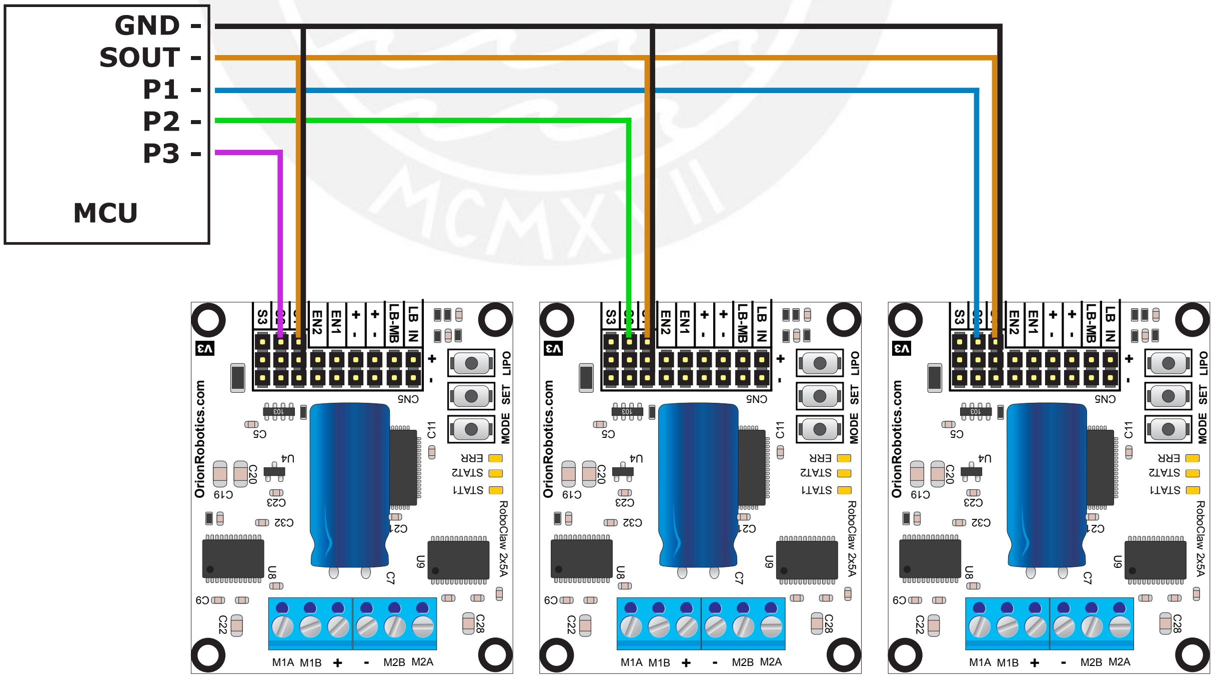
Serial Mode Options

Option	Description
1	2400
2	9600
3	19200
4	38400

Simple Serial Mode With Slave Select

Slave select is used when more than one RoboClaw is on the same serial bus. When slave select is set to ON the S2 pin becomes the select pin. Set S2 high (5V) and RoboClaw will execute the next set of commands sent to its S1 pin. Set S2 low (0V) and RoboClaw will ignore all received commands.

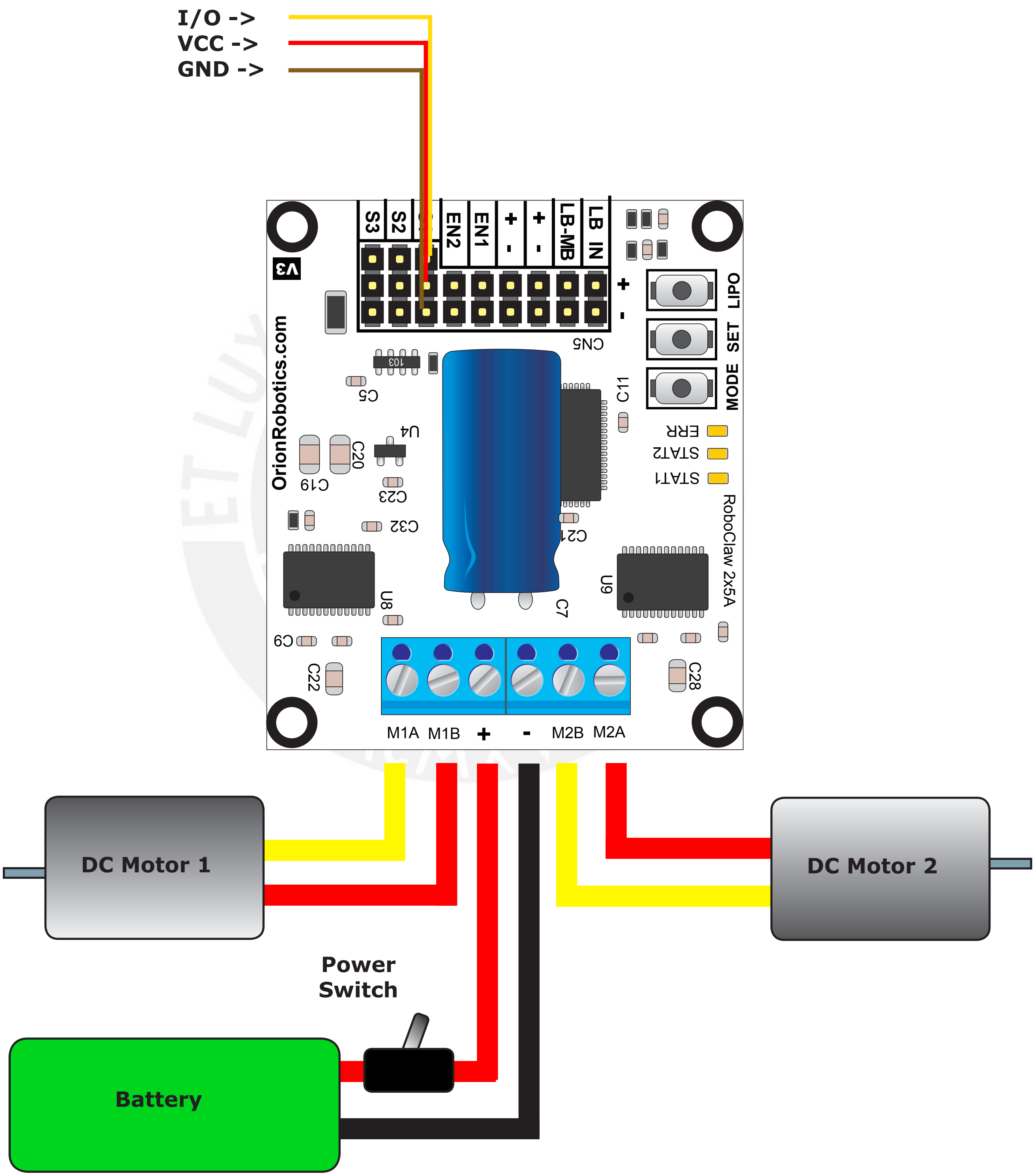
To setup up RoboClaw for serial slave make sure all RoboClaws share a common signal ground (GND) shown by the black wire. SOUT (Brown line) is connected to the S1 pin of all three RoboClaws which is the serial in of the RoboClaw. P1, P2 and P3 are connected to individual S2 pins. Only one MCU pin is connected to each RoboClaw S2 pin. To enable a RoboClaw hold its S2 pin high otherwise commands will be ignored.



Simple Serial Command Syntax
The RoboClaw simple serial is setup to control both motors with one byte sized command character. Since a byte can be anything from 0 to 255 the control of each motor is split. 1 to 127 controls channel 1 and 128 to 255 controls channel 2. Command character 0 will shut down both channels. Any other values will control speed and direction of the specific channel.

Character	Function
0	Shuts Down Channel 1 and 2
1	Channel 1 - Full Reverse
64	Channel 1 - Stop
127	Channel 1 - Full Forward
128	Channel 2 - Full Reverse
192	Channel 2 - Stop
255	Channel 2 - Full Forward

Simple Serial Wiring Example
In simple serial mode the RoboClaw can only receive serial data. Use the below wiring diagram with the following code examples. Make sure the LB-MB jumper is installed.



Simple Serial - Arduino Example
The following example will start both channels in reverse, then full speed forward. The program was written and tested with a Arduino Uno and Pin 5 connected to S1. Set mode 5 and option 3.

```
//Basic Micro RoboClaw Simple Serial Test
//Switch settings: SW2=ON and SW5=ON
//Make sure Arduino and Robo Claw share common GND!

#include “BMSerial.h”

BMSerial mySerial(5,6);

void setup() {
  mySerial.begin(19200);
}

void loop() {
  mySerial.write(1);
  mySerial.write(-1);
  delay(2000);
  mySerial.write(127);
  mySerial.write(-127);
  delay(2000);
}
```

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:

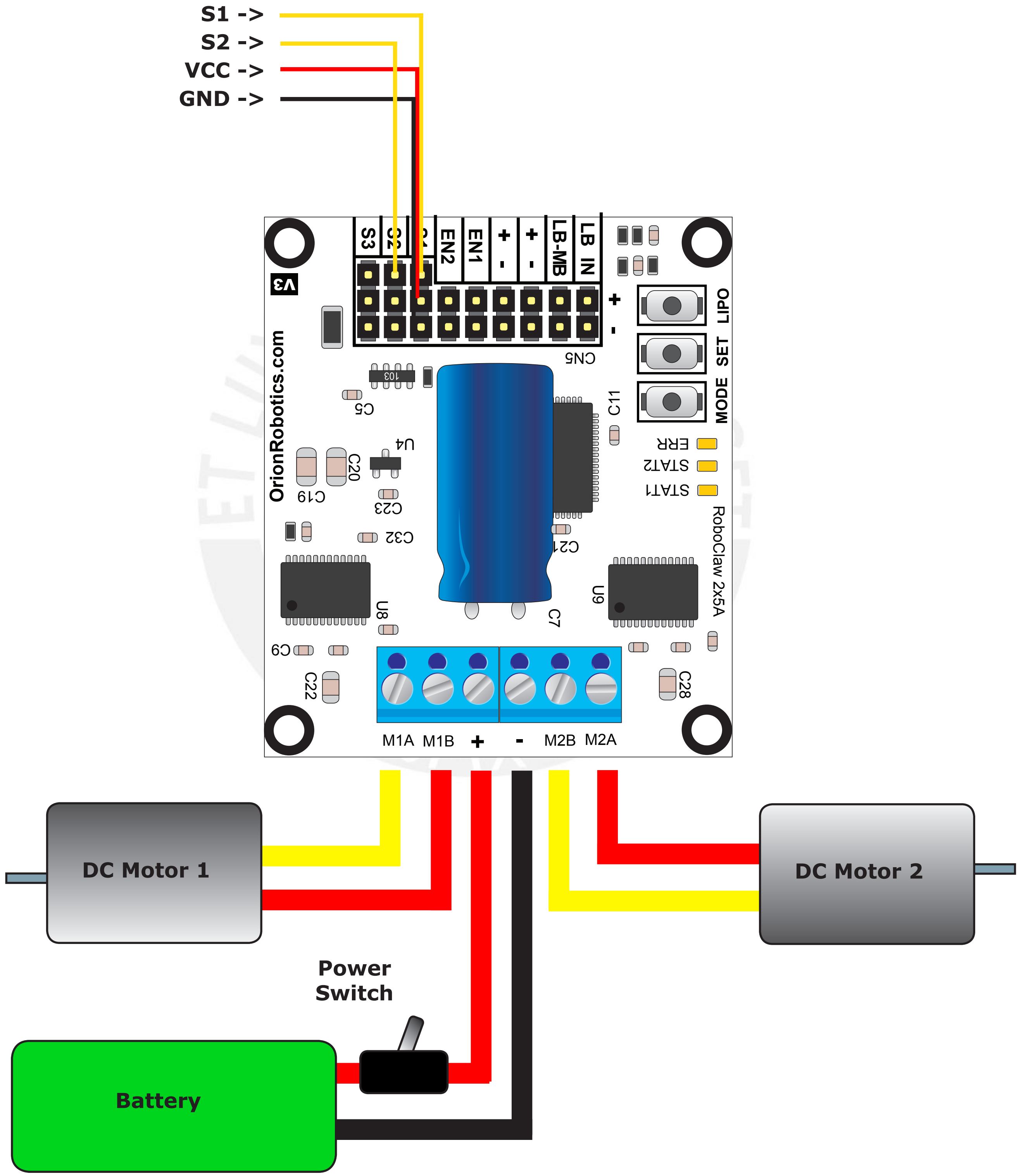
```
Send: 128, 12, 96, ((128+12=96) & 0x7F)
```

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 P, I, D and QPPS Settings
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 P, I, D and QPPS Settings
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)

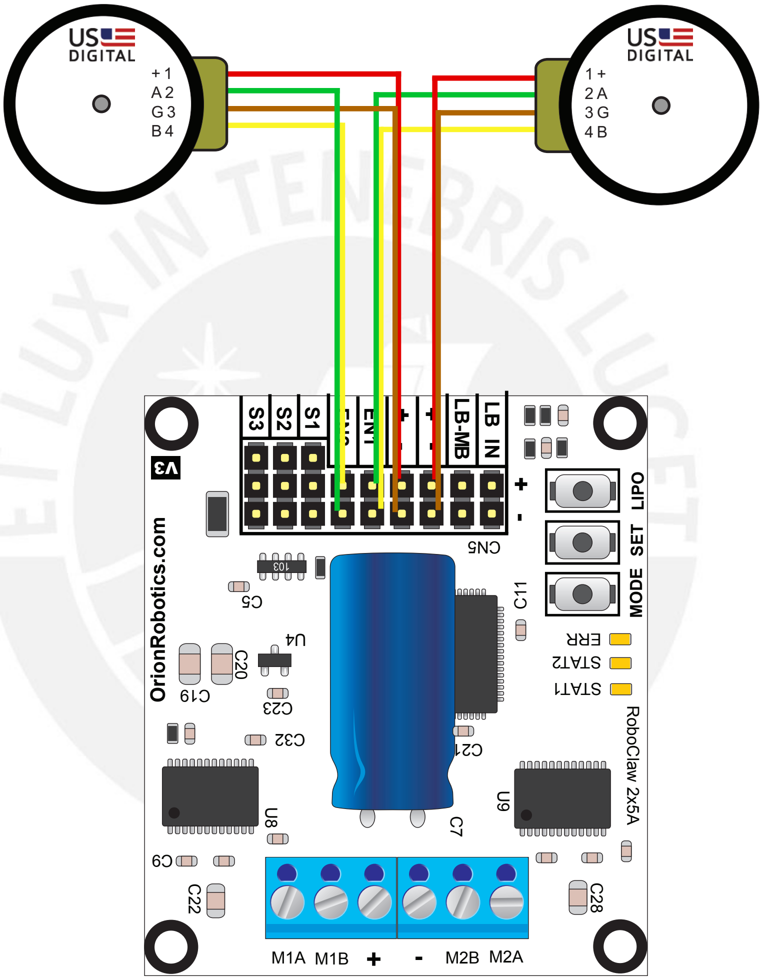
94 - Write Settings to EEPROM
Writes all settings to non-volatile memory. Command syntax:

```
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 PID Constants for M1.
29	Set 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]
```

USB Information

USB RoboClaw Power
The USB RoboClaw is self powered. Which means it is not powered from the USB cable. The USB RoboClaw must be externally powered to function correctly.

USB RoboClaw Connection
The USB RoboClaw should have its USB cable connected before powering it up unless USB mode is specifically set (mode 15). If the master controller (the PC) is powered up the USB RoboClaw will automatically detect it is connected to a powered USB master and will enter USB mode. In some cases it may be necessairy to set USB mode manually by setting RoboClaw to mode 15.

USB Comport and baudrate
The USB RoboClaw will be detected as a CDC Virtual Comport. When connected to a Windows PC a driver must be installed. The driver is available for download. On Linux or OSX the RoboClaw will be automatically detected as a virtual comport and an appropriate driver will automatically be loaded.

Unlike a real Comport the USB CDC Virtual Comport does not need a baud rate to be set. It will always communicate at the fastest speed the master and slave device can reach. This will typically be 1mbit/s.

Reading Quadrature Encoder - Arduino Example

The example was tested with an Arduino Uno. RoboClaw was connected as shown in both packet serial wiring and quadrature encoder wiring diagrams.

The example will read the speed, total ticks and direction of each encoder. Connect to the program using a terminal window set to 38400 baud. The program will display the values of each encoders current count along with each encoder status bit in binary and the direction bit. As the encoder is turned it will update the screen.

```
//Basic Micro RoboClaw Packet Serial Mode.
//Switch settings: SW3=ON, SW4=ON, SW5=ON

#include "BMSerial.h"
#include "RoboClaw.h"

#define address 0x80

#define Kp 0x00010000
#define Ki 0x00008000
#define Kd 0x00004000
#define qpps 44000

BMSerial terminal(0,1);
RoboClaw roboclaw(5,6);

void setup() {
  terminal.begin(38400);
  roboclaw.begin(38400);

  roboclaw.SetM1Constants(address,Kd,Kp,Ki,qpps);
  roboclaw.SetM2Constants(address,Kd,Kp,Ki,qpps);
}

void loop() {
  uint8_t status;
  bool valid;

  uint32_t enc1= roboclaw.ReadEncM1(address, &status, &valid);
  if(valid){
    terminal.print("Encoder1:");
    terminal.print(enc1,HEX);
    terminal.print(" ");
    terminal.print(status,HEX);
    terminal.print(" ");
  }
  uint32_t enc2 = roboclaw.ReadEncM2(address, &status, &valid);
  if(valid){
    terminal.print("Encoder2:");
    terminal.print(enc2,HEX);
    terminal.print(" ");
    terminal.print(status,HEX);
    terminal.print(" ");
  }
  uint32_t speed1 = roboclaw.ReadSpeedM1(address, &status, &valid);
  if(valid){
    terminal.print("Speed1:");
    terminal.print(speed1,HEX);
    terminal.print(" ");
  }
  uint32_t speed2 = roboclaw.ReadSpeedM2(address, &status, &valid);
  if(valid){
    terminal.print("Speed2:");
    terminal.print(speed2,HEX);
    terminal.print(" ");
  }
  terminal.println();

  delay(100);
}
```

Speed Controlled by Quadrature Encoders - Arduino Example

The following example was written using an Arduino UNO. RoboClaw was connected as shown in both packet serial wiring and quadrature encoder wiring diagrams.

The example will command a 4wheel robot to move forward, backward, right turn and left turn slowly. You can change the speed by adjusting the value of Speed and Speed2 variables.

```
//Basic Micro RoboClaw Packet Serial Mode.
//Switch settings: SW3=ON, SW4=ON, SW5=ON

#include "BMSerial.h"
#include "RoboClaw.h"

#define address 0x80


#define Kp 0x00010000
#define Ki 0x00008000
#define Kd 0x00004000
#define qpps 44000

BMSerial terminal(0,1);
RoboClaw roboclaw(5,6);

void setup() {
  terminal.begin(38400);
  roboclaw.begin(38400);

  roboclaw.SetM1Constants(address,Kd,Kp,Ki,qpps);
  roboclaw.SetM2Constants(address,Kd,Kp,Ki,qpps);
}

void displayspeed(void)
{
  uint8_t status;
  bool valid;

  uint32_t enc1= roboclaw.ReadEncM1(address, &status, &valid);
  if(valid){
    terminal.print("Encoder1:");
    terminal.print(enc1,DEC);
    terminal.print(" ");
    terminal.print(status,HEX);
    terminal.print(" ");
  }
  uint32_t enc2 = roboclaw.ReadEncM2(address, &status, &valid);
  if(valid){
    terminal.print("Encoder2:");
    terminal.print(enc2,DEC);
    terminal.print(" ");
    terminal.print(status,HEX);
    terminal.print(" ");
  }
```

```
  uint32_t speed1 = roboclaw.ReadSpeedM1(address, &status, &valid);
  if(valid){
    terminal.print("Speed1:");
    terminal.print(speed1,DEC);
    terminal.print(" ");
  }
  uint32_t speed2 = roboclaw.ReadSpeedM2(address, &status, &valid);
  if(valid){
    terminal.print("Speed2:");
    terminal.print(speed2,DEC);
    terminal.print(" ");
  }
  terminal.println();
}

void loop() {
  roboclaw.SpeedAccelDistanceM1(address,12000,12000,48000);
  uint8_t depth1,depth2;
  do{
    displayspeed();
    roboclaw.ReadBuffers(address,depth1,depth2);
  }while(depth1);
  roboclaw.SpeedAccelDistanceM1(address,12000,-12000,48000);
  do{
    displayspeed();
    roboclaw.ReadBuffers(address,depth1,depth2);
  }while(depth1);

}
```

Electrical Characteristics

Characteristic	Rating	Min	Typ	Max
Pulse Per Second	PPS	0		8,000,000
Main Battery (B+ / B-)	VDC	6		34
Logic Battery (LB+ / LB-)	VDC	6	12	34
External Current Draw (BEC)	mA			150
Logic Circuit	mA		30	
Motor Current Per Channel	A		5	10
I/O Input	VDC	0		5
I/O Output	VDC	0		3.3
Analog Voltage Range	VDC	0		2
Tempature Range	C	-40		+125

Warranty
Orion Robotics warranties its products against defects in material and workmanship for a period of 90 days. If a defect is discovered, Orion Robotics will, at our discretion, repair, replace, or refund the purchase price of the product in question. Contact us at support@orionrobotics.com. No returns will be accepted without the proper authorization.

Copyrights and Trademarks
Copyright© 2013 by Orion Robotics, Inc. All rights reserved. RoboClaw and USB RoboClaw are registered trademarks of Basic Micro Inc. Other trademarks mentioned are registered trademarks of their respective holders.

Disclaimer
Orion Robotics cannot be held responsible for any incidental, or consequential damages resulting from use of products manufactured or sold by Orion Robotics or its distributors. No products from Orion Robotics should be used in any medical devices and/or medical situations. No product should be used in a life support situation.

Contacts

Email: sales@orionrobotics.com
Tech support: support@orionrobotics.com
Web: http://www.orionrobotics.com

Discussion List
A web based discussion board is maintained at http://forums.orionrobotics.com.

Technical Support
Technical support is made available by sending an email to support@orionrobotics.com. All email will be answered within 48 hours. All general syntax and programming questions, unless deemed to be a software issue, will be referred to the on-line discussion forums.

Anexo 17: Buffer Tri State 74LS241

MOTOROLA

OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

SN54/74LS240
SN54/74LS241
SN54/74LS244

OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

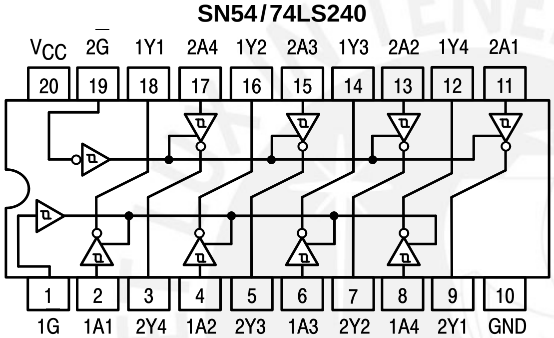
LOW POWER SCHOTTKY

The SN54/74LS240, 241 and 244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

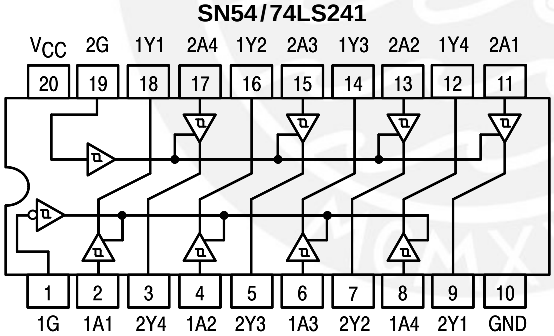
- Hysteresis at Inputs to Improve Noise Margins
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- Input Clamp Diodes Limit High-Speed Termination Effects

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

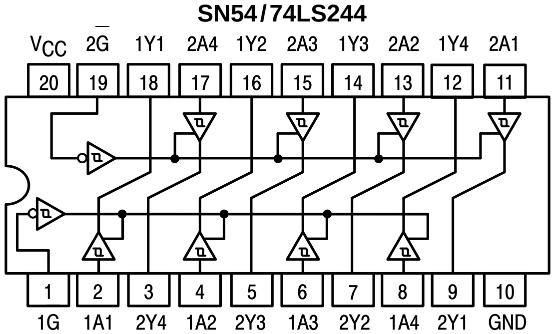
SN54/74LS240

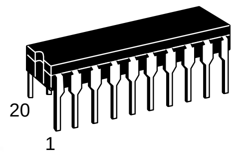


SN54/74LS241

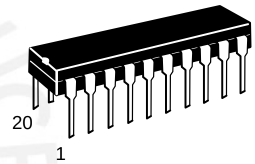


SN54/74LS244

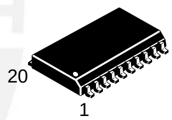




J SUFFIX
CERAMIC
CASE 732-03



N SUFFIX
PLASTIC
CASE 738-03



DW SUFFIX
SOIC
CASE 751D-03

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXDW	SOIC

TRUTH TABLES

SN54/74LS240

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	H
L	H	L
H	X	(Z)

SN54/74LS244

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	L
L	H	H
H	X	(Z)

SN54/74LS241

INPUTS		OUTPUT
1G	D	
L	L	L
L	H	H
H	X	(Z)

INPUTS		OUTPUT
2G	D	
H	L	L
H	H	H
L	X	(Z)

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = HIGH Impedance

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	–55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			–3.0	mA
		54 74			–12 –15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74			0.8		
$V_{T+}–V_{T–}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN
V_{IK}	Input Clamp Diode Voltage			–0.65	–1.5	V	V_{CC} = MIN, I_{IN} = –18 mA
V_{OH}	Output HIGH Voltage	54, 74	2.4	3.4		V	V_{CC} = MIN, I_{OH} = –3.0 mA
		54, 74	2.0			V	V_{CC} = MIN, I_{OH} = MAX
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 12 mA; V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA
I_{OZH}	Output Off Current HIGH				20	µA	V_{CC} = MAX, V_{OUT} = 2.7 V
I_{OZL}	Output Off Current LOW				–20	µA	V_{CC} = MAX, V_{OUT} = 0.4 V
I_{IH}	Input HIGH Current				20	µA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V
I_{IL}	Input LOW Current				–0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Output Short Circuit Current (Note 1)		–40		–225	mA	V_{CC} = MAX
I_{CC}	Power Supply Current Total, Output HIGH				27	mA	V_{CC} = MAX
	Total, Output LOW	LS240			44		
		LS241/244			46		
	Total at HIGH Z	LS240			50		
		LS241/244			54		

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS240		9.0 12	14 18	ns	C_L = 45 pF, R_L = 667 Ω
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS241/244		12 12	18 18	ns	
t_{PZH}	Output Enable Time to HIGH Level		15	23	ns	
t_{PZL}	Output Enable Time to LOW Level		20	30	ns	
t_{PLZ}	Output Disable Time from LOW Level		15	25	ns	C_L = 5.0 pF, R_L = 667 Ω
t_{PHZ}	Output Disable Time from HIGH Level		10	18	ns	

AC WAVEFORMS

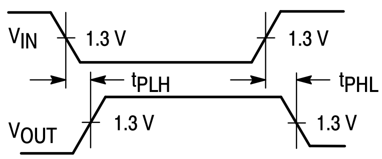


Figure 1

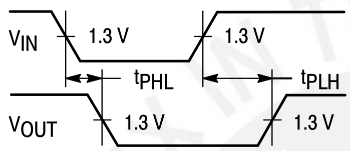


Figure 2

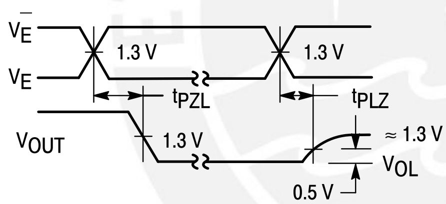


Figure 3

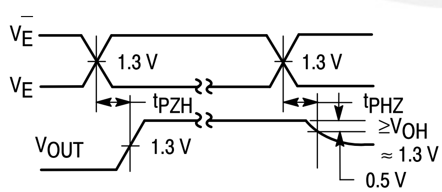
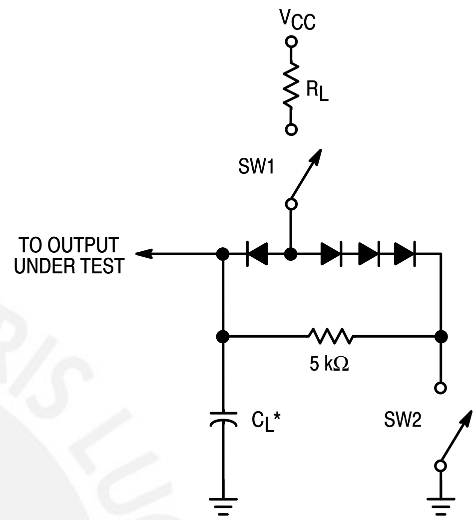


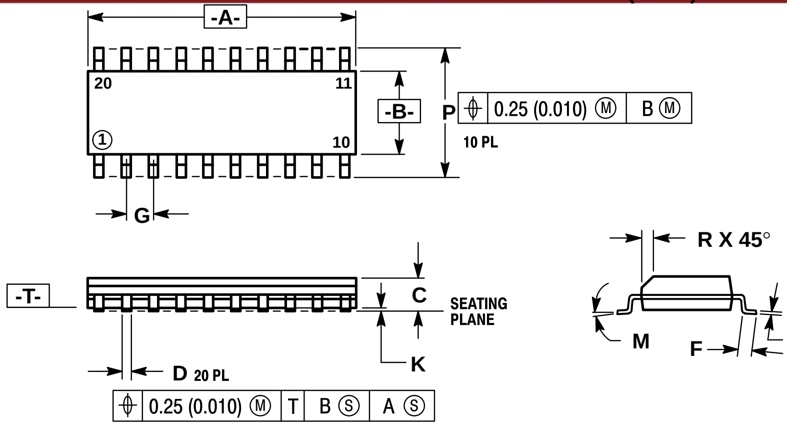
Figure 4



SWITCH POSITIONS

Figure 5

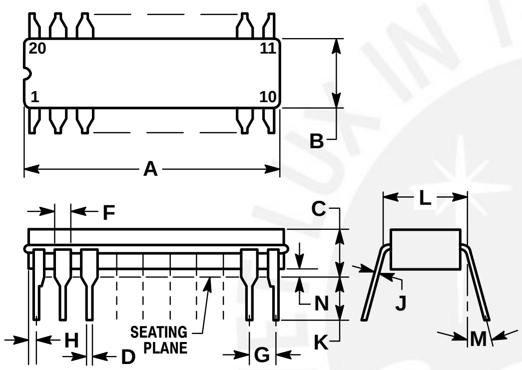
Case 751D-03 DW Suffix
20-Pin Plastic
SO-20 (WIDE)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. 751D-01, AND -02 OBSOLETE, NEW STANDARD 751D-03.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	12.65	12.95	0.499	0.510
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.50	0.90	0.020	0.035
G	1.27 BSC		0.050 BSC	
J	0.25	0.32	0.010	0.012
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

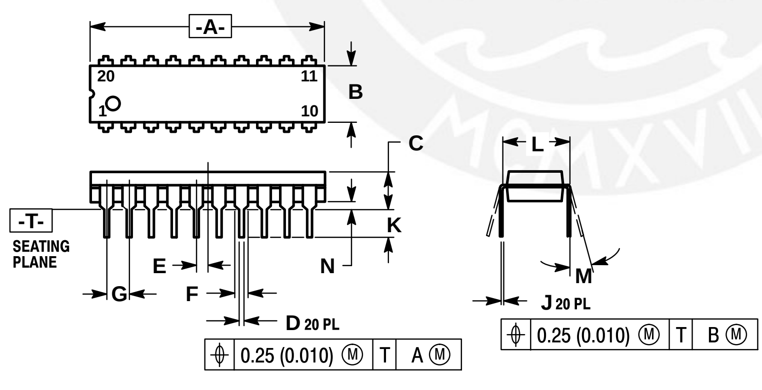
Case 732-03 J Suffix
20-Pin Ceramic Dual In-Line



NOTES:
1. LEADS WITHIN 0.25 mm (0.010) DIA., TRUE POSITION AT SEATING PLANE, AT MAXIMUM MATERIAL CONDITION.
2. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIM A AND B INCLUDES MENISCUS.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	23.88	25.15	0.940	0.990
B	6.60	7.49	0.260	0.295
C	3.81	5.08	0.150	0.200
D	0.38	0.56	0.015	0.022
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.27	0.020	0.050
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.25	1.02	0.010	0.040

Case 738-03 N Suffix
20-Pin Plastic



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION “L” TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION “B” DOES NOT INCLUDE MOLD FLASH.
5. 738-02 OBSOLETE, NEW STANDARD 738-03.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	25.66	27.17	1.010	1.070
B	6.10	6.60	0.240	0.260
C	3.81	4.57	0.150	0.180
D	0.39	0.55	0.015	0.022
E	1.27 BSC		0.050 BSC	
F	1.27	1.77	0.050	0.070
G	2.54 BSC		0.100 BSC	
J	0.21	0.38	0.008	0.015
K	2.80	3.55	0.110	0.140
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.01	0.020	0.040

SYMBOL	SW1	SW2
t_{PZH}	Open	Closed
t_{PZL}	Closed	Open
t_{PLZ}	Closed	Closed

Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

Anexo 18: Matriz de led con Controlador

WEBSITE: sparkfun.com
6175 LONGBOW DRIVE, SUITE 200
BOULDER, COLORADO USA
ZIP CODE: 80301
[303] 284.0979 [GENERAL] P
443.0048 F

RGB Matrix – Serial Backpack User Guide

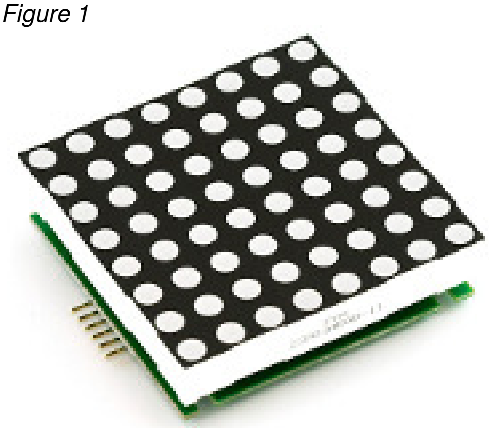
2009.04.20

Overview

The new RGB Matrix – Serial Backpack controller from SparkFun offers an easy way to control the tri-color common cathode 8x8 LED matrices. The entire controller fits behind the LED matrix so that even larger arrays of matrices may be created; external connectors are oriented to allow daisy-chaining of multiple controller boards. The boards accept a special software command to be re-configured for daisy-chained systems. Each backpack is based around an AVR microcontroller and contains all the necessary circuitry to drive the LED matrix.

By default the controller runs a simple frame buffer program that listens for image data and displays in constantly on the LEDs. Data is sent to the backpack using an SPI interface. The microcontroller may also be reprogrammed to give the LED matrix more individual intelligence.

Figure 1



Features

- Runs on 5V
- 64 Tri-Color LEDs
- Input and Output Connectors for daisy-chaining multiple matrices
- 6-pin programming headers for reprogramming the AVR

Powering the RGB Matrix

The controller must be powered with a regulated 5V supply. Power can be supplied either at the SPI Input connector or the SPI output connector as they each share the same power bus. If you are daisy-chaining multiple controllers it is recommended to supply power at the Input connector of the first board in your system, and use the Output connector to supply power to the next board in the system.

RGB Matrix – Serial Backpack User Guide

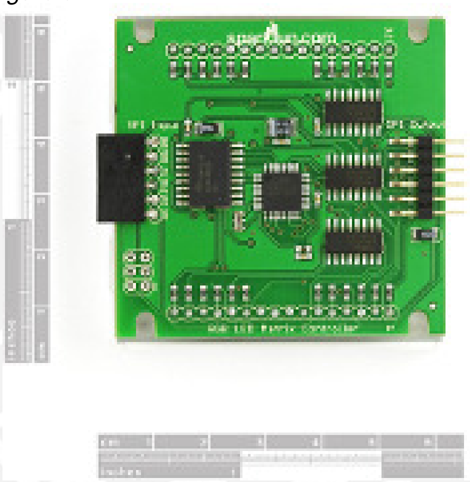
2009.04.20

Figure 2

Parameter	Min.	Recommended	Max	Unit
Voltage	4.50	5.00	5.25	V
Current	-	120(typical)	275	mA

**Current Rating is for a single board. Ratings for multiple boards will be linear (i.e. If 'N' boards are connected in the system, the maximum current draw will be N*Max Current mA.). If your power supply is not capable of providing enough power for the required load, the boards may malfunction.*

Figure 3



There is no on-board voltage regulation so it is recommended to use a known good voltage supply, and to ensure that there won't be voltage fluctuations on the power line. Make sure to properly polarize your power connection according to the indicators on the board before turning the device on.

SPI Interface to the Default Firmware

The RGB matrix backpacks' default program communicates via standard SPI protocol. Data in to the device must be provided via the MOSI pin. SPI Clock must be provided vial the SCLK pin. The device will return data on the MISO pin. All input is ignored while the CS pin is high (5V); and data is copied into the frame buffer while the CS pin is low (0V).

The device maintains a single 64 byte buffer which represents each position in the matrix. When CS is asserted (low) the device begins reading data from the SPI input and writing it sequentially to the 64 byte buffer. Simultaneously the device will output the old buffer data on the MISO line. Hence, to display an image on the matrix a set of 64 bytes must be sequentially transferred to the backpack while keeping the CS pin low (this process is slightly different for a daisy-chained system).

By default, the backpack recognizes up to 255 individual colors. The 64 bytes transferred to the backpack represent the desired color of each LED. The first 3 bits of each byte represent the Red brightness level for that LED; the second 3 bits represent the Green brightness level while the last 2 bits represent the Blue brightness level. Below is a table which illustrates how to construct your color value.

Figure 4

Buffer Byte Representing and LED Color Value							
Red			Green			Blue	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 5

Color(Brightest Setting)	Byte Value
Black	0x00
Red	0xE0
Green	0x1C
Blue	0x03
Orange	0xFC
Magenta	0xE3
Teal	0x1F
White	0xFF

**These values correspond to the brightest levels of the given color. Different variations may be achieved by changing the brightness level of each individual color.*

Example of Sending a Frame of Data

To have an RGB matrix display "black, red, green blue" on the first four positions of the first row, and black everywhere else, this would be the process:

1.) Assert CS (bring it low)

2.) Delay 0.5ms

3.) Transfer 0x00 via SPI

4.) Transfer 0xE0 via SPI

5.) Transfer 0x1C via SPI

6.) Transfer 0x03 via SPI

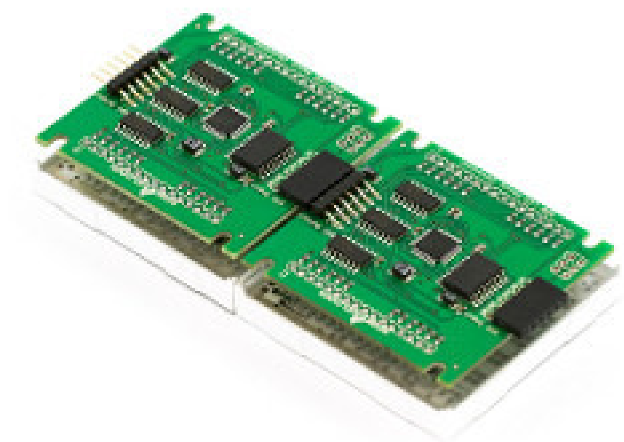
7.) Transfer 0x00 60 times via SPI

8.) De-Assert CS (bring it high)

Daisy-chaining RGB Matrix Controllers

The newest version of the RGB matrix controller has a new layout and a new version of firmware that allow the user to more easily daisy-chain multiple RGB matrix controllers. Daisy-chaining is when multiple controllers are connected together, and the data is passed from the first controller in the "chain" to the next connected controller; data is passed along until the data reaches the end of the chain.

Figure 6



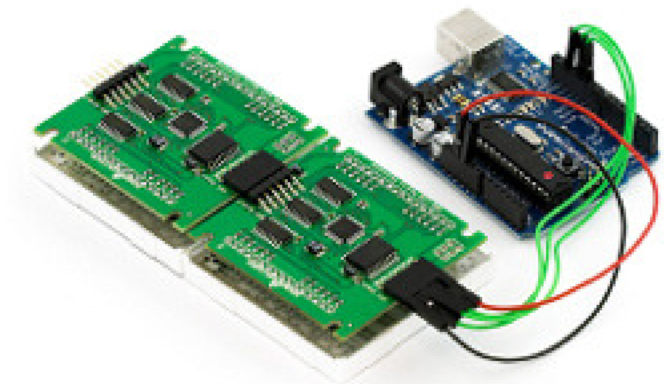
In order to daisy-chain the controllers each controller must be reconfigured for the size of the desired chain. By default the controllers come configured to operate in standalone mode, or a 1 controller chain. If there are two controllers to be daisy-chained together, each of them must be reconfigured for a 2 controller chain. To reconfigure the boards a special two byte command sequence must be sent via the SPI interface: the character '%' followed by the decimal number of the desired chain length. The controllers must be reconfigured individually (so they can't already be connected in the chain). The commands should be sent using the normal SPI protocol. Be sure to assert CS (low) before sending the two command bytes, then de-assert CS (high) when you are finished. Do not start sending the color values to the chain before de-asserting the CS pin. Also, it is recommended to limit the daisy-chain lengths to 8 boards.

When you are connecting the boards, make sure to connect the output of the 1st board in the system to the input of the 2nd board in the system. The connections from your Host system should be plugged into the Input connector of the first controller in the system. Power will be passed through the connectors to all the boards in the system. However, be sure that the power supply can provide enough current for the entire system.

Interfacing to a Daisy-chained System

If the controller boards have been reconfigured to operate in a daisy-chain there are several differences that need to be addressed to display an image on the system. Obviously now that there are more controllers connected there needs to be more data sent. Luckily sending the data to a daisy-chained system is very similar to sending data to a standalone controller. To display an image on a standalone controller all that's done is: assert CS(low), send a 64 byte buffer (frame), de-assert CS. To display an image on a daisy-chained system, this exact process must be repeated for each board in the system. The image won't be displayed until data is received for each controller in the daisy-chained system. A 10ms delay is recommended between de-asserting the CS pin and asserting it before the next frame. The first frame sent to the daisy-chained system will be displayed on the last board in the system, while the last frame sent to the system will be displayed on the first board in the system.

Figure 7



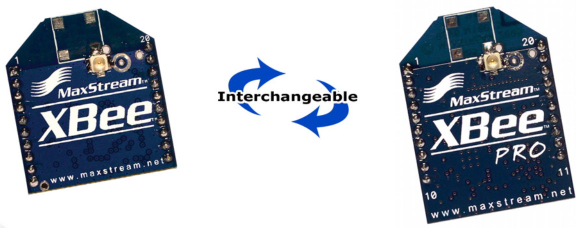
SPI Timing Recommendations:

A delay of 0.5ms is recommended between the assertion of CS and at the start of data transfer, as well as after the end of data transfer and the negation of CS. A 10ms delay is recommended between the CS pulses in a daisy-chained system. The SPI clock should not exceed 125 kHz.

Anexo 19: Módulo Xbee Pro

XBee®/XBee-PRO® RF Modules

- XBee®/XBee-PRO® RF Modules
- RF Module Operation
- RF Module Configuration
- Appendices



Product Manual v1.xEx - 802.15.4 Protocol

For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001

IEEE® 802.15.4 RF Modules by Digi International

Digi International Inc.
11001 Bren Road East
Minnetonka, MN 55343
877 912-3444 or 952 912-3444
http://www.digi.com

90000982_B
2009.09.23

Technical Support:	Phone:	(866) 765-9885 toll-free U.S.A. & Canada (801) 765-9885 Worldwide 8:00 am - 5:00 pm [U.S. Mountain Time]
	Live Chat:	www.digi.com
	Online Support:	http://www.digi.com/support/eservice/login.jsp
	Email:	rf-experts@digi.com

Contents

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.

Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100’ (30 m)
- Outdoor line-of-sight: up to 300’ (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300’ (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements

DSSS (Direct Sequence Spread Spectrum)

Each direct sequence channels has over 65,000 unique network addresses available

Source/Destination Addressing

Unicast & Broadcast Communications

Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 µA

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international varian
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 µA

ADC and I/O line support

Analog-to-digital conversion, Digital I/O

I/O Line Passing

Easy-to-Use

No configuration necessary for out-of box RF communications

Free X-CTU Software
(Testing and configuration software)

AT and API Command Modes for configuring module parameters

Extensive command set

Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements.
Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.

Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 µA	< 10 µA
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85º C (industrial)	-40 to 85º C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Austraila	C-Tick	C-Tick

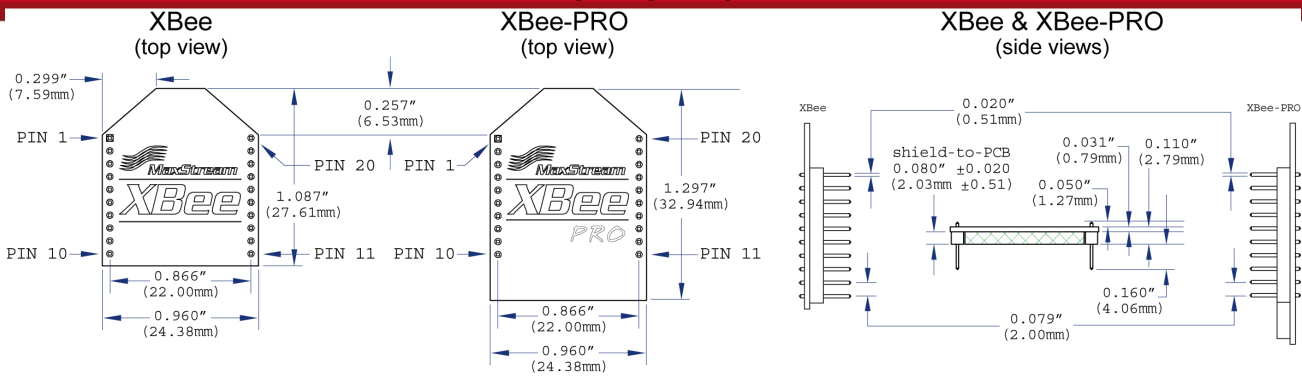
* See Appendix A for region-specific certification requirements.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors.For more information, refer to the "XBee Antennas" Knowledgebase Article located on Digi's Support Web site

Mechanical Drawings

Figure 1-01. Mechanical drawings of the XBee®/XBee-PRO® RF Modules (antenna options not shown)

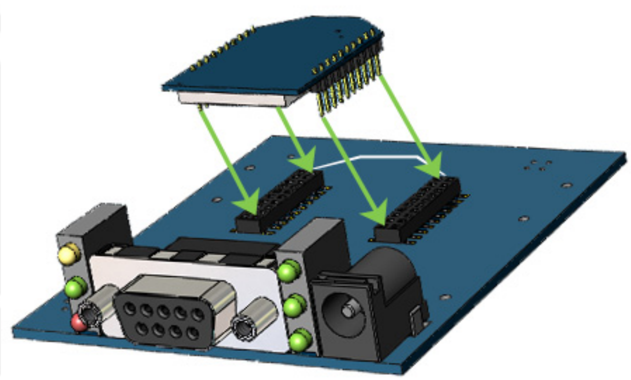
The XBee and XBee-PRO RF Modules are pin-for-pin compatible.



Mounting Considerations

The XBee®/XBee-PRO® RF Module was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-02. XBee Module Mounting to an RS-232 Interface Board.



The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles -
  Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles -
  Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles -
  Samtec P/N: SMM-110-02-SM-S

Digi also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

Pin Signals

Figure 1-03. XBee®/XBee-PRO® RF Module Pin Numbers

(top sides shown - shields on bottom)

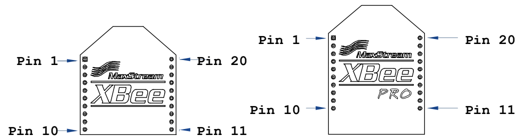


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / **CONFIG**	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	**RESET**	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	**CTS** / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / **SLEEP**	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	**RTS** / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

* Function is not supported at the time of this release

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k Ω pull-up resistor attached to **RESET**
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

Electrical Characteristics

Table 1-03. DC Characteristics (VCC = 2.8 - 3.4 VDC)

Symbol	Characteristic	Condition	Min	Typical		Max	Unit
V_{IL}	Input Low Voltage	All Digital Inputs	-	-		0.35 * VCC	V
V_{IH}	Input High Voltage	All Digital Inputs	0.7 * VCC	-		-	V
V_{OL}	Output Low Voltage	I_{OL} = 2 mA, VCC >= 2.7 V	-	-		0.5	V
V_{OH}	Output High Voltage	I_{OH} = -2 mA, VCC >= 2.7 V	VCC - 0.5	-		-	V
II_{IN}	Input Leakage Current	V_{IN} = VCC or GND, all inputs, per pin	-	0.025		1	µA
II_{OZ}	High Impedance Leakage Current	V_{IN} = VCC or GND, all I/O High-Z, per pin	-	0.025		1	µA
TX	Transmit Current	VCC = 3.3 V	-	45 (XBee)	215, 140 (PRO, Int)	-	mA
RX	Receive Current	VCC = 3.3 V	-	50 (XBee)	55 (PRO)	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10		-	µA

Table 1-04. ADC Characteristics (Operating)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V_{REFH}	VREF - Analog-to-Digital converter reference range		2.08	-	V_{DDAD}*	V
I_{REF}	VREF - Reference Supply Current	Enabled	-	200	-	µA
		Disabled or Sleep Mode	-	< 0.01	0.02	µA
V_{INDC}	Analog Input Voltage[1]		V_{SSAD} - 0.3	-	V_{DDAD} + 0.3	V

1. Maximum electrical operating range, not valid conversion range.

* V_{DDAD} is connected to VCC.

Table 1-05. ADC Timing/Performance Characteristics[1]

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
R_{AS}	Source Impedance at Input[2]		-	-	10	kΩ
V_{AIN}	Analog Input Voltage[3]		V_{REFL}		V_{REFH}	V
RES	Ideal Resolution (1 LSB)[4]	$2.08V \le V_{DDAD} \le 3.6V$	2.031	-	3.516	mV
DNL	Differential Non-linearity[5]		-	±0.5	±1.0	LSB
INL	Integral Non-linearity[6]		-	±0.5	±1.0	LSB
E_{ZS}	Zero-scale Error[7]		-	±0.4	±1.0	LSB
F_{FS}	Full-scale Error[8]		-	±0.4	±1.0	LSB
E_{IL}	Input Leakage Error[9]		-	±0.05	±5.0	LSB
E_{TU}	Total Unadjusted Error[10]		-	±1.1	±2.5	LSB

1. All ACCURACY numbers are based on processor and system being in WAIT state (very little activity and no IO switching) and that adequate low-pass filtering is present on analog input pins (filter with 0.01 µF to 0.1 µF capacitor between analog input and VREFL). Failure to observe these guidelines may result in system or microcontroller noise causing accuracy errors which will vary based on board layout and the type and magnitude of the activity.
Data transmission and reception during data conversion may cause some degradation of these specifications, depending on the number and timing of packets. It is advisable to test the ADCs in your installation if best accuracy is required.
2. R_{AS} is the real portion of the impedance of the network driving the analog input pin. Values greater than this amount may not fully charge the input circuitry of the ATD resulting in accuracy error.
3. Analog input must be between V_{REFL} and V_{REFH} for valid conversion. Values greater than V_{REFH} will convert to $3FF.
4. The resolution is the ideal step size or 1LSB = $(V_{REFH}–V_{REFL})/1024$
5. Differential non-linearity is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to and from the current code.
6. Integral non-linearity is the difference between the transition voltage to the current code and the adjusted ideal transition voltage for the current code. The adjusted ideal transition voltage is $(Current\ Code–1/2)*(1/((V_{REFH}+E_{FS})–(V_{REFL}+E_{ZS})))$.
7. Zero-scale error is the difference between the transition to the first valid code and the ideal transition to that code. The Ideal transition voltage to a given code is $(Code–1/2)*(1/(V_{REFH}–V_{REFL}))$.
8. Full-scale error is the difference between the transition to the last valid code and the ideal transition to that code. The ideal transition voltage to a given code is $(Code–1/2)*(1/(V_{REFH}–V_{REFL}))$.
9. Input leakage error is error due to input leakage across the real portion of the impedance of the network driving the analog pin. Reducing the impedance of the network reduces this error.

10. Total unadjusted error is the difference between the transition voltage to the current code and the ideal straight-line transfer function. This measure of error includes inherent quantization error (1/2LSB) and circuit error (differential, integral, zero-scale, and full-scale) error. The specified value of E_{TU} assumes zero E_{IL} (no leakage or zero real source impedance).

2. RF Module Operation

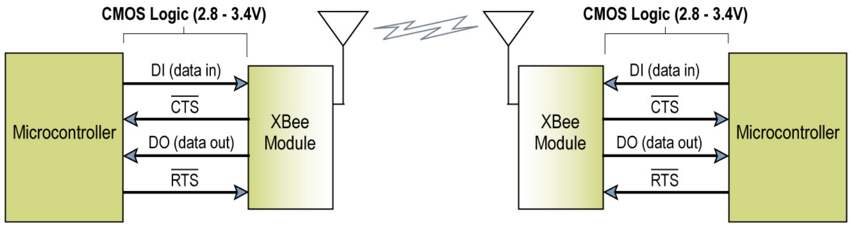
Serial Communications

The XBee®/XBee-PRO® RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 2-01. System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

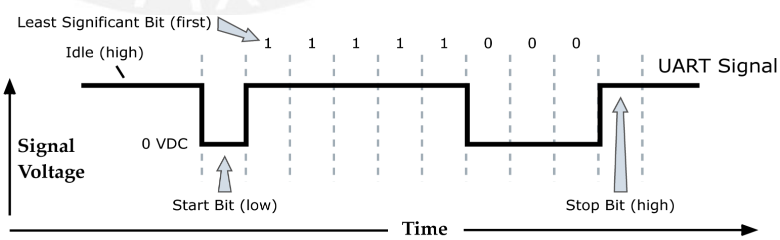


Serial Data

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 2-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)



Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

The UART baud rate and parity settings on the XBee module can be configured with the BD and SB commands, respectively. See the command table in Chapter 3 for details.

Transparent Operation

By default, XBee®/XBee-PRO® RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

Serial-to-RF Packetization

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
2. The maximum number of characters that will fit in an RF packet (100) is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

If the module cannot immediately transmit (for instance, if it is already receiving RF data), the serial data is stored in the DI Buffer. The data is packetized and sent at any RO timeout or when 100 bytes (maximum packet size) are received.

If the DI buffer becomes full, hardware or software flow control must be implemented in order to prevent overflow (loss of data between the host and module).

API Operation

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DI pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets.

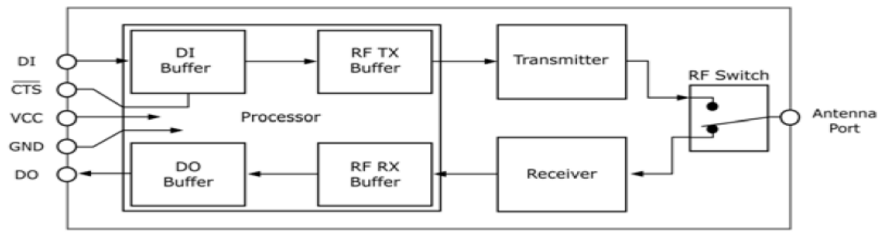
The API operation option facilitates many operations such as the examples cited below:

-> Transmitting data to multiple destinations without entering Command Mode
-> Receive success/failure status of each transmitted RF packet
-> Identify the source address of each received packet

To implement API operations, refer to API sections [p57].

Flow Control

Figure 2-03. Internal Data Flow Diagram



DI (Data In) Buffer

When serial data enters the RF module through the DI pin (pin 3), the data is stored in the DI Buffer until it can be processed.

Hardware Flow Control ($\overline{\text{CTS}}$). When the DI buffer is 17 bytes away from being full; by default, the module de-asserts $\overline{\text{CTS}}$ (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter]. $\overline{\text{CTS}}$ is re-asserted after the DI Buffer has 34 bytes of memory available.

How to eliminate the need for flow control:

1. Send messages that are smaller than the DI buffer size (202 bytes).
2. Interface at a lower baud rate [BD (Interface Data Rate) parameter] than the throughput data rate.

Case in which the DI Buffer may become full and possibly overflow:

If the module is receiving a continuous stream of RF data, any serial data that arrives on the DI pin is placed in the DI Buffer. The data in the DI buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.

Refer to the RO (Packetization Timeout), BD (Interface Data Rate) and D7 (DIO7 Configuration) command descriptions for more information.

DO (Data Out) Buffer

When RF data is received, the data enters the DO buffer and is sent out the serial port to a host device. Once the DO Buffer reaches capacity, any additional incoming RF data is lost.

Hardware Flow Control ($\overline{\text{RTS}}$). If $\overline{\text{RTS}}$ is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the DO Buffer as long as $\overline{\text{RTS}}$ (pin 16) is de-asserted.

Two cases in which the DO Buffer may become full and possibly overflow:

1. If the RF data rate is set higher than the interface data rate of the module, the module will receive data from the transmitting module faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the DO buffer because of being held off by hardware or software flow control.

Refer to the D6 (DIO6 Configuration) command description for more information.

ADC and Digital I/O Line Support

The XBee®/XBee-PRO® RF Modules support ADC (Analog-to-digital conversion) and digital I/O line passing. The following pins support multiple functions:

Table 2-01. Pin functions and their associated pin numbers and commands
AD = Analog-to-Digital Converter, DIO = Digital Input/Output
Pin functions not applicable to this section are denoted within (parenthesis).

Pin Function	Pin#	AT Command
AD0 / DIO0	20	D0
AD1 / DIO1	19	D1
AD2 / DIO2	18	D2
AD3 / DIO3 / (COORD_SEL)	17	D3
AD4 / DIO4	11	D4
AD5 / DIO5 / (ASSOCIATE)	15	D5
DIO6 / (RTS)	16	D6
DIO7 / (CTS)	12	D7
DI8 / (DTR) / (Sleep_RQ)	9	D8

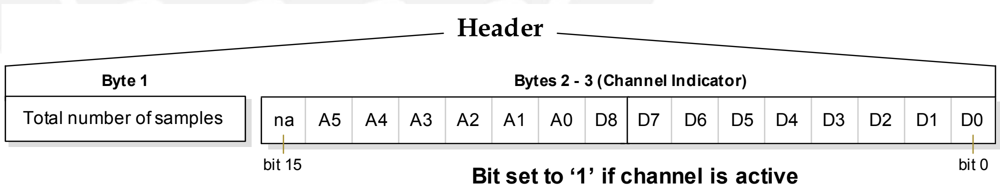
To enable ADC and DIO pin functions:

For ADC Support: Set ATDn = 2
For Digital Input support: Set ATDn = 3
For Digital Output Low support: Set ATDn = 4
For Digital Output High support: Set ATDn = 5

I/O Data Format

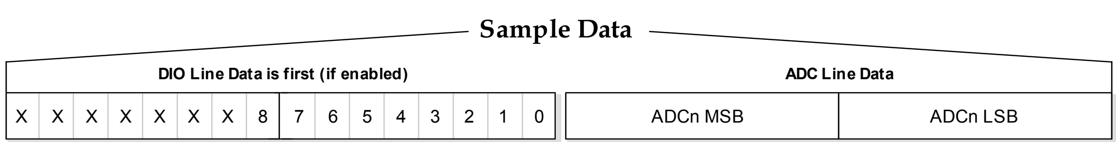
I/O data begins with a header. The first byte of the header defines the number of samples forth-coming. The last 2 bytes of the header (Channel Indicator) define which inputs are active. Each bit represents either a DIO line or ADC channel.

Figure 2-04. Header



Sample data follows the header and the channel indicator frame is used to determine how to read the sample data. If any of the DIO lines are enabled, the first 2 bytes are the DIO sample. The ADC data follows. ADC channel data is represented as an unsigned 10-bit value right-justified on a 16- bit boundary.

Figure 2-05. Sample Data



API Support

I/O data is sent out the UART using an API frame. All other data can be sent and received using Transparent Operation [refer to p11] or API framing if API mode is enabled (AP > 0).

API Operations support two RX (Receive) frame identifiers for I/O data (set 16-bit address to 0xFFFE and the module will do 64-bit addressing):

- 0x82 for RX (Receive) Packet: 64-bit address I/O
- 0x83 for RX (Receive) Packet: 16-bit address I/O

The API command header is the same as shown in the "RX (Receive) Packet: 64-bit Address" and "RX (Receive) Packet: 64-bit Address" API types [refer to p63]. RX data follows the format described in the I/O Data Format section [p13].

Applicable Commands: AP (API Enable)

Sleep Support

Automatic wakeup sampling can be suppressed by setting SO bit 1.When an RF module wakes, it will always do a sample based on any active ADC or DIO lines. This allows sampling based on the sleep cycle whether it be Cyclic Sleep (SM parameter = 4 or 5) or Pin Sleep (SM = 1 or 2). To gather more samples when awake, set the IR (Sample Rate) parameter.

For Cyclic Sleep modes: If the IR parameter is set, the module will stay awake until the IT (Samples before TX) parameter is met. The module will stay awake for ST (Time before Sleep) time.

Applicable Commands: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode), IC (DIO Change Detect), SO (Sleep Options)

DIO Pin Change Detect

When "DIO Change Detect" is enabled (using the IC command), DIO lines 0-7 are monitored. When a change is detected on a DIO line, the following will occur:

1. An RF packet is sent with the updated DIO pin levels. This packet will not contain any ADC samples.
2. Any queued samples are transmitted before the change detect data. This may result in receiving a packet with less than IT (Samples before TX) samples.

Note: Change detect will not affect Pin Sleep wake-up. The D8 pin (DTR/Sleep_RQ/DI8) is the only line that will wake a module from Pin Sleep. If not all samples are collected, the module will still enter Sleep Mode after a change detect packet is sent.

Applicable Commands: IC (DIO Change Detect), IT (Samples before TX)

NOTE: Change detect is only supported when the Dx (DIOx Configuration) parameter equals 3,4 or 5.

Sample Rate (Interval)

The Sample Rate (Interval) feature allows enabled ADC and DIO pins to be read periodically on modules that are not configured to operate in Sleep Mode. When one of the Sleep Modes is enabled and the IR (Sample Rate) parameter is set, the module will stay awake until IT (Samples before TX) samples have been collected.

Once a particular pin is enabled, the appropriate sample rate must be chosen. The maximum sample rate that can be achieved while using one A/D line is 1 sample/ms or 1 KHz (Note that the modem will not be able to keep up with transmission when IR & IT are equal to "1" and that configuring the modem to sample at rates greater than once every 20ms is not recommended).

Applicable Commands: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode)

I/O Line Passing

Virtual wires can be set up between XBee®/XBee-PRO® Modules. When an RF data packet is received that contains I/O data, the receiving module can be setup to update any enabled outputs (PWM and DIO) based on the data it receives.

Note that I/O lines are mapped in pairs. For example: AD0 can only update PWM0 and DI5 can only update DO5. The default setup is for outputs not to be updated, which results in the I/O data being sent out the UART (refer to the IU (Enable I/O Output) command). To enable the outputs to be updated, the IA (I/O Input Address) parameter must be setup with the address of the module that has the appropriate inputs enabled. This effectively binds the outputs to a particular module's input. This does not affect the ability of the module to receive I/O line data from other modules - only its ability to update enabled outputs. The IA parameter can also be setup to accept I/O data for output changes from any module by setting the IA parameter to 0xFFFF.

When outputs are changed from their non-active state, the module can be setup to return the output level to it non-active state. The timers are set using the Tn (Dn Output Timer) and PT (PWM Output Timeout) commands. The timers are reset every time a valid I/O packet (passed IA check) is received. The IC (Change Detect) and IR (Sample Rate) parameters can be setup to keep the output set to their active output if the system needs more time than the timers can handle.

Note: DI8 cannot be used for I/O line passing.

Applicable Commands: IA (I/O Input Address), Tn (Dn Output Timeout), P0 (PWM0 Configuration), P1 (PWM1 Configuration), M0 (PWM0 Output Level), M1 (PWM1 Output Level), PT (PWM Output Timeout), RP (RSSSI PWM Timer)

Configuration Example

As an example for a simple A/D link, a pair of RF modules could be set as follows:

Remote Configuration	**Base Configuration**
DL = 0x1234	DL = 0x5678
MY = 0x5678	MY = 0x1234
D0 = 2	P0 = 2
D1 = 2	P1 = 2
IR = 0x14	IU = 1
IT = 5	IA = 0x5678 (or 0xFFFF)

These settings configure the remote module to sample AD0 and AD1 once each every 20 ms. It then buffers 5 samples each before sending them back to the base module. The base should then receive a 32-Byte transmission (20 Bytes data and 12 Bytes framing) every 100 ms.

XBee®/XBee-PRO® Networks

The following terms will be used to explicate the network operations:

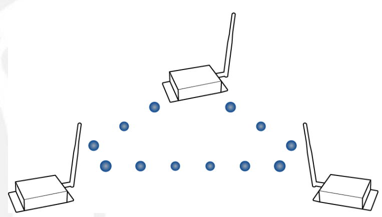
Table 2-02. Terms and definitions

Term	Definition
PAN	Personal Area Network - A data communication network that includes one or more End Devices and optionally a Coordinator.
Coordinator	A Full-function device (FFD) that provides network synchronization by polling nodes [NonBeacon (w/ Coordinator) networks only]
End Device	*When in the same network as a Coordinator* - RF modules that rely on a Coordinator for synchronization and can be put into states of sleep for low-power applications.
Association	The establishment of membership between End Devices and a Coordinator. Association is only applicable in NonBeacon (w/Coordinator) networks.

Peer-to-Peer

By default, XBee®/XBee-PRO RF Modules are configured to operate within a Peer-to-Peer network topology and therefore are not dependent upon Master/Slave relationships. NonBeacon systems operate within a Peer-to-Peer network topology and therefore are not dependent upon Master/Slave relationships. This means that modules remain synchronized without use of master/server configurations and each module in the network shares both roles of master and slave. Digi's peer-to-peer architecture features fast synchronization times and fast cold start times. This default configuration accommodates a wide range of RF data applications.

Figure 2-06. Peer-to-Peer Architecture



A peer-to-peer network can be established by configuring each module to operate as an End Device (CE = 0), disabling End Device Association on all modules (A1 = 0) and setting ID and CH parameters to be identical across the network.

NonBeacon (w/ Coordinator)

A device is configured as a Coordinator by setting the CE (Coordinator Enable) parameter to “1”. Coordinator power-up is governed by the A2 (Coordinator Association) parameter.

In a Coordinator system, the Coordinator can be configured to use direct or indirect transmissions. If the SP (Cyclic Sleep Period) parameter is set to “0”, the Coordinator will send data immediately. Otherwise, the SP parameter determines the length of time the Coordinator will retain the data before discarding it. Generally, SP (Cyclic Sleep Period) and ST (Time before Sleep) parameters should be set to match the SP and ST settings of the End Devices.

Association

Association is the establishment of membership between End Devices and a Coordinator. The establishment of membership is useful in scenarios that require a central unit (Coordinator) to relay messages to or gather data from several remote units (End Devices), assign channels or assign PAN IDs.

An RF data network that consists of one Coordinator and one or more End Devices forms a PAN (Personal Area Network). Each device in a PAN has a PAN Identifier [ID (PAN ID) parameter]. PAN IDs must be unique to prevent miscommunication between PANs. The Coordinator PAN ID is set using the ID (PAN ID) and A2 (Coordinator Association) commands.

An End Device can associate to a Coordinator without knowing the address, PAN ID or channel of the Coordinator. The A1 (End Device Association) parameter bit fields determine the flexibility of an End Device during association. The A1 parameter can be used for an End Device to dynamically set its destination address, PAN ID and/or channel.

For example: If the PAN ID of a Coordinator is known, but the operating channel is not; the A1 command on the End Device should be set to enable the 'Auto_Associate' and 'Reassign_Channel' bits. Additionally, the ID parameter should be set to match the PAN ID of the associated Coordinator.

Coordinator / End Device Setup and Operation

To configure a module to operate as a Coordinator, set the CE (Coordinator Enable) parameter to '1'. Set the CE parameter of End Devices to '0' (default). Coordinator and End Devices should contain matching firmware versions.

NonBeacon (w/ Coordinator) Systems

The Coordinator can be configured to use direct or indirect transmissions. If the SP (Cyclic Sleep Period) parameter is set to '0', the Coordinator will send data immediately. Otherwise, the SP parameter determines the length of time the Coordinator will retain the data before discarding it. Generally, SP (Cyclic Sleep Period) and ST (Time before Sleep) parameters should be set to match the SP and ST settings of the End Devices.

Coordinator Start-up

Coordinator power-up is governed by the A2 (Coordinator Association) command. On power-up, the Coordinator undergoes the following sequence of events:

1. Check A2 parameter- Reassign_PANID Flag

Set (bit 0 = 1) - The Coordinator issues an Active Scan. The Active Scan selects one channel and transmits a request to the broadcast address (0xFFFF) and broadcast PAN ID (0xFFFF). It then listens on that channel for beacons from any Coordinator operating on that channel. The listen time on each channel is determined by the SD (Scan Duration) parameter value.

Once the time expires on that channel, the Active Scan selects another channel and again transmits the BeaconRequest as before. This process continues until all channels have been scanned, or until 5 PANs have been discovered. When the Active Scan is complete, the results include a list of PAN IDs and Channels that are being used by other PANs. This list is used to assign an unique PAN ID to the new Coordinator. The ID parameter will be retained if it is not found in the Active Scan results. Otherwise, the ID (PAN ID) parameter setting will be updated to a PAN ID that was not detected.

Not Set (bit 0 = 0) - The Coordinator retains its ID setting. No Active Scan is performed.

2. Check A2 parameter - Reassign_Channel Flag (bit 1)

Set (bit 1 = 1) - The Coordinator issues an Energy Scan. The Energy Scan selects one channel and scans for energy on that channel. The duration of the scan is specified by the SD (Scan Duration) parameter. Once the scan is completed on a channel, the Energy Scan selects the next channel and begins a new scan on that channel. This process continues until all channels have been scanned.

When the Energy Scan is complete, the results include the maximal energy values detected on each channel. This list is used to determine a channel where the least energy was detected. If an Active Scan was performed (Reassign_PANID Flag set), the channels used by the detected PANs are eliminated as possible channels. Thus, the results of the Energy Scan and the Active Scan (if performed) are used to find the best channel (channel with the least energy that is not used by any detected PAN). Once the best channel has been selected, the CH (Channel) parameter value is updated to that channel.

Not Set (bit 1 = 0) - The Coordinator retains its CH setting. An Energy Scan is not performed.

3. Start Coordinator

The Coordinator starts on the specified channel (CH parameter) and PAN ID (ID parameter). Note, these may be selected in steps 1 and/or 2 above. The Coordinator will only allow End Devices to associate to it if the A2 parameter "AllowAssociation" flag is set. Once the Coordinator has successfully started, the Associate LED will blink 1 time per second. (The LED is solid if the Coordinator has not started.)

4. Coordinator Modifications

Once a Coordinator has started:
Modifying the A2 (Reassign_Channel or Reassign_PANID bits), ID, CH or MY parameters will cause the Coordinator's MAC to reset (The Coordinator RF module (including volatile RAM) is not reset). Changing the A2 AllowAssociation bit will not reset the Coordinator's MAC. In a non-beaconing system, End Devices that associated to the Coordinator prior to a MAC reset will have knowledge of the new settings on the Coordinator. Thus, if the Coordinator were to change its ID, CH or MY settings, the End Devices would no longer be able to communicate with the non-beacon Coordinator. Once a Coordinator has started, the ID, CH, MY or A2 (Reassign_Channel or Reassign_PANID bits) should not be changed.

End Device Start-up

End Device power-up is governed by the A1 (End Device Association) command. On power-up, the End Device undergoes the following sequence of events:

1. Check A1 parameter - AutoAssociate Bit

Set (bit 2 = 1) - End Device will attempt to associate to a Coordinator. (refer to steps 2-3).

Not Set (bit 2 = 0) - End Device will not attempt to associate to a Coordinator. The End Device will operate as specified by its ID, CH and MY parameters. Association is considered complete and the Associate LED will blink quickly (5 times per second). When the AutoAssociate bit is not set, the remaining steps (2-3) do not apply.

2. Discover Coordinator (if Auto-Associate Bit Set)

The End Device issues an Active Scan. The Active Scan selects one channel and transmits a BeaconRequest command to the broadcast address (0xFFFF) and broadcast PAN ID (0xFFFF). It then listens on that channel for beacons from any Coordinator operating on that channel. The listen time on each channel is determined by the SD parameter.

Once the time expires on that channel, the Active Scan selects another channel and again transmits the BeaconRequest command as before. This process continues until all channels have been scanned, or until 5 PANs have been discovered. When the Active Scan is complete, the results include a list of PAN IDs and Channels that are being used by detected PANs.

The End Device selects a Coordinator to associate with according to the A1 parameter "Reassign_PANID" and "Reassign_Channel" flags:

Reassign_PANID Bit Set (bit 0 = 1)- End Device can associate with a PAN with any ID value.

Reassign_PANID Bit Not Set (bit 0 = 0) - End Device will only associate with a PAN whose ID setting matches the ID setting of the End Device.

Reassign_Channel Bit Set (bit 1 = 1) - End Device can associate with a PAN with any CH value.

Reassign_Channel Bit Not Set (bit 1 = 0)- End Device will only associate with a PAN whose CH setting matches the CH setting of the End Device.

After applying these filters to the discovered Coordinators, if multiple candidate PANs exist, the End Device will select the PAN whose transmission link quality is the strongest. If no valid Coordinator is found, the End Device will either go to sleep (as dictated by its SM (Sleep Mode) parameter) or retry Association.

Note - An End Device will also disqualify Coordinators if they are not allowing association (A2 - AllowAssociation bit); or, if the Coordinator is not using the same NonBeacon scheme as the End Device. (They must both be programmed with NonBeacon code.)

3. Associate to Valid Coordinator

Once a valid Coordinator is found (step 2), the End Device sends an AssociationRequest message to the Coordinator. It then waits for an AssociationConfirmation to be sent from the Coordinator. Once the Confirmation is received, the End Device is Associated and the Associate LED will blink rapidly (2 times per second). The LED is solid if the End Device has not associated.

4. End Device Changes once an End Device has associated

Changing A1, ID or CH parameters will cause the End Device to disassociate and restart the Association procedure.

If the End Device fails to associate, the AI command can give some indication of the failure.

XBee®/XBee-PRO® Addressing

Every RF data packet sent over-the-air contains a Source Address and Destination Address field in its header. The RF module conforms to the 802.15.4 specification and supports both short 16-bit addresses and long 64-bit addresses. A unique 64-bit IEEE source address is assigned at the factory and can be read with the SL (Serial Number Low) and SH (Serial Number High) commands. Short addressing must be configured manually. A module will use its unique 64-bit address as its Source Address if its MY (16-bit Source Address) value is "0xFFFF" or "0xFFFE".

To send a packet to a specific module using 64-bit addressing: Set the Destination Address (DL + DH) of the sender to match the Source Address (SL + SH) of the intended destination module.

To send a packet to a specific module using 16-bit addressing: Set DL (Destination Address Low) parameter to equal the MY parameter of the intended destination module and set the DH (Destination Address High) parameter to '0'.

Unicast Mode

By default, the RF module operates in Unicast Mode. Unicast Mode is the only mode that supports retries. While in this mode, receiving modules send an ACK (acknowledgement) of RF packet reception to the transmitter. If the transmitting module does not receive the ACK, it will re-send the packet up to three times or until the ACK is received.

Short 16-bit addresses. The module can be configured to use short 16-bit addresses as the Source Address by setting (MY < 0xFFFE). Setting the DH parameter (DH = 0) will configure the Destination Address to be a short 16-bit address (if DL < 0xFFFE). For two modules to communicate using short addressing, the Destination Address of the transmitter module must match the MY parameter of the receiver.

The following table shows a sample network configuration that would enable Unicast Mode communications using short 16-bit addresses.

Table 2-03. Sample Unicast Network Configuration (using 16-bit addressing)

Parameter	RF Module 1	RF Module 2
MY (Source Address)	0x01	0x02
DH (Destination Address High)	0	0
DL (Destination Address Low)	0x02	0x01

Long 64-bit addresses. The RF module's serial number (SL parameter concatenated to the SH parameter) can be used as a 64-bit source address when the MY (16-bit Source Address) parameter is disabled. When the MY parameter is disabled (MY = 0xFFFF or 0xFFFE), the module's source address is set to the 64-bit IEEE address stored in the SH and SL parameters.

When an End Device associates to a Coordinator, its MY parameter is set to 0xFFFE to enable 64-bit addressing. The 64-bit address of the module is stored as SH and SL parameters. To send a packet to a specific module, the Destination Address (DL + DH) on the sender must match the Source Address (SL + SH) of the desired receiver.

Broadcast Mode

Any RF module within range will accept a packet that contains a broadcast address. When configured to operate in Broadcast Mode, receiving modules do not send ACKs (Acknowledgements) and transmitting modules do not automatically re-send packets as is the case in Unicast Mode.

To send a broadcast packet to all modules regardless of 16-bit or 64-bit addressing, set the destination addresses of all the modules as shown below.

Sample Network Configuration (All modules in the network):

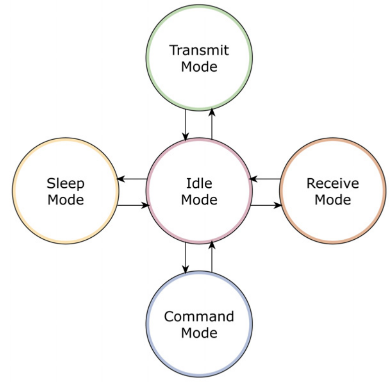
- DL (Destination Low Address) = 0x0000FFFF
- DH (Destination High Address) = 0x00000000 (default value)

NOTE: When programming the module, parameters are entered in hexadecimal notation (without the "0x" prefix). Leading zeros may be omitted.

Modes of Operation

XBee®/XBee-PRO® RF Modules operate in five modes.

Figure 2-07. Modes of Operation



Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. The module shifts into the other modes of operation under the following conditions:

- Transmit Mode (Serial data is received in the DI Buffer)
- Receive Mode (Valid RF data is received through the antenna)
- Sleep Mode (Sleep Mode condition is met)
- Command Mode (Command Mode Sequence is issued)

Transmit/Receive Modes

RF Data Packets

Each transmitted data packet contains a Source Address and Destination Address field. The Source Address matches the address of the transmitting module as specified by the MY (Source Address) parameter (if MY >= 0xFFFE), the SH (Serial Number High) parameter or the SL (Serial Number Low) parameter. The <Destination Address> field is created from the DH (Destination Address High) and DL (Destination Address Low) parameter values. The Source Address and/or Destination Address fields will either contain a 16-bit short or long 64-bit long address.

The RF data packet structure follows the 802.15.4 specification.

[Refer to the XBee/XBee-PRO Addressing section for more information]

Direct and Indirect Transmission

There are two methods to transmit data:

- Direct Transmission - data is transmitted immediately to the Destination Address
- Indirect Transmission - A packet is retained for a period of time and is only transmitted after the destination module (Source Address = Destination Address) requests the data.

Indirect Transmissions can only occur on a Coordinator. Thus, if all nodes in a network are End Devices, only Direct Transmissions will occur. Indirect Transmissions are useful to ensure packet delivery to a sleeping node. The Coordinator currently is able to retain up to 2 indirect messages.

Direct Transmission

A Coordinator can be configured to use only Direct Transmission by setting the SP (Cyclic Sleep Period) parameter to "0". Also, a Coordinator using indirect transmissions will revert to direct transmission if it knows the destination module is awake.

To enable this behavior, the ST (Time before Sleep) value of the Coordinator must be set to match the ST value of the End Device. Once the End Device either transmits data to the Coordinator or polls the Coordinator for data, the Coordinator will use direct transmission for all subsequent data transmissions to that module address until ST time occurs with no activity (at which point it will revert to using indirect transmissions for that module address). "No activity" means no transmission or reception of messages with a specific address. Global messages will not reset the ST timer.

Indirect Transmission

To configure Indirect Transmissions in a PAN (Personal Area Network), the SP (Cyclic Sleep Period) parameter value on the Coordinator must be set to match the longest sleep value of any End Device. The sleep period value on the Coordinator determines how long (time or number of beacons) the Coordinator will retain an indirect message before discarding it.

An End Device must poll the Coordinator once it wakes from Sleep to determine if the Coordinator has an indirect message for it. For Cyclic Sleep Modes, this is done automatically every time the module wakes (after SP time). For Pin Sleep Modes, the A1 (End Device Association) parameter value must be set to enable Coordinator polling on pin wake-up. Alternatively, an End Device can use the FP (Force Poll) command to poll the Coordinator as needed.

CCA (Clear Channel Assessment)

Prior to transmitting a packet, a CCA (Clear Channel Assessment) is performed on the channel to determine if the channel is available for transmission. The detected energy on the channel is compared with the CA (Clear Channel Assessment) parameter value. If the detected energy exceeds the CA parameter value, the packet is not transmitted.

Also, a delay is inserted before a transmission takes place. This delay is settable using the RN (Backoff Exponent) parameter. If RN is set to "0", then there is no delay before the first CCA is performed. The RN parameter value is the equivalent of the "minBE" parameter in the 802.15.4 specification. The transmit sequence follows the 802.15.4 specification.

By default, the MM (MAC Mode) parameter = 0. On a CCA failure, the module will attempt to re-send the packet up to two additional times.

When in Unicast packets with RR (Retries) = 0, the module will execute two CCA retries. Broadcast packets always get two CCA retries.

Acknowledgement

If the transmission is not a broadcast message, the module will expect to receive an acknowledgement from the destination node. If an acknowledgement is not received, the packet will be resent up to 3 more times. If the acknowledgement is not received after all transmissions, an ACK failure is recorded.

Sleep Mode

Sleep Modes enable the RF module to enter states of low-power consumption when not in use. In order to enter Sleep Mode, one of the following conditions must be met (in addition to the module having a non-zero SM parameter value):

- Sleep_RQ (pin 9) is asserted and the module is in a pin sleep mode (SM = 1, 2, or 5)
- The module is idle (no data transmission or reception) for the amount of time defined by the ST (Time before Sleep) parameter. [NOTE: ST is only active when SM = 4-5.]

Table 2-04. Sleep Mode Configurations

Sleep Mode Setting	Transition into Sleep Mode	Transition out of Sleep Mode (wake)	Characteristics	Related Commands	Power Consumption
Pin Hibernate (SM = 1)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Lowest Power	(SM)	< 10 µA (@3.0 VCC)
Pin Doze (SM = 2)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Fastest wake-up	(SM)	< 50 µA
Cyclic Sleep (SM = 4)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present / When SM = 5	(SM), SP, ST	< 50 µA when sleeping
Cyclic Sleep (SM = 5)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters or or on a falling edge transition of the SLEEP_RQ pin.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present. Module also wakes on a falling edge of SLEEP_RQ	(SM), SP, ST	< 50 µA when sleeping

The SM command is central to setting Sleep Mode configurations. By default, Sleep Modes are disabled (SM = 0) and the module remains in Idle/Receive Mode. When in this state, the module is constantly ready to respond to serial or RF activity.

Pin/Host-controlled Sleep Modes

The transient current when waking from pin sleep (SM = 1 or 2) does not exceed the idle current of the module. The current ramps up exponentially to its idle current.

Pin Hibernate (SM = 1)

- Pin/Host-controlled
- Typical power-down current: < 10 µA (@3.0 VCC)
- Wake-up time: 13.2 msec

Pin Hibernate Mode minimizes quiescent power (power consumed when in a state of rest or inactivity). This mode is voltage level-activated; when Sleep_RQ (pin 9) is asserted, the module will finish any transmit, receive or association activities, enter Idle Mode, and then enter a state of sleep. The module will not respond to either serial or RF activity while in pin sleep.

To wake a sleeping module operating in Pin Hibernate Mode, de-assert Sleep_RQ (pin 9). The module will wake when Sleep_RQ is de-asserted and is ready to transmit or receive when the CTS line is low. When waking the module, the pin must be de-asserted at least two 'byte times' after CTS goes low. This assures that there is time for the data to enter the DI buffer.

Pin Doze (SM = 2)

- Pin/Host-controlled
- Typical power-down current: < 50 µA
- Wake-up time: 2 msec

Pin Doze Mode functions as does Pin Hibernate Mode; however, Pin Doze features faster wake-up time and higher power consumption.

To wake a sleeping module operating in Pin Doze Mode, de-assert Sleep_RQ (pin 9). The module will wake when Sleep_RQ is de-asserted and is ready to transmit or receive when the CTS line is low. When waking the module, the pin must be de-asserted at least two 'byte times' after CTS goes low. This assures that there is time for the data to enter the DI buffer.

Cyclic Sleep Modes

Cyclic Sleep Remote (SM = 4)

- Typical Power-down Current: < 50 µA (when asleep)
- Wake-up time: 2 msec

The Cyclic Sleep Modes allow modules to periodically check for RF data. When the SM parameter is set to '4', the module is configured to sleep, then wakes once a cycle to check for data from a module configured as a Cyclic Sleep Coordinator (SM = 0, CE = 1). The Cyclic Sleep Remote sends a poll request to the coordinator at a specific interval set by the SP (Cyclic Sleep Period) parameter. The coordinator will transmit any queued data addressed to that specific remote upon receiving the poll request.

If no data is queued for the remote, the coordinator will not transmit and the remote will return to sleep for another cycle. If queued data is transmitted back to the remote, it will stay awake to allow for back and forth communication until the ST (Time before Sleep) timer expires.

Also note that $\overline{CTS}$ will go low each time the remote wakes, allowing for communication initiated by the remote host if desired.

Cyclic Sleep Remote with Pin Wake-up (SM = 5)

Use this mode to wake a sleeping remote module through either the RF interface or by the de-assertion of Sleep_RQ for event-driven communications. The cyclic sleep mode works as described above (Cyclic Sleep Remote) with the addition of a pin-controlled wake-up at the remote module. The Sleep_RQ pin is edge-triggered, not level-triggered. The module will wake when a low is detected then set $\overline{CTS}$ low as soon as it is ready to transmit or receive.

Any activity will reset the ST (Time before Sleep) timer so the module will go back to sleep only after there is no activity for the duration of the timer. Once the module wakes (pin-controlled), further pin activity is ignored. The module transitions back into sleep according to the ST time regardless of the state of the pin.

[Cyclic Sleep Coordinator (SM = 6)]

- Typical current = Receive current
- Always awake

NOTE: The SM=6 parameter value exists solely for backwards compatibility with firmware version 1.x60. If backwards compatibility with the older firmware version is not required, always use the CE (Coordinator Enable) command to configure a module as a Coordinator.

This mode configures a module to wake cyclic sleeping remotes through RF interfacing. The Coordinator will accept a message addressed to a specific remote 16 or 64-bit address and hold it in a buffer until the remote wakes and sends a poll request. Messages not sent directly (buffered and requested) are called "Indirect messages". The Coordinator only queues one indirect message at a time. The Coordinator will hold the indirect message for a period 2.5 times the sleeping period indicated by the SP (Cyclic Sleep Period) parameter. The Coordinator's SP parameter should be set to match the value used by the remotes.

Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming characters are interpreted as commands. Two Command Mode options are supported: AT Command Mode [refer to section below] and API Command Mode [p57].

AT Command Mode

To Enter AT Command Mode:

Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

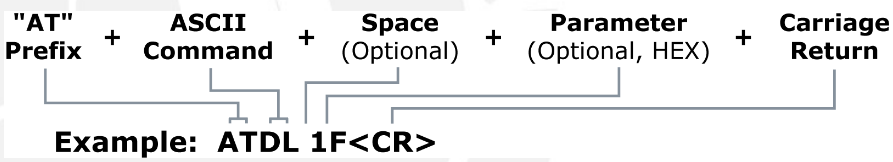
All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the 'Baud' setting on the "PC Settings" tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps).

To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

Figure 2-08. Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

System Response. When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an "OK" message. If execution of a command results in an error, the module returns an "ERROR" message.

To Exit AT Command Mode:

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).
   [OR]
2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to the RF Module Configuration chapter [p26].

3. RF Module Configuration

Programming the RF Module

Refer to the Command Mode section [p25] for more information about entering Command Mode, sending AT commands and exiting Command Mode. For information regarding module programming using API Mode, refer to the API Operation sections [p57].

Programming Examples

Setup

The programming examples in this section require the installation of Digi's X-CTU Software and a serial connection to a PC. (Digi stocks RS-232 and USB boards to facilitate interfacing with a PC.)

1. Install Digi's X-CTU Software to a PC by double-clicking the "setup_X-CTU.exe" file. (The file is located on the Digi CD and www.digi.com/xctu.)
2. Mount the RF module to an interface board, then connect the module assembly to a PC.
3. Launch the X-CTU Software and select the 'PC Settings' tab. Verify the baud and parity settings of the Com Port match those of the RF module.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the 'Baud' setting on the 'PC Settings' tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (which corresponds to 9600 bps).

Sample Configuration: Modify RF Module Destination Address

Example: Utilize the X-CTU "Terminal" tab to change the RF module's DL (Destination Address Low) parameter and save the new address to non-volatile memory.

After establishing a serial connection between the RF module and a PC [refer to the 'Setup' section above], select the "Terminal" tab of the X-CTU Software and enter the following command lines ('CR' stands for carriage return):

Method 1 (One line per command)

Send AT Command	System Response
+++	OK <CR> (Enter into Command Mode)
ATDL <Enter>	{current value} <CR> (Read Destination Address Low)
ATDL1A0D <Enter>	OK <CR> (Modify Destination Address Low)
ATWR <Enter>	OK <CR> (Write to non-volatile memory)
ATCN <Enter>	OK <CR> (Exit Command Mode)

Method 2 (Multiple commands on one line)

Send AT Command	System Response
+++	OK <CR> (Enter into Command Mode)
ATDL <Enter>	{current value} <CR> (Read Destination Address Low)
ATDL1A0D,WR,CN <Enter>	OK<CR> OK<CR> OK<CR>

Sample Configuration: Restore RF Module Defaults

Example: Utilize the X-CTU "Modem Configuration" tab to restore default parameter values.

After establishing a connection between the module and a PC [refer to the 'Setup' section above], select the "Modem Configuration" tab of the X-CTU Software.

1. Select the 'Read' button.
2. Select the 'Restore' button.

Remote Configuration Commands

The API firmware has provisions to send configuration commands to remote devices using the Remote Command Request API frame (see API Operation). This API frame can be used to send commands to a remote module to read or set command parameters.

The API firmware has provisions to send configuration commands (set or read) to a remote module using the Remote Command Request API frame (see API Operations). Remote commands can be issued to read or set command parameters on a remote device.

Sending a Remote Command

To send a remote command, the Remote Command Request frame should be populated with values for the 64 bit and 16 bit addresses. If 64 bit addressing is desired then the 16 bit address field should be filled with 0xFFFE. If any value other than 0xFFFE is used in the 16 bit address field then the 64 bit address field will be ignored and 16 bit addressing will be used. If a command response is desired, the Frame ID should be set to a non-zero value.

Applying Changes on Remote

When remote commands are used to change command parameter settings on a remote device, parameter changes do not take effect until the changes are applied. For example, changing the BD parameter will not change the actual serial interface rate on the remote until the changes are applied. Changes can be applied using remote commands in one of three ways:

Set the apply changes option bit in the API frame

Issue an AC command to the remote device

Issue a WR + FR command to the remote device to save changes and reset the device.

Remote Command Responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote will send a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its UART. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it will include the register value.

The device that sends a remote command will not receive a remote command response frame if:

The destination device could not be reached

The frame ID in the remote command request is set to 0.

Command Reference Tables

XBee®/XBee-PRO® RF Modules expect numerical values in hexadecimal. Hexadecimal values are designated by a "0x" prefix. Decimal equivalents are designated by a "d" suffix. Commands are contained within the following command categories (listed in the order that their tables appear):

- Special
- Networking & Security
- RF Interfacing
- Sleep (Low Power)
- Serial Interfacing
- I/O Settings
- Diagnostics
- AT Command Options

All modules within a PAN should operate using the same firmware version.

Special

Table 3-01. XBee-PRO Commands - Special

AT Command	Command Category	Name and Description	Parameter Range	Default
WR	Special	**Write**. Write parameter values to non-volatile memory so that parameter modifications persist through subsequent power-up or reset. Note: Once WR is issued, no additional characters should be sent to the module until after the response "OK\r" is received.	-	-
RE	Special	**Restore Defaults**. Restore module parameters to factory defaults.	-	-
FR (v1.x80*)	Special	**Software Reset**. Responds immediately with an OK then performs a hard reset ~100ms later.	-	-

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Networking & Security

Table 3-02. XBee®/XBee-PRO® Commands - Networking & Security (Sub-categories designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
CH	Networking {Addressing}	**Channel**. Set/Read the channel number used for transmitting and receiving data between RF modules (uses 802.15.4 protocol channel numbers).	0x0B - 0x1A (XBee) 0x0C - 0x17 (XBee-PRO)	0x0C (12d)
ID	Networking {Addressing}	**PAN ID**. Set/Read the PAN (Personal Area Network) ID. Use 0xFFFF to broadcast messages to all PANs.	0 - 0xFFFF	0x3332 (13106d)
DH	Networking {Addressing}	**Destination Address High**. Set/Read the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission. To transmit using a 16-bit address, set DH parameter to zero and DL less than 0xFFFF. 0x000000000000FFFF is the broadcast address for the PAN.	0 - 0xFFFFFFFF	0
DL	Networking {Addressing}	**Destination Address Low**. Set/Read the lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission. To transmit using a 16-bit address, set DH parameter to zero and DL less than 0xFFFF. 0x000000000000FFFF is the broadcast address for the PAN.	0 - 0xFFFFFFFF	0
MY	Networking {Addressing}	**16-bit Source Address.** Set/Read the RF module 16-bit source address. Set MY = 0xFFFF to disable reception of packets with 16-bit addresses. 64-bit source address (serial number) and broadcast address (0x000000000000FFFF) is always enabled.	0 - 0xFFFF	0
SH	Networking {Addressing}	**Serial Number High**. Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	0 - 0xFFFFFFFF [read-only]	Factory-set
SL	Networking {Addressing}	**Serial Number Low**. Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	0 - 0xFFFFFFFF [read-only]	Factory-set
RR (v1.xA0*)	Networking {Addressing}	**XBee Retries**. Set/Read the maximum number of retries the module will execute in addition to the 3 retries provided by the 802.15.4 MAC. For each XBee retry, the 802.15.4 MAC can execute up to 3 retries.	0 - 6	0
RN	Networking {Addressing}	**Random Delay Slots.** Set/Read the minimum value of the back-off exponent in the CSMA-CA algorithm that is used for collision avoidance. If RN = 0, collision avoidance is disabled during the first iteration of the algorithm (802.15.4 - macMinBE).	0 - 3 [exponent]	0
MM (v1.x80*)	Networking {Addressing}	**MAC Mode.** MAC Mode. Set/Read MAC Mode value. MAC Mode enables/disables the use of a Digi header in the 802.15.4 RF packet. When Modes 0 or 3 are enabled (MM=0,3), duplicate packet detection is enabled as well as certain AT commands. Please see the detailed MM description on page 47 for additional information.	0 - 3 0 = Digi Mode 1 = 802.15.4 (no ACKs) 2 = 802.15.4 (with ACKs) 3 = Digi Mode (no ACKs)	0
NI (v1.x80*)	Networking {Identification}	**Node Identifier.** Stores a string identifier. The register only accepts printable ASCII data. A string can not start with a space. Carriage return ends command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command.	20-character ASCII string	-
ND (v1.x80*)	Networking {Identification}	**Node Discover.** Discovers and reports all RF modules found. The following information is reported for each module discovered (the example cites use of Transparent operation (AT command format) - refer to the long ND command description regarding differences between Transparent and API operation). MY<CR> SH<CR> SL<CR> DB<CR> NI<CR><CR> The amount of time the module allows for responses is determined by the NT parameter. In Transparent operation, command completion is designated by a <CR> (carriage return). ND also accepts a Node Identifier as a parameter. In this case, only a module matching the supplied identifier will respond. If ND self-response is enabled (NO=1) the module initiating the node discover will also output a response for itself.	optional 20-character NI value	
NT (v1.xA0*)	Networking {Identification}	**Node Discover Time.** Set/Read the amount of time a node will wait for responses from other nodes when using the ND (Node Discover) command.	0x01 - 0xFC [x 100 ms]	0x19

Table 3-02. XBee®/XBee-PRO® Commands - Networking & Security (Sub-categories designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
NO (v1xC5)	Networking {Identification}	**Node Discover Options**. Enables node discover self-response on the module.	0-1	0
DN (v1.x80*)	Networking {Identification}	**Destination Node.** Resolves an NI (Node Identifier) string to a physical address. The following events occur upon successful command execution: 1. DL and DH are set to the address of the module with the matching Node Identifier. 2. "OK" is returned. 3. RF module automatically exits AT Command Mode If there is no response from a module within 200 msec or a parameter is not specified (left blank), the command is terminated and an "ERROR" message is returned.	20-character ASCII string	-
CE (v1.x80*)	Networking {Association}	**Coordinator Enable**. Set/Read the coordinator setting.	0 - 1 0 = End Device 1 = Coordinator	0
SC (v1.x80*)	Networking {Association}	**Scan Channels**. Set/Read list of channels to scan for all Active and Energy Scans as a bitfield. This affects scans initiated in command mode (AS, ED) and during End Device Association and Coordinator startup: bit 0 - 0x0B bit 4 - 0x0F bit 8 - 0x13 bit12 - 0x17 bit 1 - 0x0C bit 5 - 0x10 bit 9 - 0x14 bit13 - 0x18 bit 2 - 0x0D bit 6 - 0x11 bit 10 - 0x15 bit14 - 0x19 bit 3 - 0x0E bit 7 - 0x12 bit 11 - 0x16 bit 15 - 0x1A	0 - 0xFFFF [bitfield] (bits 0, 14, 15 not allowed on the XBee-PRO)	0x1FFE (all XBee-PRO Channels)
SD (v1.x80*)	Networking {Association}	**Scan Duration**. Set/Read the scan duration exponent. ***End Device*** - Duration of Active Scan during Association. ***Coordinator*** - If 'ReassignPANID' option is set on Coordinator [refer to A2 parameter], SD determines the length of time the Coordinator will scan channels to locate existing PANs. If 'ReassignChannel' option is set, SD determines how long the Coordinator will perform an Energy Scan to determine which channel it will operate on. 'Scan Time' is measured as (# of channels to scan] * (2 ^ SD) * 15.36ms). The number of channels to scan is set by the SC command. The XBee can scan up to 16 channels (SC = 0xFFFF). The XBee PRO can scan up to 13 channels (SC = 0x3FFE). Example: The values below show results for a 13 channel scan: If SD = 0, time = 0.18 sec SD = 8, time = 47.19 sec SD = 2, time = 0.74 sec SD = 10, time = 3.15 min SD = 4, time = 2.95 sec SD = 12, time = 12.58 min SD = 6, time = 11.80 sec SD = 14, time = 50.33 min	0-0x0F [exponent]	4
A1 (v1.x80*)	Networking {Association}	**End Device Association**. Set/Read End Device association options. bit 0 - ReassignPanID 0 - Will only associate with Coordinator operating on PAN ID that matches module ID 1 - May associate with Coordinator operating on any PAN ID bit 1 - ReassignChannel 0 - Will only associate with Coordinator operating on matching CH Channel setting 1 - May associate with Coordinator operating on any Channel bit 2 - AutoAssociate 0 - Device will not attempt Association 1 - Device attempts Association until success Note: This bit is used only for Non-Beacon systems. End Devices in Beacon-enabled system must always associate to a Coordinator bit 3 - PollCoordOnPinWake 0 - Pin Wake will not poll the Coordinator for indirect (pending) data 1 - Pin Wake will send Poll Request to Coordinator to extract any pending data bits 4 - 7 are reserved	0 - 0x0F [bitfield]	0
A2 (v1.x80*)	Networking {Association}	**Coordinator Association**. Set/Read Coordinator association options. bit 0 - ReassignPanID 0 - Coordinator will not perform Active Scan to locate available PAN ID. It will operate on ID (PAN ID). 1 - Coordinator will perform Active Scan to determine an available ID (PAN ID). If a PAN ID conflict is found, the ID parameter will change. bit 1 - ReassignChannel - 0 - Coordinator will not perform Energy Scan to determine free channel. It will operate on the channel determined by the CH parameter. 1 - Coordinator will perform Energy Scan to find a free channel, then operate on that channel. bit 2 - AllowAssociation - 0 - Coordinator will not allow any devices to associate to it. 1 - Coordinator will allow devices to associate to it. bits 3 - 7 are reserved	0 - 7 [bitfield]	0

Table 3-02. XBee®/XBee-PRO® Commands - Networking & Security (Sub-categories designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
AI (v1.x80*)	Networking {Association}	**Association Indication**. Read errors with the last association request: 0x00 - Successful Completion - Coordinator successfully started or End Device association complete 0x01 - Active Scan Timeout 0x02 - Active Scan found no PANs 0x03 - Active Scan found PAN, but the CoordinatorAllowAssociation bit is not set 0x04 - Active Scan found PAN, but Coordinator and End Device are not configured to support beacons 0x05 - Active Scan found PAN, but the Coordinator ID parameter does not match the ID parameter of the End Device 0x06 - Active Scan found PAN, but the Coordinator CH parameter does not match the CH parameter of the End Device 0x07 - Energy Scan Timeout 0x08 - Coordinator start request failed 0x09 - Coordinator could not start due to invalid parameter 0x0A - Coordinator Realignment is in progress 0x0B - Association Request not sent 0x0C - Association Request timed out - no reply was received 0x0D - Association Request had an Invalid Parameter 0x0E - Association Request Channel Access Failure. Request was not transmitted - CCA failure 0x0F - Remote Coordinator did not send an ACK after Association Request was sent 0x10 - Remote Coordinator did not reply to the Association Request, but an ACK was received after sending the request 0x11 - [reserved] 0x12 - Sync-Loss - Lost synchronization with a Beaconing Coordinator 0x13 - Disassociated - No longer associated to Coordinator 0xFF - RF Module is attempting to associate	0 - 0x13 [read-only]	-
DA (v1.x80*)	Networking {Association}	**Force Disassociation**. End Device will immediately disassociate from a Coordinator (if associated) and reattempt to associate.	-	-
FP (v1.x80*)	Networking {Association}	**Force Poll**. Request indirect messages being held by a coordinator.	-	-
AS (v1.x80*)	Networking {Association}	**Active Scan**. Send Beacon Request to Broadcast Address (0xFFFF) and Broadcast PAN (0xFFFF) on every channel. The parameter determines the time the radio will listen for Beacons on each channel. A PanDescriptor is created and returned for every Beacon received from the scan. Each PanDescriptor contains the following information: CoordAddress (SH, SL)<CR> CoordPanID (ID)<CR> CoordAddrMode <CR> 0x02 = 16-bit Short Address 0x03 = 64-bit Long Address Channel (CH parameter) <CR> SecurityUse<CR> ACLEntry<CR> SecurityFailure<CR> SuperFrameSpec<CR> (2 bytes): bit 15 - Association Permitted (MSB) bit 14 - PAN Coordinator bit 13 - Reserved bit 12 - Battery Life Extension bits 8-11 - Final CAP Slot bits 4-7 - Superframe Order bits 0-3 - Beacon Order GtsPermit<CR> RSSI<CR> (RSSI is returned as -dBm) TimeStamp<CR> (3 bytes) <CR> A carriage return <CR> is sent at the end of the AS command. The Active Scan is capable of returning up to 5 PanDescriptors in a scan. The actual scan time on each channel is measured as Time = [(2 ^SD PARAM) * 15.36] ms. Note the total scan time is this time multiplied by the number of channels to be scanned (16 for the XBee and 13 for the XBee-PRO). Also refer to SD command description.	0 - 6	-
ED (v1.x80*)	Networking {Association}	**Energy Scan**. Send an Energy Detect Scan. This parameter determines the length of scan on each channel. The maximal energy on each channel is returned & each value is followed by a carriage return. An additional carriage return is sent at the end of the command. The values returned represent the detected energy level in units of -dBm. The actual scan time on each channel is measured as Time = [(2 ^ED) * 15.36] ms. Note the total scan time is this time multiplied by the number of channels to be scanned (refer to SD parameter).	0 - 6	-
EE (v1.xA0*)	Networking {Security}	**AES Encryption Enable**. Disable/Enable 128-bit AES encryption support. Use in conjunction with the KY command.	0 - 1	0 (disabled)
KY (v1.xA0*)	Networking {Security}	**AES Encryption Key**. Set the 128-bit AES (Advanced Encryption Standard) key for encrypting/decrypting data. The KY register cannot be read.	0 - (any 16-Byte value)	-

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

RF Interfacing

Table 3-03. XBee/XBee-PRO Commands - RF Interfacing

AT Command	Command Category	Name and Description	Parameter Range	Default
PL	RF Interfacing	**Power Level**. Select/Read the power level at which the RF module transmits conducted power.	0 - 4 (XBee / XBee-PRO) 0 = -10 / 10 dBm 1 = -6 / 12 dBm 2 = -4 / 14 dBm 3 = -2 / 16 dBm 4 = 0 / 18 dBm XBee-PRO International variant: PL=4: 10 dBm PL=3: 8 dBm PL=2: 2 dBm PL=1: -3 dBm PL=0: -3 dBm	4
CA (v1.x80*)	RF Interfacing	**CCA Threshold**. Set/read the CCA (Clear Channel Assessment) threshold. Prior to transmitting a packet, a CCA is performed to detect energy on the channel. If the detected energy is above the CCA Threshold, the module will not transmit the packet.	0x24 - 0x50 [-dBm]	0x2C (-44d dBm)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Sleep (Low Power)

Table 3-04. XBee®/XBee-PRO® Commands - Sleep (Low Power)

AT Command	Command Category	Name and Description	Parameter Range	Default
SM	Sleep (Low Power)	**Sleep Mode**. Set/Read Sleep Mode configurations.	0 - 5 0 = No Sleep 1 = Pin Hibernate 2 = Pin Doze 3 = Reserved 4 = Cyclic sleep remote 5 = Cyclic sleep remote w/ pin wake-up 6 = [Sleep Coordinator] for backwards compatibility w/ v1.x6 only; otherwise, use CE command.	0
SO	Sleep (Low Power)	Sleep Options Set/Read the sleep mode options. Bit 0 - Poll wakeup disable 0 - Normal operations. A module configured for cyclic sleep will poll for data on waking. 1 - Disable wakeup poll. A module configured for cyclic sleep will not poll for data on waking. Bit 1 - ADC/DIO wakeup sampling disable. 0 - Normal operations. A module configured in a sleep mode with ADC/DIO sampling enabled will automatically perform a sampling on wakeup. 1 - Suppress sample on wakeup. A module configured in a sleep mode with ADC/DIO sampling enabled will not automatically sample on wakeup.	0-4	0
ST	Sleep (Low Power)	**Time before Sleep.** <NonBeacon firmware> Set/Read time period of inactivity (no serial or RF data is sent or received) before activating Sleep Mode. ST parameter is only valid with Cyclic Sleep settings (SM = 4 - 5). Coordinator and End Device ST values must be equal. Also note, the GT parameter value must always be less than the ST value. (If GT > ST, the configuration will render the module unable to enter into command mode.) If the ST parameter is modified, also modify the GT parameter accordingly.	1 - 0xFFFF [x 1 ms]	0x1388 (5000d)
SP	Sleep (Low Power)	**Cyclic Sleep Period.** <NonBeacon firmware> Set/Read sleep period for cyclic sleeping remotes. Coordinator and End Device SP values should always be equal. To send Direct Messages, set SP = 0. *End Device* - SP determines the sleep period for cyclic sleeping remotes. Maximum sleep period is 268 seconds (0x68B0). *Coordinator* - If non-zero, SP determines the time to hold an indirect message before discarding it. A Coordinator will discard indirect messages after a period of (2.5 * SP).	0 - 0x68B0 [x 10 ms]	0
DP (1.x80*)	Sleep (Low Power)	**Disassociated Cyclic Sleep Period.** <NonBeacon firmware> *End Device* - Set/Read time period of sleep for cyclic sleeping remotes that are configured for Association but are not associated to a Coordinator. (i.e. If a device is configured to associate, configured as a Cyclic Sleep remote, but does not find a Coordinator, it will sleep for DP time before reattempting association.) Maximum sleep period is 268 seconds (0x68B0). DP should be > 0 for NonBeacon systems.	1 - 0x68B0 [x 10 ms]	0x3E8 (1000d)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Serial Interfacing

Table 3-05. XBee-PRO Commands - Serial Interfacing

AT Command	Command Category	Name and Description	Parameter Range	Default
BD	Serial Interfacing	**Interface Data Rate**. Set/Read the serial interface data rate for communications between the RF module serial port and host. Request non-standard baud rates with values above 0x80 using a terminal window. Read the BD register to find actual baud rate achieved.	0 - 7 (standard baud rates) 0 = 1200 bps 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200 0x80 - 0x3D090 (non-standard baud rates up to 250 Kbps)	3
RO	Serial Interfacing	**Packetization Timeout**. Set/Read number of character times of inter-character delay required before transmission. Set to zero to transmit characters as they arrive instead of buffering them into one RF packet.	0 - 0xFF [x character times]	3
AP (v1.x80*)	Serial Interfacing	**API Enable**. Disable/Enable API Mode.	0 - 2 0 =Disabled 1 = API enabled 2 = API enabled (w/escaped control characters)	0
NB	Serial Interfacing	**Parity.** Set/Read parity settings.	0 - 4 0 = 8-bit no parity 1 = 8-bit even 2 = 8-bit odd 3 = 8-bit mark 4 = 8-bit space	0
PR (v1.x80*)	Serial Interfacing	**Pull-up Resistor Enable**. Set/Read bitfield to configure internal pull-up resistor status for I/O lines Bitfield Map: bit 0 - AD4/DIO4 (pin11) bit 1 - AD3 / DIO3 (pin17) bit 2 - AD2/DIO2 (pin18) bit 3 - AD1/DIO1 (pin19) bit 4 - AD0 / DIO0 (pin20) bit 5 - RTS / AD6 / DIO6 (pin16) bit 6 - DTR / SLEEP_RQ / DI8 (pin9) bit 7 - DIN/CONFIG (pin3) Bit set to "1" specifies pull-up enabled; "0" specifies no pull-up	0 - 0xFF	0xFF

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

I/O Settings

Table 3-06. XBee-PRO Commands - I/O Settings (sub-category designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
D8	I/O Settings	**DI8 Configuration**. Select/Read options for the DI8 line (pin 9) of the RF module.	0 - 1 0 = Disabled 3 = DI (1,2,4 & 5 n/a)	0
D7 (v1.x80*)	I/O Settings	**DIO7 Configuration**. Select/Read settings for the DIO7 line (pin 12) of the RF module. Options include CTS flow control and I/O line settings.	0 - 1 0 = Disabled 1 = CTS Flow Control 2 = (n/a) 3 = DI 4 = DO low 5 = DO high 6 = RS485 Tx Enable Low 7 = RS485 Tx Enable High	1
D6 (v1.x80*)	I/O Settings	**DIO6 Configuration**. Select/Read settings for the DIO6 line (pin 16) of the RF module. Options include RTS flow control and I/O line settings.	0 - 1 0 = Disabled 1 = RTS flow control 2 = (n/a) 3 = DI 4 = DO low 5 = DO high	0

Table 3-06. XBee-PRO Commands - I/O Settings (sub-category designated within {brackets})

AT Command	Command Category	Name and Description	Parameter Range	Default
D5 (v1.x80*)	I/O Settings	**DIO5 Configuration**. Configure settings for the DIO5 line (pin 15) of the RF module. Options include Associated LED indicator (blinks when associated) and I/O line settings.	0 - 1 0 = Disabled 1 = Associated indicator 2 = ADC 3 = DI 4 = DO low 5 = DO high	1
D0 - D4 (v1.xA0*)	I/O Settings	**(DIO4 -DIO4) Configuration**. Select/Read settings for the following lines: AD0/DIO0 (pin 20), AD1/DIO1 (pin 19), AD2/DIO2 (pin 18), AD3/DIO3 (pin 17), AD4/DIO4 (pin 11). Options include: Analog-to-digital converter, Digital Input and Digital Output.	0 - 1 0 = Disabled 1 = (n/a) 2 = ADC 3 = DI 4 = DO low 5 = DO high	0
IU (v1.xA0*)	I/O Settings	**I/O Output Enable.** Disables/Enables I/O data received to be sent out UART. The data is sent using an API frame regardless of the current AP parameter value.	0 - 1 0 = Disabled 1 = Enabled	1
IT (v1.xA0*)	I/O Settings	**Samples before TX.** Set/Read the number of samples to collect before transmitting data. Maximum number of samples is dependent upon the number of enabled inputs.	1 - 0xFF	1
IS (v1.xA0*)	I/O Settings	**Force Sample.** Force a read of all enabled inputs (DI or ADC). Data is returned through the UART. If no inputs are defined (DI or ADC), this command will return error.	8-bit bitmap (each bit represents the level of an I/O line setup as an output)	-
IO (v1.xA0*)	I/O Settings	**Digital Output Level.** Set digital output level to allow DIO lines that are setup as outputs to be changed through Command Mode.	-	-
IC (v1.xA0*)	I/O Settings	**DIO Change Detect.** Set/Read bitfield values for change detect monitoring. Each bit enables monitoring of DIO0 - DIO7 for changes. If detected, data is transmitted with DIO data only. Any samples queued waiting for transmission will be sent first.	0 - 0xFF [bitfield]	0 (disabled)
IR (v1.xA0*)	I/O Settings	**Sample Rate.** Set/Read sample rate. When set, this parameter causes the module to sample all enabled inputs at a specified interval.	0 - 0xFFFF [x 1 msec]	0
IA (v1.xA0*)	I/O Settings {I/O Line Passing}	**I/O Input Address**. Set/Read addresses of module to which outputs are bound. Setting all bytes to 0xFF will not allow any received I/O packet to change outputs. Setting address to 0xFFFF will allow any received I/O packet to change outputs.	0 - 0xFFFFFFFFFFFFFFFF	0xFFFFFFFF FFFFFFFF
T0 - T7 (v1.xA0*)	I/O Settings {I/O Line Passing}	**(D0 - D7) Output Timeout.** Set/Read Output timeout values for lines that correspond with the D0 - D7 parameters. When output is set (due to I/O line passing) to a non-default level, a timer is started which when expired will set the output to it default level. The timer is reset when a valid I/O packet is received.	0 - 0xFF [x 100 ms]	0xFF
P0	I/O Settings {I/O Line Passing}	**PWM0 Configuration**. Select/Read function for PWM0 pin.	0 - 2 0 = Disabled 1 = RSSI 2 = PWM Output	1
P1 (v1.xA0*)	I/O Settings {I/O Line Passing}	**PWM1 Configuration**. Select/Read function for PWM1 pin.	0 - 2 0 = Disabled 1 = RSSI 2 = PWM Output	0
M0 (v1.xA0*)	I/O Settings {I/O Line Passing}	**PWM0 Output Level**. Set/Read the PWM0 output level.	0 - 0x03FF	-
M1 (v1.xA0*)	I/O Settings {I/O Line Passing}	**PWM1 Output Level**. Set/Read the PWM1 output level.	0 - 0x03FF	-
PT (v1.xA0*)	I/O Settings {I/O Line Passing}	**PWM Output Timeout.** Set/Read output timeout value for both PWM outputs. When PWM is set to a non-zero value: Due to I/O line passing, a time is started which when expired will set the PWM output to zero. The timer is reset when a valid I/O packet is received.]	0 - 0xFF [x 100 ms]	0xFF
RP	I/O Settings {I/O Line Passing}	**RSSI PWM Timer.** Set/Read PWM timer register. Set the duration of PWM (pulse width modulation) signal output on the RSSI pin. The signal duty cycle is updated with each received packet and is shut off when the timer expires.]	0 - 0xFF [x 100 ms]	0x28 (40d)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Diagnostics

Table 3-07. XBee®/XBee-PRO® Commands - Diagnostics

AT Command	Command Category	Name and Description	Parameter Range	Default
VR	Diagnostics	**Firmware Version**. Read firmware version of the RF module.	0 - 0xFFFF [read-only]	Factory-set
VL (v1.x80*)	Diagnostics	**Firmware Version - Verbose**. Read detailed version information (including application build date, MAC, PHY and bootloader versions). The VL command has been deprecated in version 10C9. It is not supported in firmware versions after 10C8	-	-

Table 3-07. XBee®/XBee-PRO® Commands - Diagnostics

AT Command	Command Category	Name and Description	Parameter Range	Default
HV (v1.x80*)	Diagnostics	**Hardware Version**. Read hardware version of the RF module.	0 - 0xFFFF [read-only]	Factory-set
DB	Diagnostics	**Received Signal Strength**. Read signal level [in dB] of last good packet received (RSSI). Absolute value is reported. (For example: 0x58 = -88 dBm) Reported value is accurate between -40 dBm and RX sensitivity.	0x17-0x5C (XBee) 0x24-0x64 (XBee-PRO) [read-only]	-
EC (v1.x80*)	Diagnostics	**CCA Failures.** Reset/Read count of CCA (Clear Channel Assessment) failures. This parameter value increments when the module does not transmit a packet because it detected energy above the CCA threshold level set with CA command. This count saturates at its maximum value. Set count to “0” to reset count.	0 - 0xFFFF	-
EA (v1.x80*)	Diagnostics	**ACK Failures**. Reset/Read count of acknowledgment failures. This parameter value increments when the module expires its transmission retries without receiving an ACK on a packet transmission. This count saturates at its maximum value. Set the parameter to “0” to reset count.	0 - 0xFFFF	-
ED (v1.x80*)	Diagnostics	**Energy Scan**. Send ‘Energy Detect Scan’. ED parameter determines the length of scan on each channel. The maximal energy on each channel is returned and each value is followed by a carriage return. Values returned represent detected energy levels in units of -dBm. Actual scan time on each channel is measured as Time = [(2 ^ SD) * 15.36] ms. Total scan time is this time multiplied by the number of channels to be scanned.	0 - 6	-

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

AT Command Options

Table 3-08. XBee®/XBee-PRO® Commands - AT Command Options

AT Command	Command Category	Name and Description	Parameter Range	Default
CT	AT Command Mode Options	**Command Mode Timeout.** Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	2 - 0xFFFF [x 100 ms]	0x64 (100d)
CN	AT Command Mode Options	**Exit Command Mode**. Explicitly exit the module from AT Command Mode.	--	--
AC (v1.xA0*)	AT Command Mode Options	**Apply Changes**. Explicitly apply changes to queued parameter value(s) and re-initialize module.	--	--
GT	AT Command Mode Options	**Guard Times**. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT+ CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	2 - 0x0CE4 [x 1 ms]	0x3E8 (1000d)
CC	AT Command Mode Options	**Command Sequence Character**. Set/Read the ASCII character value to be used between Guard Times of the AT Command Mode Sequence (GT+CC+GT). The AT Command Mode Sequence enters the RF module into AT Command Mode.	0 - 0xFF	0x2B ('+' ASCII)

* Firmware version in which the command was first introduced (firmware versions are numbered in hexadecimal notation.)

Command Descriptions

Command descriptions in this section are listed alphabetically. Command categories are designated within "< >" symbols that follow each command title. XBee®/XBee-PRO® RF Modules expect parameter values in hexadecimal (designated by the "0x" prefix).

All modules operating within the same network should contain the same firmware version.

A1 (End Device Association) Command

<Networking {Association}> The A1 command is used to set and read association options for an End Device.

Use the table below to determine End Device behavior in relation to the A1 parameter.

AT Command: ATA1

Parameter Range: 0 – 0x0F [bitfield]

Default Parameter Value: 0

Related Commands: ID (PAN ID), NI (Node Identifier), CH (Channel), CE (Coordinator Enable), A2 (Coordinator Association)

Minimum Firmware Version Required: v1.x80

Bit number	End Device Association Option
0 - ReassignPanID	0 - Will only associate with Coordinator operating on PAN ID that matches Node Identifier
	1 - May associate with Coordinator operating on any PAN ID
1 - ReassignChannel	0 - Will only associate with Coordinator operating on Channel that matches CH setting
	1 - May associate with Coordinator operating on any Channel
2 - AutoAssociate	0 - Device will not attempt Association
	1 - Device attempts Association until success Note: This bit is used only for Non-Beacon systems. End Devices in a Beaconing system must always associate to a Coordinator
3 - PollCoordOnPinWake	0 - Pin Wake will not poll the Coordinator for pending (indirect) Data
	1 - Pin Wake will send Poll Request to Coordinator to extract any pending data
4 - 7	[reserved]

A2 (Coordinator Association) Command

<Networking {Association}> The A2 command is used to set and read association options of the Coordinator.

Use the table below to determine Coordinator behavior in relation to the A2 parameter.

AT Command: ATA2

Parameter Range: 0 – 7 [bitfield]

Default Parameter Value: 0

Related Commands: ID (PAN ID), NI (Node Identifier), CH (Channel), CE (Coordinator Enable), A1 (End Device Association), AS Active Scan), ED (Energy Scan)

Minimum Firmware Version Required: v1.x80

Bit number	End Device Association Option
0 - ReassignPanID	0 - Coordinator will not perform Active Scan to locate available PAN ID. It will operate on ID (PAN ID).
	1 - Coordinator will perform Active Scan to determine an available ID (PAN ID). If a PAN ID conflict is found, the ID parameter will change.
1 - ReassignChannel	0 - Coordinator will not perform Energy Scan to determine free channel. It will operate on the channel determined by the CH parameter.
	1 - Coordinator will perform Energy Scan to find a free channel, then operate on that channel.
2 - AllowAssociate	0 - Coordinator will not allow any devices to associate to it.
	1 - Coordinator will allow devices to associate to it.
3 - 7	[reserved]

The binary equivalent of the default value (0x06) is 00000110. 'Bit 0' is the last digit of the sequence.

AC (Apply Changes) Command

<AT Command Mode Options> The AC command is used to explicitly apply changes to module parameter values. 'Applying changes' means that the module is re-initialized based on changes made to its parameter values. Once changes are applied, the module immediately operates according to the new parameter values.

AT Command: ATAC	
Minimum Firmware Version Required: v1.xA0	

This behavior is in contrast to issuing the WR (Write) command. The WR command saves parameter values to non-volatile memory, but the module still operates according to previously saved values until the module is re-booted or the CN (Exit AT Command Mode) command is issued.

Refer to the "AT Command - Queue Parameter Value" API type for more information.

AI (Association Indication) Command

<Networking {Association}> The AI command is used to indicate occurrences of errors during the last association request.

Use the table below to determine meaning of the returned values.

AT Command: ATAI	
Parameter Range: 0 - 0x13 [read-only]	
Related Commands: AS (Active Scan), ID (PAN ID), CH (Channel), ED (Energy Scan), A1 (End Device Association), A2 (Coordinator Association), CE (Coordinator Enable)	
Minimum Firmware Version Required: v1.x80	

Returned Value (Hex)	Association Indication
0x00	Successful Completion - Coordinator successfully started or End Device association complete
0x01	Active Scan Timeout
0x02	Active Scan found no PANs
0x03	Active Scan found PAN, but the Coordinator Allow Association bit is not set
0x04	Active Scan found PAN, but Coordinator and End Device are not configured to support beacons
0x05	Active Scan found PAN, but Coordinator ID (PAN ID) value does not match the ID of the End Device
0x06	Active Scan found PAN, but Coordinator CH (Channel) value does not match the CH of the End Device
0x07	Energy Scan Timeout
0x08	Coordinator start request failed
0x09	Coordinator could not start due to Invalid Parameter
0x0A	Coordinator Realignment is in progress
0x0B	Association Request not sent
0x0C	Association Request timed out - no reply was received
0x0D	Association Request had an Invalid Parameter
0x0E	Association Request Channel Access Failure - Request was not transmitted - CCA failure
0x0F	Remote Coordinator did not send an ACK after Association Request was sent
0x10	Remote Coordinator did not reply to the Association Request, but an ACK was received after sending the request
0x11	[reserved]
0x12	Sync-Loss - Lost synchronization with a Beaconing Coordinator
0x13	Disassociated - No longer associated to Coordinator
0xFF	RF Module is attempting to associate

AP (API Enable) Command

<Serial Interfacing> The AP command is used to enable the RF module to operate using a frame-based API instead of using the default Transparent (UART) mode.

AT Command: ATAP

Parameter Range:0 – 2

Parameter	Configuration
0	Disabled (Transparent operation)
1	API enabled
2	API enabled (with escaped characters)

Default Parameter Value:0

Minimum Firmware Version Required: v1.x80

Refer to the API Operation section when API operation is enabled (AP = 1 or 2).

AS (Active Scan) Command

<Network {Association}> The AS command is used to send a Beacon Request to a Broadcast (0xFFFF) and Broadcast PAN (0xFFFF) on every channel. The parameter determines the amount of time the RF module will listen for Beacons on each channel. A 'PanDescriptor' is created and returned for every Beacon received from the scan. Each PanDescriptor contains the following information:

AT Command: ATAS

Parameter Range: 0 – 6

Related Command: SD (Scan Duration), DL (Destination Low Address), DH (Destination High Address), ID (PAN ID), CH (Channel)

Minimum Firmware Version Required: v1.x80

CoordAddress (SH + SL parameters)<CR> (NOTE: If MY on the coordinator is set less than 0xFFFF, the MY value is displayed)
CoordPanID (ID parameter)<CR>
CoordAddrMode <CR>
  0x02 = 16-bit Short Address
  0x03 = 64-bit Long Address
Channel (CH parameter) <CR>
SecurityUse<CR>
ACLEntry<CR>
SecurityFailure<CR>
SuperFrameSpec<CR> (2 bytes):
  bit 15 - Association Permitted (MSB)
  bit 14 - PAN Coordinator
  bit 13 - Reserved
  bit 12 - Battery Life Extension
  bits 8-11 - Final CAP Slot
  bits 4-7 - Superframe Order
  bits 0-3 - Beacon Order
GtsPermit<CR>
RSSI<CR> (- RSSI is returned as -dBm)
TimeStamp<CR> (3 bytes)
<CR> (A carriage return <CR> is sent at the end of the AS command.

The Active Scan is capable of returning up to 5 PanDescriptors in a scan. The actual scan time on each channel is measured as Time = [(2 ^ (SD Parameter)) * 15.36] ms. Total scan time is this time multiplied by the number of channels to be scanned (16 for the XBee, 12 for the XBee-PRO).

NOTE: Refer the scan table in the SD description to determine scan times. If using API Mode, no <CR>'s are returned in the response. Refer to the API Mode Operation section.

BD (Interface Data Rate) Command

<Serial Interfacing> The BD command is used to set and read the serial interface data rate used between the RF module and host. This parameter determines the rate at which serial data is sent to the module from the host. Modified interface data rates do not take effect until the CN (Exit AT Command Mode) command is issued and the system returns the 'OK' response.

When parameters 0-7 are sent to the module, the respective interface data rates are used (as shown in the table on the right).

The RF data rate is not affected by the BD parameter. If the interface data rate is set higher than the RF data rate, a flow control configuration may need to be implemented.

AT Command: ATBD

Parameter Range:0 – 7 (standard rates)
0x80-0x3D090 (non-standard rates up to 250 Kbps)

Parameter	Configuration (bps)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

Default Parameter Value:3

Non-standard Interface Data Rates:
Any value above 0x07 will be interpreted as an actual baud rate. When a value above 0x07 is sent, the closest interface data rate represented by the number is stored in the BD register. For example, a rate of 19200 bps can be set by sending the following command line "ATBD4B00". NOTE: When using Digi's X-CTU Software, non-standard interface data rates can only be set and read using the X-CTU 'Terminal' tab. Non-standard rates are not accessible through the 'Modem Configuration' tab.

When the BD command is sent with a non-standard interface data rate, the UART will adjust to accommodate the requested interface rate. In most cases, the clock resolution will cause the stored BD parameter to vary from the parameter that was sent (refer to the table below). Reading the BD command (send "ATBD" command without an associated parameter value) will return the value actually stored in the module's BD register.

Parameters Sent Versus Parameters Stored

BD Parameter Sent (HEX)	Interface Data Rate (bps)	BD Parameter Stored (HEX)
0	1200	0
4	19,200	4
7	115,200*	7
12C	300	12B
1C200	115,200	1B207

* The 115,200 baud rate setting is actually at 111,111 baud (-3.5% target UART speed).

CA (CCA Threshold) Command

<RF Interfacing> CA command is used to set and read CCA (Clear Channel Assessment) thresholds.

Prior to transmitting a packet, a CCA is performed to detect energy on the transmit channel. If the detected energy is above the CCA Threshold, the RF module will not transmit the packet.

AT Command: ATCA

Parameter Range: 0 – 0x50 [-dBm]

Default Parameter Value: 0x2C (-44 decimal dBm)

Minimum Firmware Version Required: v1.x80

CC (Command Sequence Character) Command

<AT Command Mode Options> The CC command is used to set and read the ASCII character used between guard times of the AT Command Mode Sequence (GT + CC + GT). This sequence enters the RF module into AT Command Mode so that data entering the module from the host is recognized as commands instead of payload.

AT Command: ATCC
Parameter Range: 0 – 0xFF
Default Parameter Value: 0x2B (ASCII "+")
Related Command: GT (Guard Times)

The AT Command Sequence is explained further in the AT Command Mode section.

CE (Coordinator Enable) Command

<Networking {Association} The CE command is used to set and read the behavior (End Device vs. Coordinator) of the RF module.

AT Command: ATCE

Parameter Range:0 – 1

Parameter	Configuration
0	End Device
1	Coordinator

Default Parameter Value:0

Minimum Firmware Version Required: v1.x80

CH (Channel) Command

<Networking {Addressing}> The CH command is used to set/read the operating channel on which RF connections are made between RF modules. The channel is one of three addressing options available to the module. The other options are the PAN ID (ID command) and destination addresses (DL & DH commands).

AT Command: ATCH
Parameter Range: 0x0B – 0x1A (XBee) 0x0C – 0x17 (XBee-PRO)
Default Parameter Value: 0x0C (12 decimal)
Related Commands: ID (PAN ID), DL (Destination Address Low, DH (Destination Address High)

In order for modules to communicate with each other, the modules must share the same channel number. Different channels can be used to prevent modules in one network from listening to transmissions of another. Adjacent channel rejection is 23 dB.

The module uses channel numbers of the 802.15.4 standard.

Center Frequency = 2.405 + (CH - 11d) * 5 MHz (d = decimal)

Refer to the XBee/XBee-PRO Addressing section for more information.

CN (Exit Command Mode) Command

<AT Command Mode Options> The CN command is used to explicitly exit the RF module from AT Command Mode.

AT Command: ATCN

CT (Command Mode Timeout) Command

<AT Command Mode Options> The CT command is used to set and read the amount of inactive time that elapses before the RF module automatically exits from AT Command Mode and returns to Idle Mode.

Use the CN (Exit Command Mode) command to exit AT Command Mode manually.

AT Command: ATCT
Parameter Range:2 – 0xFFFF [x 100 milliseconds]
Default Parameter Value: 0x64 (100 decimal (which equals 10 decimal seconds))
Number of bytes returned: 2
Related Command: CN (Exit Command Mode)

D0 - D4 (DIOn Configuration) Commands

<I/O Settings> The D0, D1, D2, D3 and D4 commands are used to select/read the behavior of their respective AD/DIO lines (pins 20, 19, 18, 17 and 11 respectively).

Options include:

- Analog-to-digital converter
- Digital input
- Digital output

AT Commands: ATD0, ATD1, ATD2, ATD3, ATD4

Parameter Range:0 – 5

Parameter	Configuration
0	Disabled
1	n/a
2	ADC
3	DI
4	DO low
5	DO high

Default Parameter Value:0

Minimum Firmware Version Required: 1.x.A0

D5 (DIO5 Configuration) Command

<I/O Settings> The D5 command is used to select/read the behavior of the DIO5 line (pin 15).

Options include:

- Associated Indicator (LED blinks when the module is associated)
- Analog-to-digital converter
- Digital input
- Digital output

AT Command: ATD5

Parameter Range:0 – 5

Parameter	Configuration
0	Disabled
1	Associated Indicator
2	ADC
3	DI
4	DO low
5	DO high

Default Parameter Value:1

Parameters 2-5 supported as of firmware version 1.xA0

D6 (DIO6 Configuration) Command

<I/O Settings> The D6 command is used to select/read the behavior of the DIO6 line (pin 16). Options include:

- RTS flow control
- Analog-to-digital converter
- Digital input
- Digital output

AT Command: ATD6

Parameter Range:0 – 5

Parameter	Configuration
0	Disabled
1	RTS Flow Control
2	n/a
3	DI
4	DO low
5	DO high

Default Parameter Value:0

Parameters 3-5 supported as of firmware version 1.xA0

D7 (DIO7 Configuration) Command

<I/O Settings> The D7 command is used to select/read the behavior of the DIO7 line (pin 12). Options include:

- CTS flow control
- Analog-to-digital converter
- Digital input
- Digital output
- RS485 TX Enable (this output is 3V CMOS level, and is useful in a 3V CMOS to RS485 conversion circuit)

AT Command: ATD7

Parameter Range:0 – 5

Parameter	Configuration
0	Disabled
1	CTS Flow Control
2	n/a
3	DI
4	DO low
5	DO high
6	RS485 TX Enable Low
7	RS485 TX Enable High

Default Parameter Value:1

Parameters 3-7 supported as of firmware version 1.x.A0

D8 (DI8 Configuration) Command

<I/O Settings> The D8 command is used to select/read the behavior of the DI8 line (pin 9). This command enables configuring the pin to function as a digital input. This line is also used with Pin Sleep.

AT Command: ATD8

Parameter Range:0 – 5
(1, 2, 4 & 5 n/a)

Parameter	Configuration
0	Disabled
3	DI

Default Parameter Value:0

Minimum Firmware Version Required: 1.xA0

DA (Force Disassociation) Command

<(Special)> The DA command is used to immediately disassociate an End Device from a Coordinator and reattempt to associate.

AT Command: ATDA

Minimum Firmware Version Required: v1.x80

DB (Received Signal Strength) Command

<Diagnostics> DB parameter is used to read the received signal strength (in dBm) of the last RF packet received. Reported values are accurate between -40 dBm and the RF module's receiver sensitivity.

AT Command: ATDB

Parameter Range [read-only]:
0x17-0x5C (XBee), 0x24-0x64 (XBee-PRO)

Absolute values are reported. For example: 0x58 = -88 dBm (decimal). If no packets have been received (since last reset, power cycle or sleep event), "0" will be reported.

DH (Destination Address High) Command

<Networking {Addressing}> The DH command is used to set and read the upper 32 bits of the RF module's 64-bit destination address. When combined with the DL (Destination Address Low) parameter, it defines the destination address used for transmission.

An module will only communicate with other modules having the same channel (CH parameter), PAN ID (ID parameter) and destination address (DH + DL parameters).

AT Command: ATDH

Parameter Range: 0 – 0xFFFFFFFF

Default Parameter Value: 0

Related Commands: DL (Destination Address Low), CH (Channel), ID (PAN VID), MY (Source Address)

To transmit using a 16-bit address, set the DH parameter to zero and the DL parameter less than 0xFFFF. 0x000000000000FFFF (DL concatenated to DH) is the broadcast address for the PAN.

Refer to the XBee/XBee-PRO Addressing section for more information.

DL (Destination Address Low) Command

<Networking {Addressing}> The DL command is used to set and read the lower 32 bits of the RF module's 64-bit destination address. When combined with the DH (Destination Address High) parameter, it defines the destination address used for transmission.

AT Command: ATDL
Parameter Range: 0 - 0xFFFFFFFF
Default Parameter Value: 0
Related Commands: DH (Destination Address High), CH (Channel), ID (PAN VID), MY (Source Address)

A module will only communicate with other modules having the same channel (CH parameter), PAN ID (ID parameter) and destination address (DH + DL parameters).

To transmit using a 16-bit address, set the DH parameter to zero and the DL parameter less than 0xFFFF. 0x000000000000FFFF (DL concatenated to DH) is the broadcast address for the PAN.

Refer to the XBee/XBee-PRO Addressing section for more information.

DN (Destination Node) Command

<Networking {Identification}> The DN command is used to resolve a NI (Node Identifier) string to a physical address. The following events occur upon successful command execution:

AT Command: ATDN
Parameter Range: 20-character ASCII String
Minimum Firmware Version Required: v1.x80

1. DL and DH are set to the address of the module with the matching NI (Node Identifier).
2. 'OK' is returned.
3. RF module automatically exits AT Command Mode.

If there is no response from a modem within 200 msec or a parameter is not specified (left blank), the command is terminated and an 'ERROR' message is returned.

DP (Disassociation Cyclic Sleep Period) Command

<Sleep Mode (Low Power)>

NonBeacon Firmware

End Device - The DP command is used to set and read the time period of sleep for cyclic sleeping remotes that are configured for Association but are not associated to a Coordinator. (i.e. If a device is configured to associate, configured as a Cyclic Sleep remote, but does not find a Coordinator; it will sleep for DP time before reattempting association.) Maximum sleep period is 268 seconds (0x68B0). DP should be > 0 for NonBeacon systems.

AT Command: ATDP
Parameter Range: 1 – 0x68B0 [x 10 milliseconds]
Default Parameter Value:0x3E8 (1000 decimal)
Related Commands: SM (Sleep Mode), SP (Cyclic Sleep Period), ST (Time before Sleep)
Minimum Firmware Version Required: v1.x80

EA (ACK Failures) Command

<Diagnostics> The EA command is used to reset and read the count of ACK (acknowledgement) failures. This parameter value increments when the module expires its transmission retries without receiving an ACK on a packet transmission. This count saturates at its maximum value.

AT Command: ATEA
Parameter Range:0 – 0xFFFF
Minimum Firmware Version Required: v1.x80

Set the parameter to "0" to reset count.

EC (CCA Failures) Command

<Diagnostics> The EC command is used to read and reset the count of CCA (Clear Channel Assessment) failures. This parameter value increments when the RF module does not transmit a packet due to the detection of energy that is above the CCA threshold level (set with CA command). This count saturates at its maximum value.

Set the EC parameter to "0" to reset count.

AT Command: ATEC
Parameter Range:0 – 0xFFFF
Related Command: CA (CCA Threshold)
Minimum Firmware Version Required: v1.x80

ED (Energy Scan) Command

<Networking {Association}> The ED command is used to send an "Energy Detect Scan". This parameter determines the length of scan on each channel. The maximal energy on each channel is returned and each value is followed by a carriage return. An additional carriage return is sent at the end of the command.

AT Command: ATED
Parameter Range:0 – 6
Related Command: SD (Scan Duration), SC (Scan Channel)
Minimum Firmware Version Required: v1.x80

The values returned represent the detected energy level in units of -dBm. The actual scan time on each channel is measured as Time = [(2 ^ ED PARAM) * 15.36] ms.

Note: Total scan time is this time multiplied by the number of channels to be scanned. Also refer to the SD (Scan Duration) table. Use the SC (Scan Channel) command to choose which channels to scan.

EE (AES Encryption Enable) Command

<Networking {Security}> The EE command is used to set/read the parameter that disables/ enables 128-bit AES encryption.

The XBee®/XBee-PRO® firmware uses the 802.15.4 Default Security protocol and uses AES encryption with a 128-bit key. AES encryption dictates that all modules in the network use the same key and the maximum RF packet size is 95 Bytes.

AT Command: ATEE

Parameter Range:0 – 1

Parameter	Configuration
0	Disabled
1	Enabled

Default Parameter Value:0

Related Commands: KY (Encryption Key), AP (API Enable), MM (MAC Mode)

Minimum Firmware Version Required: v1.xA0

When encryption is enabled, the module will always use its 64-bit long address as the source address for RF packets. This does not affect how the MY (Source Address), DH (Destination Address High) and DL (Destination Address Low) parameters work

If MM (MAC Mode) > 0 and AP (API Enable) parameter > 0:
With encryption enabled and a 16-bit short address set, receiving modules will only be able to issue RX (Receive) 64-bit indicators. This is not an issue when MM = 0.

If a module with a non-matching key detects RF data, but has an incorrect key: When encryption is enabled, non-encrypted RF packets received will be rejected and will not be sent out the UART.

Transparent Operation --> All RF packets are sent encrypted if the key is set.

API Operation --> Receive frames use an option bit to indicate that the packet was encrypted.

FP (Force Poll) Command

<Networking (Association)> The FP command is used to request indirect messages being held by a Coordinator.

AT Command: ATFP
Minimum Firmware Version Required: v1.x80

FR (Software Reset) Command

<Special> The FR command is used to force a software reset on the RF module. The reset simulates powering off and then on again the module.

AT Command: ATFR
Minimum Firmware Version Required: v1.x80

GT (Guard Times) Command

<AT Command Mode Options> GT Command is used to set the DI (data in from host) time-of-silence that surrounds the AT command sequence character (CC Command) of the AT Command Mode sequence (GT + CC + GT).

The DI time-of-silence is used to prevent inadvertent entrance into AT Command Mode.

Refer to the Command Mode section for more information regarding the AT Command Mode Sequence.

AT Command: ATGT
Parameter Range:2 – 0x0CE4 [x 1 millisecond]
Default Parameter Value:0x3E8 (1000 decimal)
Related Command: CC (Command Sequence Character)

HV (Hardware Version) Command

<Diagnostics> The HV command is used to read the hardware version of the RF module.

AT Command: ATHV
Parameter Range:0 – 0xFFFF [Read–only]
Minimum Firmware Version Required: v1.x80

IA (I/O Input Address) Command

<I/O Settings {I/O Line Passing}> The IA command is used to bind a module output to a specific address. Outputs will only change if received from this address. The IA command can be used to set/read both 16 and 64-bit addresses.

Setting all bytes to 0xFF will not allow the reception of any I/O packet to change outputs. Setting the IA address to 0xFFFF will cause the module to accept all I/O packets.

AT Command: ATIA
Parameter Range:0 – 0xFFFFFFFFFFFFFFFF
Default Parameter Value:0xFFFFFFFFFFFFFFFF (will not allow any received I/O packet to change outputs)
Minimum Firmware Version Required: v1.xA0

IC (DIO Change Detect) Command

<I/O Settings> Set/Read bitfield values for change detect monitoring. Each bit enables monitoring of DIO0 - DIO7 for changes.

If detected, data is transmitted with DIO data only. Any samples queued waiting for transmission will be sent first.

AT Command: ATIC
Parameter Range:0 – 0xFF [bitfield]
Default Parameter Value:0 (disabled)
Minimum Firmware Version Required: 1.xA0

Refer to the “ADC and Digital I/O Line Support” sections of the “RF Module Operations” chapter for more information.

ID (Pan ID) Command

<Networking {Addressing}> The ID command is used to set and read the PAN (Personal Area Network) ID of the RF module. Only modules with matching PAN IDs can communicate with each other. Unique PAN IDs enable control of which RF packets are received by a module.

AT Command: ATID
Parameter Range: 0 – 0xFFFF
Default Parameter Value:0x3332 (13106 decimal)

Setting the ID parameter to 0xFFFF indicates a global transmission for all PANs. It does not indicate a global receive.

IO (Digital Output Level) Command

<I/O Settings> The IO command is used to set digital output levels. This allows DIO lines setup as outputs to be changed through Command Mode.

AT Command: ATIO
Parameter Range: 8-bit bitmap (where each bit represents the level of an I/O line that is setup as an output.)
Minimum Firmware Version Required: v1.xA0

IR (Sample Rate) Command

<I/O Settings> The IR command is used to set/ read the sample rate. When set, the module will sample all enabled DIO/ADC lines at a specified interval. This command allows periodic reads of the ADC and DIO lines in a non-Sleep Mode setup. A sample rate which requires transmissions at a rate greater than once every 20ms is not recommended.

AT Command: ATIR
Parameter Range: 0 - 0xFFFF [x 1 msec] (cannot guarantee 1 ms timing when IT=1)
Default Parameter Value:0
Related Command: IT (Samples before TX)
Minimum Firmware Version Required: v1.xA0

Example: When IR = 0x14, the sample rate is 20 ms (or 50 Hz).

IS (Force Sample) Command

<I/O Settings> The IS command is used to force a read of all enabled DIO/ADC lines. The data is returned through the UART.

When operating in Transparent Mode (AP=0), the data is retuned in the following format:

AT Command: ATIS
Parameter Range: 1 - 0xFF
Default Parameter Value:1
Minimum Firmware Version Required: v1.xA0

All bytes are converted to ASCII:
number of samples<CR>
channel mask<CR>
DIO data<CR> (If DIO lines are enabled<CR>
ADC channel Data<cr> <-This will repeat for every enabled ADC channel<CR>
<CR> (end of data noted by extra <CR>)

When operating in API mode (AP > 0), the command will immediately return an 'OK' response. The data will follow in the normal API format for DIO data.

IT (Samples before TX) Command

<I/O Settings> The IT command is used to set/ read the number of DIO and ADC samples to collect before transmitting data.

One ADC sample is considered complete when all enabled ADC channels have been read. The module can buffer up to 93 Bytes of sample data.

AT Command: ATIT
Parameter Range: 1 - 0xFF
Default Parameter Value:1
Minimum Firmware Version Required: v1.xA0

Since the module uses a 10-bit A/D converter, each sample uses two Bytes. This leads to a maximum buffer size of 46 samples or IT=0x2E.

When Sleep Modes are enabled and IR (Sample Rate) is set, the module will remain awake until IT samples have been collected.

IU (I/O Output Enable) Command

<I/O Settings> The IU command is used to disable/enable I/O UART output. When enabled (IU = 1), received I/O line data packets are sent out the UART. The data is sent using an API frame regardless of the current AP parameter value.

AT Command: ATIU	
Parameter Range:0 – 1	
Parameter	Configuration
0	Disabled – Received I/O line data packets will be NOT sent out UART.
1	Enabled – Received I/O line data will be sent out UART
Default Parameter Value:1	
Minimum Firmware Version Required: 1.xA0	

KY (AES Encryption Key) Command

<Networking {Security}> The KY command is used to set the 128-bit AES (Advanced Encryption Standard) key for encrypting/decrypting data. Once set, the key cannot be read out of the module by any means.

AT Command: ATKY
Parameter Range:0 – (any 16-Byte value)
Default Parameter Value:0
Related Command: EE (Encryption Enable)
Minimum Firmware Version Required: v1.xA0

The entire payload of the packet is encrypted using the key and the CRC is computed across the ciphertext. When encryption is enabled, each packet carries an additional 16 Bytes to convey the random CBC Initialization Vector (IV) to the receiver(s). The KY value may be "0" or any 128-bit value. Any other value, including entering KY by itself with no parameters, is invalid. All ATKY entries (valid or not) are received with a returned 'OK'.

A module with the wrong key (or no key) will receive encrypted data, but the data driven out the serial port will be meaningless. A module with a key and encryption enabled will receive data sent from a module without a key and the correct unencrypted data output will be sent out the serial port. Because CBC mode is utilized, repetitive data appears differently in different transmissions due to the randomly-generated IV.

When queried, the system will return an 'OK' message and the value of the key will not be returned.

M0 (PWM0 Output Level) Command

<I/O Settings> The M0 command is used to set/read the output level of the PWM0 line (pin 6).

Before setting the line as an output:
1. Enable PWM0 output (P0 = 2)
2. Apply settings (use CN or AC)

The PWM period is 64 µsec and there are 0x03FF (1023 decimal) steps within this period. When M0 = 0 (0% PWM), 0x01FF (50% PWM), 0x03FF (100% PWM), etc.

AT Command: ATM0
Parameter Range:0 – 0x03FF [steps]
Default Parameter Value:0
Related Commands: P0 (PWM0 Enable), AC (Apply Changes), CN (Exit Command Mode)
Minimum Firmware Version Required: v1.xA0

M1 (PWM1 Output Level) Command

<I/O Settings> The M1 command is used to set/read the output level of the PWM1 line (pin 7).

Before setting the line as an output:
1. Enable PWM1 output (P1 = 2)
2. Apply settings (use CN or AC)

AT Command: ATM1
Parameter Range:0 – 0x03FF
Default Parameter Value:0
Related Commands: P1 (PWM1 Enable), AC (Apply Changes), CN (Exit Command Mode)
Minimum Firmware Version Required: v1.xA0

MM (MAC Mode) Command

<Networking {Addressing}> The MM command is used to set and read the MAC Mode value. The MM command disables/enables the use of a Digi header contained in the 802.15.4 RF packet. By default (MM = 0), Digi Mode is enabled and the module adds an extra header to the data portion of the 802.15.4 packet. This enables the following features:

- ND and DN command support
- Duplicate packet detection when using ACKs
- "RR command
- "DIO/AIO sampling support

The MM command allows users to turn off the use of the extra header. Modes 1 and 2 are strict 802.15.4 modes. If the Digi header is disabled, ND and DN parameters are also disabled.

Note: When MM=0 or 3, application and CCA failure retries are not supported.

AT Command: ATMM

Parameter Range:0 – 3

Parameter	Configuration
0	Digi Mode (802.15.4 + Digi header)
1	802.15.4 (no ACKs)
2	802.15.4 (with ACKs)
3	Digi Mode (no ACKs)

Default Parameter Value:0

Related Commands: ND (Node Discover), DN (Destination Node)

Minimum Firmware Version Required: v1.x80

MY (16-bit Source Address) Command

<Networking {Addressing}> The MY command is used to set and read the 16-bit source address of the RF module.

By setting MY to 0xFFFF, the reception of RF packets having a 16-bit address is disabled. The 64-bit address is the module's serial number and is always enabled.

AT Command: ATMY

Parameter Range: 0 – 0xFFFF

Default Parameter Value: 0

Related Commands: DH (Destination Address High), DL (Destination Address Low), CH (Channel), ID (PAN ID)

NB (Parity) Command

<Serial Interfacing> The NB command is used to select/read the parity settings of the RF module for UART communications.

Note: the module does not actually calculate and check the parity; it only interfaces with devices at the configured parity and stop bit settings.

AT Command: ATNB

Parameter Range: 0 – 4

Parameter	Configuration
0	8-bit no parity
1	8-bit even
2	8-bit odd
3	8-bit mark
4	8-bit space

Default Parameter Value: 0

Number of bytes returned: 1

ND (Node Discover) Command

AT Command: ATND
Range: optional 20-character NI value
Related Commands: CH (Channel), ID (Pan ID), MY (Source Address), SH (Serial Number High), SL (Serial Number Low), NI (Node Identifier), NT (Node Discover Time)
Minimum Firmware Version Required: v1.x80

<Networking {Identification}> The ND command is used to discover and report all modules on its current operating channel (CH parameter) and PAN ID (ID parameter). ND also accepts an NI (Node Identifier) value as a parameter. In this case, only a module matching the supplied identifier will respond.

ND uses a 64-bit long address when sending and responding to an ND request. The ND command causes a module to transmit a globally addressed ND command packet. The amount of time allowed for responses is determined by the NT (Node Discover Time) parameter.

In AT Command mode, command completion is designated by a carriage return (0x0D). Since two carriage returns end a command response, the application will receive three carriage returns at the end of the command. If no responses are received, the application should only receive one carriage return. When in API mode, the application should receive a frame (with no data) and status (set to 'OK') at the end of the command. When the ND command packet is received, the remote sets up a random time delay (up to 2.2 sec) before replying as follows:

Node Discover Response (AT command mode format - Transparent operation):
MY (Source Address) value<CR>
SH (Serial Number High) value<CR>
SL (Serial Number Low) value<CR>
DB (Received Signal Strength) value<CR>
NI (Node Identifier) value<CR>
<CR> (This is part of the response and not the end of command indicator.)

Node Discover Response (API format - data is binary (except for NI)):
2 bytes for MY (Source Address) value
4 bytes for SH (Serial Number High) value
4 bytes for SL (Serial Number Low) value
1 byte for DB (Received Signal Strength) value
NULL-terminated string for NI (Node Identifier) value (max 20 bytes w/out NULL terminator)

NI (Node Identifier) Command

AT Command: ATNI
Parameter Range: 20-character ASCII string
Related Commands: ND (Node Discover), DN (Destination Node)
Minimum Firmware Version Required: v1.x80

<Networking {Identification}> The NI command is used to set and read a string for identifying a particular node.

Rules:

- Register only accepts printable ASCII data.
- A string can not start with a space.
- A carriage return ends command
- Command will automatically end when maximum bytes for the string have been entered.

This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command.

NO (Node Discover Options) Command

AT Command: ATNO
Parameter Range: "0-1
Related Commands: ND (Node Discover), DN (Destination Node)
Minimum Firmware Version Required: v1.xC5

<Networking {Identification}> The NO command is used to suppress/include a self-response to Node Discover commands. When NO=1 a module doing a Node Discover will include a response entry for itself.

NT (Node Discover Time) Command

<Networking {Identification}> The NT command is used to set the amount of time a base node will wait for responses from other nodes when using the ND (Node Discover) command. The NT value is transmitted with the ND command.

Remote nodes will set up a random hold-off time based on this time. The remotes will adjust this time down by 250 ms to give each node the ability to respond before the base ends the command. Once the ND command has ended, any response received on the base will be discarded.

AT Command: ATNT
Parameter Range: 0x01 – 0xFC [x 100 msec]
Default: 0x19 (2.5 decimal seconds)
Related Commands: ND (Node Discover)
Minimum Firmware Version Required: 1.xA0

P0 (PWM0 Configuration) Command

<I/O Setting {I/O Line Passing}> The P0 command is used to select/read the function for PWM0 (Pulse Width Modulation output 0). This command enables the option of translating incoming data to a PWM so that the output can be translated back into analog form.

With the IA (I/O Input Address) parameter correctly set, AD0 values can automatically be passed to PWM0.

AT Command: ATP0
The second character in the command is the number zero (“0”), not the letter “O”.

Parameter Range: 0 – 2

Parameter	Configuration
0	Disabled
1	RSSI
2	PWM0 Output

Default Parameter Value: 1

P1 (PWM1 Configuration) Command

<I/O Setting {I/O Line Passing}> The P1 command is used to select/read the function for PWM1 (Pulse Width Modulation output 1). This command enables the option of translating incoming data to a PWM so that the output can be translated back into analog form.

With the IA (I/O Input Address) parameter correctly set, AD1 values can automatically be passed to PWM1.

AT Command: ATP1

Parameter Range: 0 – 2

Parameter	Configuration
0	Disabled
1	RSSI
2	PWM1 Output

Default Parameter Value: 0

Minimum Firmware Version Required: v1.xA0

PL (Power Level) Command

<RF Interfacing> The PL command is used to select and read the power level at which the RF module transmits conducted power.

When operating in Europe, XBee-PRO 802.15.4 modules must operate at or below a transmit power output level of 10dBm. Customers have 2 choices for transmitting at or below 10dBm:

- Order the standard XBee-PRO module and change the PL command to "0" (10dBm),
- Order the International variant of the XBee-PRO module, which has a maximum transmit output power of 10dBm.

AT Command: ATPL

Parameter Range: 0 – 4

Parameter	XBee	XBee-PRO	XBee-PRO International variant
0	-10 dBm	10 dBm	PL=4: 10 dBm
1	-6 dBm	12 dBm	PL=3: 8 dBm
2	-4 dBm	14 dBm	PL=2: 2 dBm
3	-2 dBm	16 dBm	PL=1: -3 dBm
4	0 dBm	18 dBm	PL=0: -3 dBm

Default Parameter Value: 4

PR (Pull-up Resistor) Command

<Serial Interfacing> The PR command is used to set and read the bit field that is used to configure internal the pull-up resistor status for I/O lines. "1" specifies the pull-up resistor is enabled. "0" specifies no pull up.

AT Command: ATPR
Parameter Range: 0 – 0xFF
Default Parameter Value: 0xFF (all pull-up resistors are enabled)
Minimum Firmware Version Required: v1.x80

bit 0 - AD4/DIO4 (pin 11)
bit 1 - AD3/DIO3 (pin 17)
bit 2 - AD2/DIO2 (pin 18)
bit 3 - AD1/DIO1 (pin 19)
bit 4 - AD0/DIO0 (pin 20)
bit 5 - AD6/DIO6 (pin 16)
bit 6 - DI8 (pin 9)
bit 7 - DIN/CONFIG (pin 3)

For example: Sending the command "ATPR 6F" will turn bits 0, 1, 2, 3, 5 and 6 ON; and bits 4 & 7 will be turned OFF. (The binary equivalent of "0x6F" is "01101111". Note that 'bit 0' is the last digit in the bitfield.

PT (PWM Output Timeout) Command

<I/O Settings {I/O Line Passing}> The PT command is used to set/read the output timeout value for both PWM outputs.

When PWM is set to a non-zero value: Due to I/O line passing, a time is started which when expired will set the PWM output to zero. The timer is reset when a valid I/O packet is received.

AT Command: ATPT
Parameter Range: 0 – 0xFF [x 100 msec]
Default Parameter Value: 0xFF
Minimum Firmware Version Required: 1.xA0

RE (Restore Defaults) Command

AT Command: ATRE

<(Special)> The RE command is used to restore all configurable parameters to their factory default settings. The RE command does not write restored values to non-volatile (persistent) memory. Issue the WR (Write) command subsequent to issuing the RE command to save restored parameter values to non-volatile memory.

RN (Random Delay Slots) Command

<Networking & Security> The RN command is used to set and read the minimum value of the back-off exponent in the CSMA-CA algorithm. The CSMA-CA algorithm was engineered for collision avoidance (random delays are inserted to prevent data loss caused by data collisions).

AT Command: ATRN
Parameter Range: 0 – 3 [exponent]
Default Parameter Value: 0

If RN = 0, collision avoidance is disabled during the first iteration of the algorithm (802.15.4 - macMinBE).

CSMA-CA stands for "Carrier Sense Multiple Access - Collision Avoidance". Unlike CSMA-CD (reacts to network transmissions after collisions have been detected), CSMA-CA acts to prevent data collisions before they occur. As soon as a module receives a packet that is to be transmitted, it checks if the channel is clear (no other module is transmitting). If the channel is clear, the packet is sent over-the-air. If the channel is not clear, the module waits for a randomly selected period of time, then checks again to see if the channel is clear. After a time, the process ends and the data is lost.

RO (Packetization Timeout) Command

<Serial Interfacing> RO command is used to set and read the number of character times of inter-character delay required before transmission.

RF transmission commences when data is detected in the DI (data in from host) buffer and RO character times of silence are detected on the UART receive lines (after receiving at least 1 byte).

AT Command: ATRO
Parameter Range:0 – 0xFF [x character times]
Default Parameter Value: 3

RF transmission will also commence after 100 Bytes (maximum packet size) are received in the DI buffer.

Set the RO parameter to '0' to transmit characters as they arrive instead of buffering them into one RF packet.

RP (RSSI PWM Timer) Command

<I/O Settings {I/O Line Passing}> The RP command is used to enable PWM (Pulse Width Modulation) output on the RF module. The output is calibrated to show the level a received RF signal is above the sensitivity level of the module. The PWM pulses vary from 24 to 100%. Zero percent means PWM output is inactive. One to 24% percent means the received RF signal is at or below the published sensitivity level of the module. The following table shows levels above sensitivity and PWM values.

AT Command: ATRP
Parameter Range:0 – 0xFF [x 100 msec]
Default Parameter Value: 0x28 (40 decimal)

The total period of the PWM output is 64 µs. Because there are 445 steps in the PWM output, the minimum step size is 144 ns.

PWM Percentages

dB above Sensitivity	PWM percentage (high period / total period)
10	41%
20	58%
30	75%

A non-zero value defines the time that the PWM output will be active with the RSSI value of the last received RF packet. After the set time when no RF packets are received, the PWM output will be set low (0 percent PWM) until another RF packet is received. The PWM output will also be set low at power-up until the first RF packet is received. A parameter value of 0xFF permanently enables the PWM output and it will always reflect the value of the last received RF packet.

RR (XBee Retries) Command

<Networking {Addressing}> The RR command is used set/read the maximum number of retries the module will execute in addition to the 3 retries provided by the 802.15.4 MAC. For each XBee retry, the 802.15.4 MAC can execute up to 3 retries.

AT Command: ATRR
Parameter Range: 0 – 6
Default: 0
Minimum Firmware Version Required: 1.xA0

This values does not need to be set on all modules for retries to work. If retries are enabled, the transmitting module will set a bit in the Digi RF Packet header which requests the receiving module to send an ACK (acknowledgement). If the transmitting module does not receive an ACK within 200 msec, it will re-send the packet within a random period up to 48 msec. Each XBee retry can potentially result in the MAC sending the packet 4 times (1 try plus 3 retries). Note that retries are not attempted for packets that are purged when transmitting with a Cyclic Sleep Coordinator.

SC (Scan Channels) Command

<Networking {Association}> The SC command is used to set and read the list of channels to scan for all Active and Energy Scans as a bit field.

This affects scans initiated in command mode [AS (Active Scan) and ED (Energy Scan) commands] and during End Device Association and Coordinator startup.

AT Command: ATSC
Parameter Range: 1-0xFFFF [Bitfield] (bits 0, 14, 15 are not allowed when using the XBee-PRO)
Default Parameter Value: 0x1FFE (all XBee-PRO channels)
Related Commands: ED (Energy Scan), SD (Scan Duration)
Minimum Firmware Version Required: v1.x80

bit 0 - 0x0B	bit 4 - 0x0F	bit 8 - 0x13	bit 12 - 0x17
bit 1 - 0x0C	bit 5 - 0x10	bit 9 - 0x14	bit 13 - 0x18
bit 2 - 0x0D	bit 6 - 0x11	bit 10 - 0x15	bit 14 - 0x19
bit 3 - 0x0E	bit 7 - 0x12	bit 11 - 0x16	bit 15 - 0x1A

SD (Scan Duration) Command

<Networking {Association}> The SD command is used to set and read the exponent value that determines the duration (in time) of a scan.

End Device (Duration of Active Scan during Association) - In a Beacon system, set SD = BE of the Coordinator. SD must be set at least to the highest BE parameter of any Beaconing Coordinator with which an End Device or Coordinator wish to discover.

AT Command: ATSD
Parameter Range: 0 - 0x0F
Default Parameter Value: 4
Related Commands: ED (Energy Scan), SC (Scan Channel)
Minimum Firmware Version Required: v1.x80

Coordinator - If the 'ReassignPANID' option is set on the Coordinator [refer to A2 parameter], the SD parameter determines the length of time the Coordinator will scan channels to locate existing PANs. If the 'ReassignChannel' option is set, SD determines how long the Coordinator will perform an Energy Scan to determine which channel it will operate on.

Scan Time is measured as ((# of Channels to Scan) * (2 ^ SD) * 15.36ms). The number of channels to scan is set by the SC command. The XBee RF Module can scan up to 16 channels (SC = 0xFFFF). The XBee PRO RF Module can scan up to 12 channels (SC = 0x1FFE).

Examples: Values below show results for a 12-channel scan

If SD = 0, time = 0.18 sec	SD = 8, time = 47.19 sec
SD = 2, time = 0.74 sec	SD = 10, time = 3.15 min
SD = 4, time = 2.95 sec	SD = 12, time = 12.58 min
SD = 6, time = 11.80 sec	SD = 14, time = 50.33 min

SH (Serial Number High) Command

<Diagnostics> The SH command is used to read the high 32 bits of the RF module's unique IEEE 64-bit address.

The module serial number is set at the factory and is read-only.

AT Command: ATSH
Parameter Range: 0 - 0xFFFFFFFF [read-only]
Related Commands: SL (Serial Number Low), MY (Source Address)

SL (Serial Number Low) Command

<Diagnostics> The SL command is used to read the low 32 bits of the RF module's unique IEEE 64-bit address.

The module serial number is set at the factory and is read-only.

AT Command: ATSL
Parameter Range: 0 - 0xFFFFFFFF [read-only]
Related Commands: SH (Serial Number High), MY (Source Address)

SM (Sleep Mode) Command

<Sleep Mode (Low Power)> The SM command is used to set and read Sleep Mode settings. By default, Sleep Modes are disabled (SM = 0) and the RF module remains in Idle/ Receive Mode. When in this state, the module is constantly ready to respond to either serial or RF activity.

* The Sleep Coordinator option (SM=6) only exists for backwards compatibility with firmware version 1.x06 only. In all other cases, use the CE command to enable a Coordinator.

AT Command: ATSM

Parameter Range: 0 – 6

Parameter	Configuration
0	Disabled
1	Pin Hibernate
2	Pin Doze
3	(reserved)
4	Cyclic Sleep Remote
5	Cyclic Sleep Remote (with Pin Wake-up)
6	Sleep Coordinator*

Default Parameter Value: 0

SO (Sleep Mode Command)

Sleep (Low Power) Sleep Options Set/Read the sleep mode options.

Bit 0 - Poll wakeup disable

- 0 - Normal operations. A module configured for cyclic sleep will poll for data on waking.
- 1 - Disable wakeup poll. A module configured for cyclic sleep will not poll for data on waking.

Bit 1 - ADC/DIO wakeup sampling disable.

- 0 - Normal operations. A module configured in a sleep mode with ADC/DIO sampling enabled will automatically perform a sampling on wakeup.
- 1 - Suppress sample on wakeup. A module configured in a sleep mode with ADC/DIO sampling enabled will not automatically sample on wakeup.

AT Command: ATSO	
Parameter Range:	0-4
Default Parameter Value:	
Related Commands: SM (Sleep Mode), ST (Time before Sleep), DP (Disassociation Cyclic Sleep Period, BE (Beacon Order)	

SP (Cyclic Sleep Period) Command

<Sleep Mode (Low Power)> The SP command is used to set and read the duration of time in which a remote RF module sleeps. After the cyclic sleep period is over, the module wakes and checks for data. If data is not present, the module goes back to sleep. The maximum sleep period is 268 seconds (SP = 0x68B0).

The SP parameter is only valid if the module is configured to operate in Cyclic Sleep (SM = 4-6). Coordinator and End Device SP values should always be equal.

To send Direct Messages, set SP = 0.

AT Command: ATSP	
Parameter Range:	NonBeacon Firmware: 0-0x68B0 [x 10 milliseconds]
Default Parameter Value:	
Related Commands: SM (Sleep Mode), ST (Time before Sleep), DP (Disassociation Cyclic Sleep Period, BE (Beacon Order)	

NonBeacon Firmware

End Device - SP determines the sleep period for cyclic sleeping remotes. Maximum sleep period is 268 seconds (0x68B0).

Coordinator - If non-zero, SP determines the time to hold an indirect message before discarding it. A Coordinator will discard indirect messages after a period of (2.5 * SP).

ST (Time before Sleep) Command

<Sleep Mode (Low Power)> The ST command is used to set and read the period of inactivity (no serial or RF data is sent or received) before activating Sleep Mode.

NonBeacon Firmware

Set/Read time period of inactivity (no serial or RF data is sent or received) before activating Sleep Mode. ST parameter is only valid with Cyclic Sleep settings (SM = 4 - 5).

Coordinator and End Device ST values must be equal.

AT Command: ATST	
Parameter Range:	NonBeacon Firmware: 1 – 0xFFFF [x 1 millisecond]
Default Parameter Value:	
Related Commands: SM (Sleep Mode), ST (Time before Sleep)	

T0 - T7 ((D0-D7) Output Timeout) Command

<I/O Settings {I/O Line Passing}> The T0, T1, T2, T3, T4, T5, T6 and T7 commands are used to set/read output timeout values for the lines that correspond with the D0 - D7 parameters. When output is set (due to I/O line passing) to a non-default level, a timer is started which when expired, will set the output to its default level. The timer is reset when a valid I/O packet is received. The Tn parameter defines the permissible amount of time to stay in a non-default (active) state. If Tn = 0, Output Timeout is disabled (output levels are held indefinitely).

AT Commands: ATT0 – ATT7
Parameter Range:0 – 0xFF [x 100 msec]
Default Parameter Value:0xFF
Minimum Firmware Version Required: v1.xA0

VL (Firmware Version - Verbose)

<Diagnostics> The VL command is used to read detailed version information about the RF module. The information includes:
application build date; MAC, PHY and bootloader versions; and build dates. This command was removed from firmware 1xC9 and later versions.

AT Command: ATVL
Parameter Range:0 – 0xFF [x 100 milliseconds]
Default Parameter Value: 0x28 (40 decimal)
Minimum Firmware Version Required: v1.x80 – v1.xC8

VR (Firmware Version) Command

<Diagnostics> The VR command is used to read which firmware version is stored in the module.

AT Command: ATVR
Parameter Range: 0 – 0xFFFF [read only]

XBee version numbers will have four significant digits. The reported number will show three or four numbers and is stated in hexadecimal notation. A version can be reported as "ABC" or "ABCD". Digits ABC are the main release number and D is the revision number from the main release. "D" is not required and if it is not present, a zero is assumed for D. "B" is a variant designator. The following variants exist:

- "0" = Non-Beacon Enabled 802.15.4 Code
- "1" = Beacon Enabled 802.15.4 Code

WR (Write) Command

<(Special)> The WR command is used to write configurable parameters to the RF module's non-volatile memory. Parameter values remain in the module's memory until overwritten by subsequent use of the WR Command.

AT Command: ATWR

If changes are made without writing them to non-volatile memory, the module reverts back to previously saved parameters the next time the module is powered-on.

NOTE: Once the WR command is sent to the module, no additional characters should be sent until after the “OK/r” response is received.

API Operation

By default, XBee®/XBee-PRO® RF Modules act as a serial line replacement (Transparent Operation) - all UART data received through the DI pin is queued up for RF transmission. When the module receives an RF packet, the data is sent out the DO pin with no additional information.

Inherent to Transparent Operation are the following behaviors:

- If module parameter registers are to be set or queried, a special operation is required for transitioning the module into Command Mode.
- In point-to-multipoint systems, the application must send extra information so that the receiving module(s) can distinguish between data coming from different remotes.

As an alternative to the default Transparent Operation, API (Application Programming Interface) Operations are available. API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame.

API Frame Specifications

Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

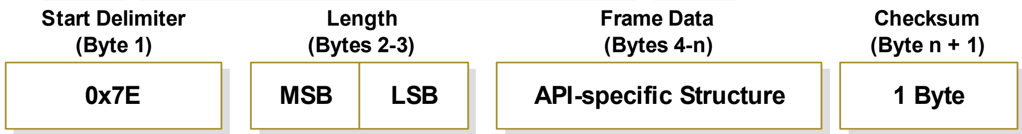
- AP = 0 (default): Transparent Operation (UART Serial line replacement)
  API modes are disabled.
- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the data is silently discarded.

API Operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 3-01. UART Data Frame Structure:

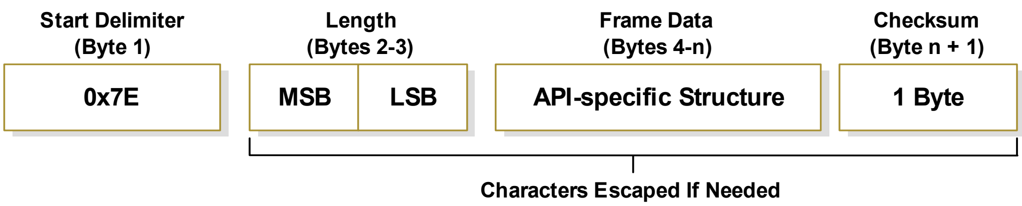


MSB = Most Significant Byte, LSB = Least Significant Byte

API Operation - with Escape Characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined as follows:

Figure 3-02. UART Data Frame Structure - with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters. When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the UART or UART data frame operation. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20.

Data bytes that need to be escaped:

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Example - Raw UART Data Frame (before escaping interfering bytes):
0x7E 0x00 0x02 0x23 0x11 0xCB

0x11 needs to be escaped which results in the following frame:
0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note: In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as: 0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB.

Checksum

To test data integrity, a checksum is calculated and verified on non-escaped data.

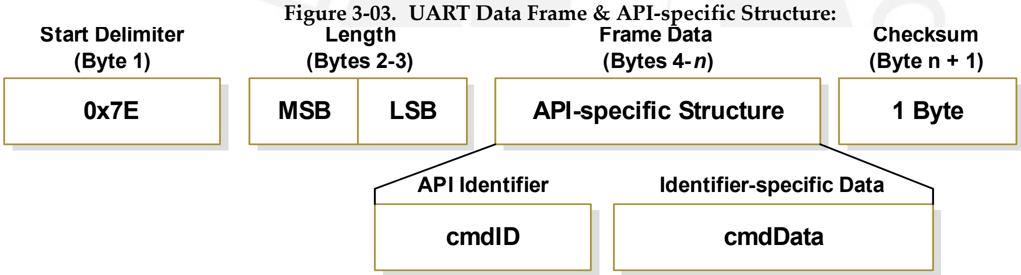
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract from 0xFF.

To verify: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

API Types

Frame data of the UART data frame forms an API-specific structure as follows:

Figure 3-03. UART Data Frame & API-specific Structure:

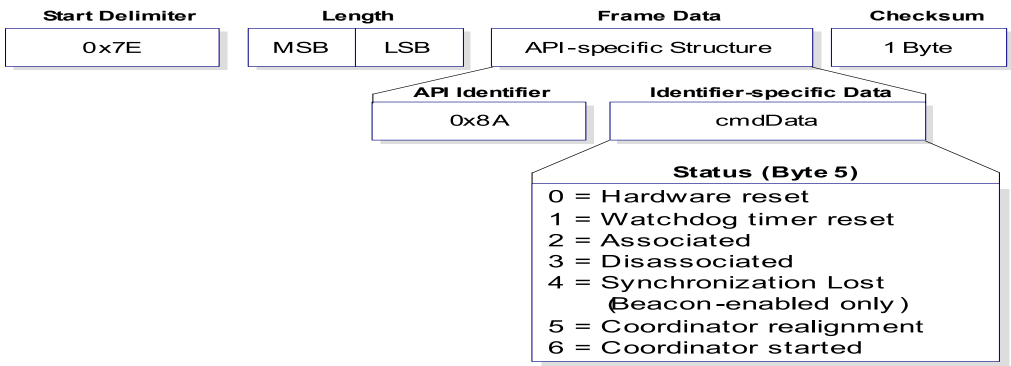


The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Refer to the sections that follow for more information regarding the supported API types. Note that multi-byte values are sent big endian.

Modem Status

API Identifier: 0x8A
RF module status messages are sent from the module in response to specific conditions.

Figure 3-04. Modem Status Frames



AT Command

API Identifier Value: 0x08

The “AT Command” API type allows for module parameters to be queried or set. When using this command ID, new parameter values are applied immediately. This includes any register set with the “AT Command - Queue Parameter Value” (0x09) API type.

Figure 3-05. AT Command Frames

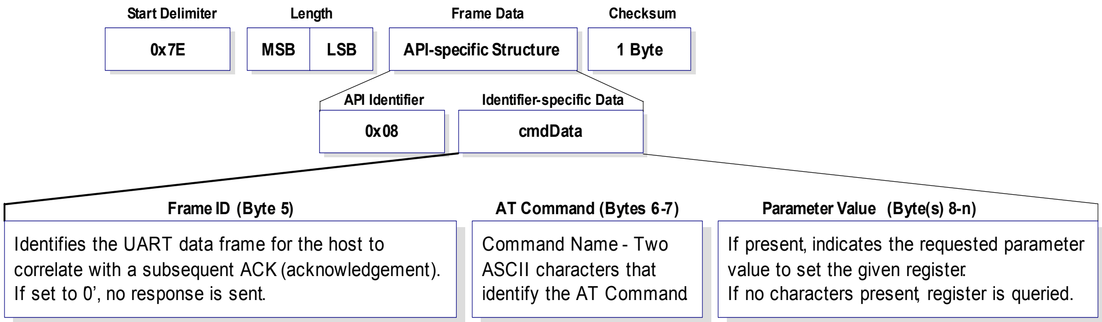


Figure 3-06. Example: API frames when reading the DL parameter value of the module.

Byte 1	Bytes 2-3		Byte 4	Byte 5	Bytes 6-7		Byte 8
0x7E	0x00	0x04	0x08	0x52 (R)	0x44 (D)	0x4C (L)	0x15
Start Delimiter	Length*		API Identifier	Frame ID**	AT Command		Checksum

** Length [Bytes] = API Identifier + Frame ID + AT Command*

*** “R” value was arbitrarily selected.*

Figure 3-07. Example: API frames when modifying the DL parameter value of the module.

Byte 1	Bytes 2-3		Byte 4	Byte 5	Bytes 6-7		Bytes 8-11	Byte 12
0x7E	0x00	0x08	0x08	0x4D (M)	0x44 (D)	0x4C (L)	0x00000FFF	0x0C
Start Delimiter	Length*		API Identifier	Frame ID**	AT Command		Parameter Value	Checksum

** Length [Bytes] = API Identifier + Frame ID + AT Command + Parameter Value*

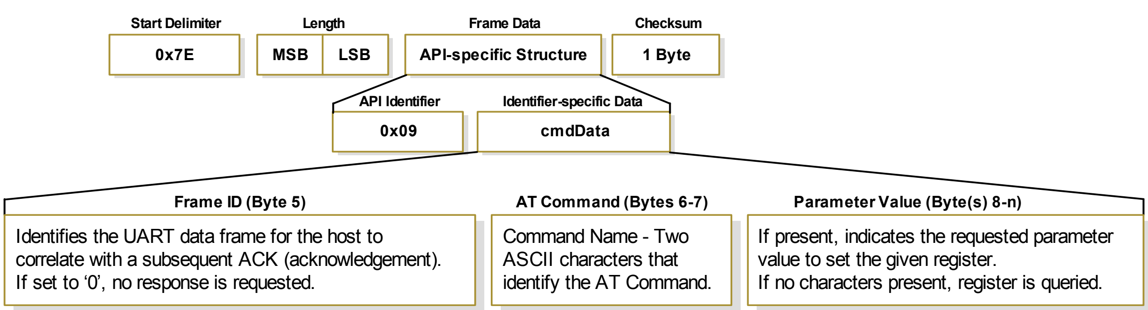
*** “M” value was arbitrarily selected.*

AT Command - Queue Parameter Value

API Identifier Value: 0x09

This API type allows module parameters to be queried or set. In contrast to the “AT Command” API type, new parameter values are queued and not applied until either the “AT Command” (0x08) API type or the AC (Apply Changes) command is issued. Register queries (reading parameter values) are returned immediately.

Figure 3-08. AT Command Frames
(Note that frames are identical to the “AT Command” API type except for the API identifier.)



AT Command Response

API Identifier Value: 0x88
Response to previous command.

In response to an AT Command message, the module will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) and AS (Active Scan) commands). These commands will end by sending a frame with a status of ATCMD_OK and no cmdData.

Figure 3-09. AT Command Response Frames.

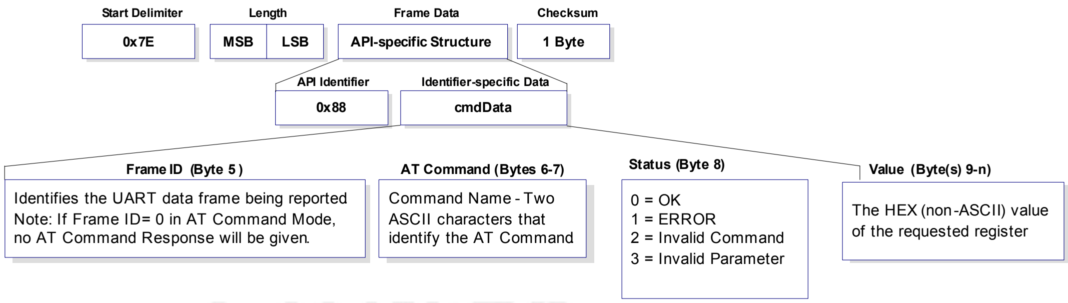
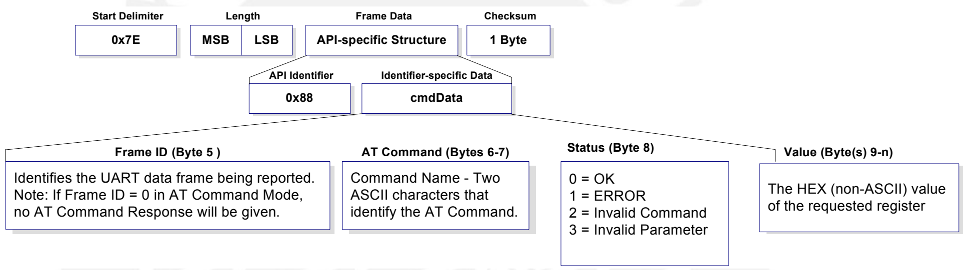


Figure 3-10. AT Command Response Frames.

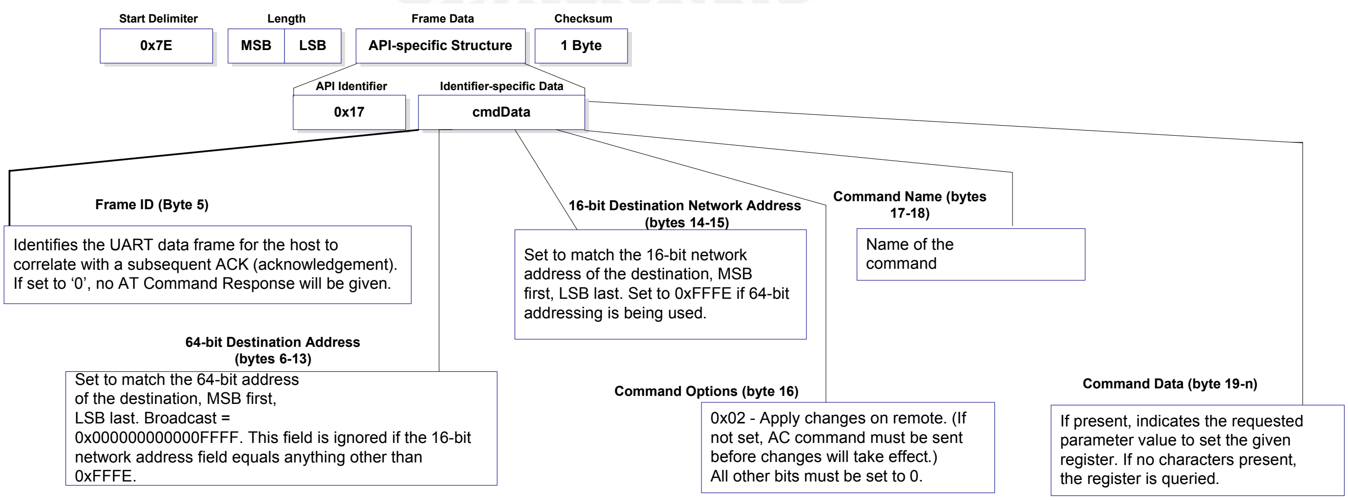


Remote AT Command Request

API Identifier Value: 0x17

Allows for module parameter registers on a remote device to be queried or set

Figure 3-11. Remote AT Command Request

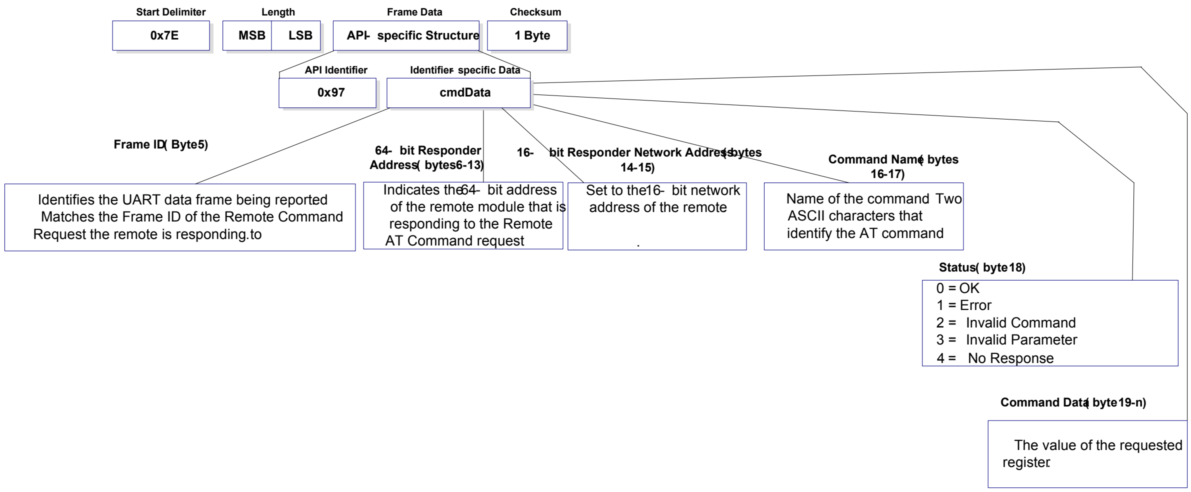


Remote Command Response

API Identifier Value: 0x97

If a module receives a remote command response RF data frame in response to a Remote AT Command Request, the module will send a Remote AT Command Response message out the UART. Some commands may send back multiple frames--for example, Node Discover (ND) command.

Figure 3-12. Remote AT Command Response.

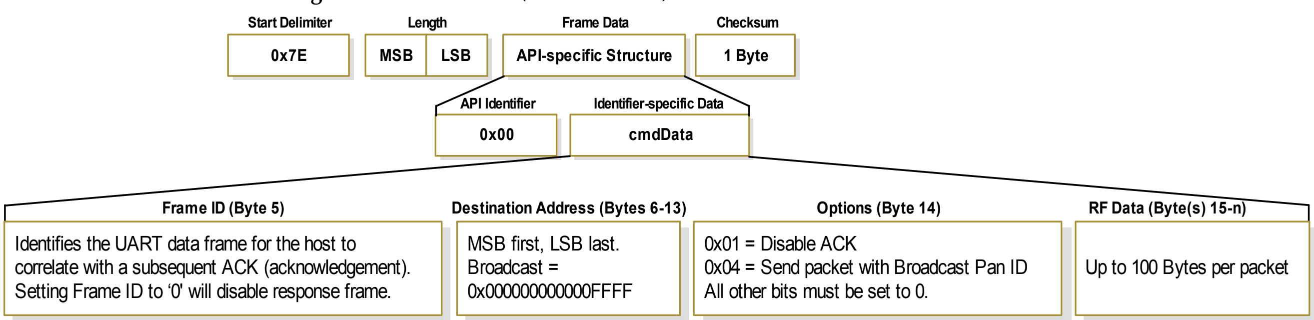


TX (Transmit) Request: 64-bit address

API Identifier Value: 0x00

A TX Request message will cause the module to send RF Data as an RF Packet.

Figure 3-13. TX Packet (64-bit address) Frames

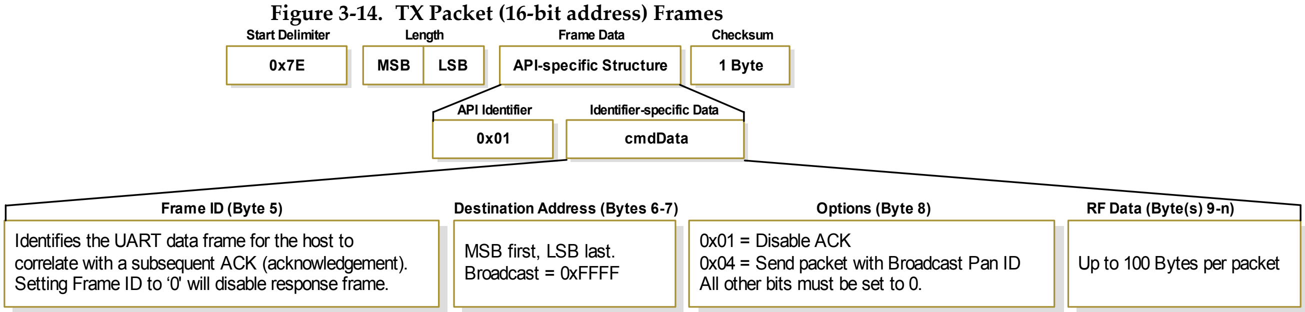


TX (Transmit) Request: 16-bit address

API Identifier Value: 0x01

A TX Request message will cause the module to send RF Data as an RF Packet.

Figure 3-14. TX Packet (16-bit address) Frames

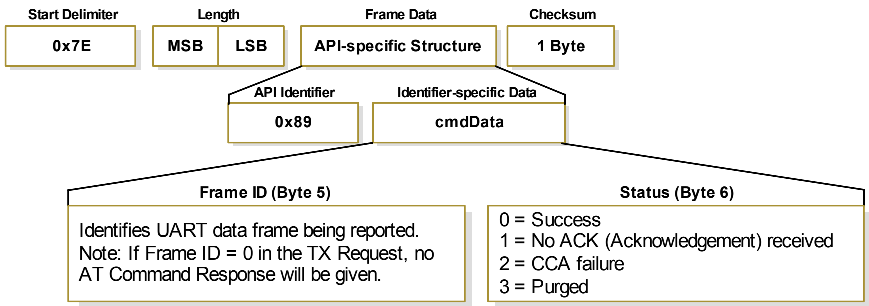


TX (Transmit) Status

API Identifier Value: 0x89

When a TX Request is completed, the module sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

Figure 3-15. TX Status Frames



NOTES:

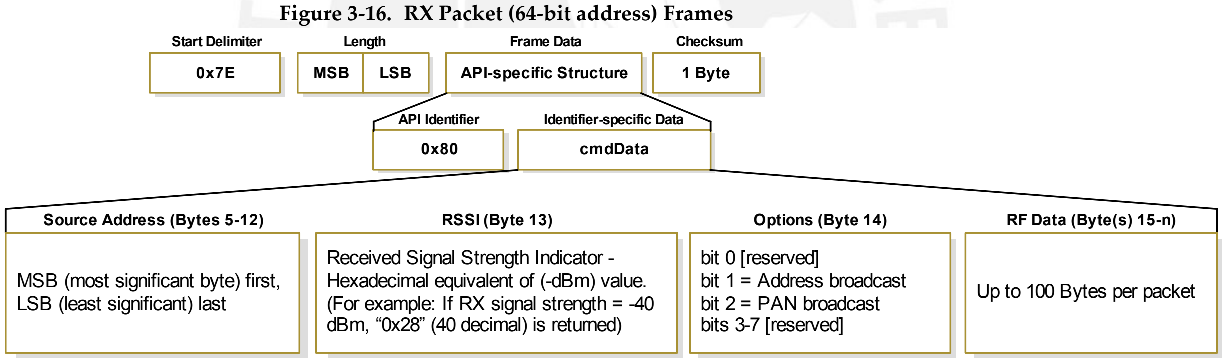
- "STATUS = 1" occurs when all retries are expired and no ACK is received.
- If transmitter broadcasts (destination address = 0x000000000000FFFF), only "STATUS = 0 or 2" will be returned.
- "STATUS = 3" occurs when Coordinator times out of an indirect transmission. Timeout is defined as (2.5 x SP (Cyclic Sleep Period) parameter value).

RX (Receive) Packet: 64-bit Address

API Identifier Value: 0x80

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-16. RX Packet (64-bit address) Frames

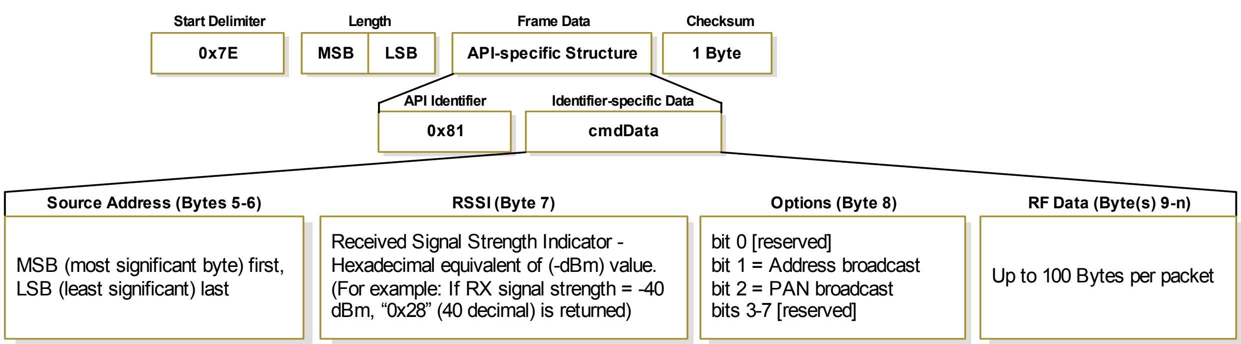


RX (Receive) Packet: 16-bit Address

API Identifier Value: 0x81

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-17. RX Packet (16-bit address) Frames



Appendix A: Agency Certifications

United States (FCC)

XBee®/XBee-PRO® RF Modules comply with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.

To fulfill FCC Certification requirements, the OEM must comply with the following regulations:

1. The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product [Figure A-01].
2. XBee®/XBee-PRO® RF Modules may only be used with antennas that have been tested and approved for use with this module [refer to the antenna tables in this section].

OEM Labeling Requirements

WARNING: The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product enclosure that displays the contents shown in the figure below.

Figure 4-01. Required FCC Label for OEM products containing the XBee®/XBee-PRO® RF Module

Contains FCC ID: OUR-XBEE/OUR-XBEEPRO**

The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i.*) this device may not cause harmful interference and (*ii.*) this device must accept any interference received, including interference that may cause undesired operation.

* The FCC ID for the XBee is "OUR-XBEE". The FCC ID for the XBee-PRO is "OUR-XBEEPRO".

FCC Notices

IMPORTANT: The XBee®/XBee-PRO® RF Module has been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Modifications not expressly approved by Digi could void the user's authority to operate the equipment.

IMPORTANT: OEMs must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC Rules.

IMPORTANT: The RF module has been certified for remote and base radio applications. If the module will be used for portable applications, the device must undergo SAR testing.

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 and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: Re-orient or relocate the receiving antenna, Increase the separation between the equipment and receiver, Connect equipment and receiver to outlets on different circuits, or Consult the dealer or an experienced radio/TV technician for help.

FCC-Approved Antennas (2.4 GHz)

XBee/XBee-PRO RF Modules can be installed using antennas and cables constructed with standard connectors (Type-N, SMA, TNC, etc.) if the installation is performed professionally and according to FCC guidelines. For installations not performed by a professional, non-standard connectors (RPSMA, RPTNC, etc) must be used.

The modules are FCC-approved for fixed base station and mobile applications on channels 0x0B - 0x1A (XBee) and 0x0C - 0x17 (XBee-PRO). If the antenna is mounted at least 20cm (8 in.) from nearby persons, the application is considered a mobile application. Antennas not listed in the table must be tested to comply with FCC Section 15.203 (Unique Antenna Connectors) and Section 15.247 (Emissions).

XBee RF Modules (1 mW): XBee Modules have been tested and approved for use with all of the antennas listed in the tables below (Cable-loss IS NOT required).

XBee-PRO RF Modules (60 mW): XBee-PRO Modules have been tested and approved for use with the antennas listed in the tables below (Cable-loss IS required when using antennas listed in the second table.

The antennas in the tables below have been approved for use with this module. Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

Antennas approved for use with the XBee®/XBee-PRO® RF Modules (Cable-loss is not required.)

Part Number	Type (Description)	Gain	Application*	Min. Separation
A24-HASM-450	Dipole (Half-wave articulated RPSMA - 4.5")	2.1 dBi	Fixed/Mobile	20 cm
A24-HABSM	Dipole (Articulated RPSMA)	2.1 dBi	Fixed	20 cm
A24-HABUF-P5I	Dipole (Half-wave articulated bulkhead mount U.FL. w/ 5" pigtail)	2.1 dBi	Fixed	20 cm
A24-HASM-525	Dipole (Half-wave articulated RPSMA - 5.25")	2.1 dBi	Fixed/Mobile	20 cm
A24-QI	Monopole (Integrated whip)	1.5 dBi	Fixed	20 cm

Antennas approved for use with the XBee RF Modules (Cable-loss is required)

Part Number	Type (Description)	Gain	Application*	Min. Separation	Required Cable-loss
Omni-Directional Class Antennas					
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	1.7 dB
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	1.9 dB
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	2.9 dB
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	3.9 dB
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	4.9 dB
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	4.9 dB
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	5.4 dB
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	6.4 dB
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	6.4 dB
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	7.9 dB
Omni-Directional Class Antennas					
A24-C1	Surface Mount	-1.5 dBi	Fixed/Mobile	20 cm	-
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	0.2 dB
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	0.7 dB
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	2.7 dB
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	5.7 dB
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	
Panel Class Antennas					
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	1.5 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	1.5 dB
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	6 dB
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	7 dB
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	8 dB
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	9 dB

Antennas approved for use with the XBee®/XBee-PRO® RF Modules (Cable-loss is required)

Part Number	Type (Description)	Gain	Application*	Min. Separation	Required Cable-loss
A24-C1	Surface Mount	-1.5 dBi	Fixed/Mobile	20 cm	-
A24-Y4NF	Yagi (4-element)	6.0 dBi	Fixed	2 m	8.1 dB
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	10.9 dB
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	11.1 dB
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	12.1 dB
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	13.1 dB
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	14.1 dB
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	14.1 dB
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	14.6 dB
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	15.6 dB
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	15.6 dB
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	17.1 dB
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	4.2 dB
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	5.1 dB
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	7.1 dB
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	10.1 dB
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	11.6 dB
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	12.1 dB
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	14.1 dB
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	17.1 dB
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	9.3 dB
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	9.3 dB
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	8.6 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	8.6 dB
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	13.1 dB
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	14.1 dB
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	15.1 dB
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	16.1 dB
A24-P19NF	Flat Panel	19.0 dBi	Fixed	2 m	19.1 dB

*** If using the RF module in a portable application** (For example - If the module is used in a handheld device and the antenna is less than 20cm from the human body when the device is operation): The integrator is responsible for passing additional SAR (Specific Absorption Rate) testing based on FCC rules 2.1091 and FCC Guidelines for Human Exposure to Radio Frequency Electromagnetic Fields, OET Bulletin and Supplement C. The testing results will be submitted to the FCC for approval prior to selling the integrated unit. The required SAR testing measures emissions from the module and how they affect the person.

RF Exposure

WARNING: To satisfy FCC RF exposure requirements for mobile transmitting devices, a separation distance of 20 cm or more should be maintained between the antenna of this device and persons during device operation. To ensure compliance, operations at closer than this distance is not recommended. The antenna used for this transmitter must not be co-located in conjunction with any other antenna or transmitter.

The preceding statement must be included as a CAUTION statement in OEM product manuals in order to alert users of FCC RF Exposure compliance.

Europe (ETSI)

The XBee RF Modules have been certified for use in several European countries. For a complete list, refer to www.digi.com

If the XBee RF Modules are incorporated into a product, the manufacturer must ensure compliance of the final product to the European harmonized EMC and low-voltage/safety standards. A Declaration of Conformity must be issued for each of these standards and kept on file as described in Annex II of the R&TTE Directive.

Furthermore, the manufacturer must maintain a copy of the XBee user manual documentation and ensure the final product does not exceed the specified power ratings, antenna specifications, and/or installation requirements as specified in the user manual. If any of these specifications are exceeded in the final product, a submission must be made to a notified body for compliance testing to all required standards.

OEM Labeling Requirements

The 'CE' marking must be affixed to a visible location on the OEM product.

CE Labeling Requirements

The CE mark shall consist of the initials "CE" taking the following form:

- If the CE marking is reduced or enlarged, the proportions given in the above graduated drawing must be respected.
- The CE marking must have a height of at least 5mm except where this is not possible on account of the nature of the apparatus.
- The CE marking must be affixed visibly, legibly, and indelibly.

Restrictions

Power Output: When operating in Europe, XBee-PRO 802.15.4 modules must operate at or below a transmit power output level of 10dBm. Customers have two choices for transmitting at or below 10dBm:

a. Order the standard XBee-PRO module and change the PL command to 0 (10dBm)

b. Order the International variant of the XBee-PRO module, which has a maximum transmit output power of 10dBm (@ PL=4).

Additionally, European regulations stipulate an EIRP power maximum of 12.86 dBm (19 mW) for the XBee-PRO and 12.11 dBm for the XBee when integrating antennas.

France: Outdoor use limited to 10 mW EIRP within the band 2454-2483.5 MHz.

Norway: Norway prohibits operation near Ny-Alesund in Svalbard. More information can be found at the Norway Posts and Telecommunications site (www.npt.no).

Declarations of Conformity

Digi has issued Declarations of Conformity for the XBee RF Modules concerning emissions, EMC and safety. Files can be obtained by contacting Digi Support.

Important Note:

Digi does not list the entire set of standards that must be met for each country. Digi customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. For more information relating to European compliance of an OEM product incorporating the XBee RF Module, contact Digi, or refer to the following web sites:

CEPT ERC 70-03E - Technical Requirements, European restrictions and general requirements: Available at www.ero.dk/.

R&TTE Directive - Equipment requirements, placement on market: Available at www.ero.dk/.

Approved Antennas

When integrating high-gain antennas, European regulations stipulate EIRP power maximums. Use the following guidelines to determine which antennas to design into an application.

XBee-PRO RF Module

The following antenna types have been tested and approved for use with the XBee Module:

Antenna Type: Yagi
RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maximum of 14 dBm). Any Yagi type antenna with 14 dBi gain or less can be used with no cable-loss.

Antenna Type: Omni-directional
RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maxi-

mum of 14 dBm). Any Omni-directional type antenna with 14 dBi gain or less can be used with no cable-loss.

Antenna Type: Flat Panel

RF module was tested and approved with 19 dBi antenna gain with 4.8 dB cable-loss (EIRP Maximum of 14.2 dBm). Any Flat Panel type antenna with 14.2 dBi gain or less can be used with no cable-loss.

XBee-PRO RF Module (@ 10 dBm Transmit Power, PL parameter value must equal 0, or use International variant)

The following antennas have been tested and approved for use with the embedded XBee-PRO RF Module:

- Dipole (2.1 dBi, Omni-directional, Articulated RPSMA, Digi part number A24-HABSM)
- Chip Antenna (-1.5 dBi)
- Attached Monopole Whip (1.5 dBi)

The RF modem encasement was designed to accommodate the RPSMA antenna option.

Canada (IC)

Labeling Requirements

Labeling requirements for Industry Canada are similar to those of the FCC. A clearly visible label on the outside of the final product enclosure must display the following text:

Contains Model XBee Radio, IC: 4214A-XBEE
Contains Model XBee-PRO Radio, IC: 4214A-XBEEPRO

The integrator is responsible for its product to comply with IC ICES-003 & FCC Part 15, Sub. B - Unintentional Radiators. ICES-003 is the same as FCC Part 15 Sub. B and Industry Canada accepts FCC test report or CISPR 22 test report for compliance with ICES-003.

Japan

In order to gain approval for use in Japan, the XBee RF module or the International variant of the XBee-PRO RF module (which has 10 dBm transmit output power) must be used.

Labeling Requirements

A clearly visible label on the outside of the final product enclosure must display the following text:

ID: 005NYCA0378

Appendix B. Additional Information

1-Year Warranty

XBee®/XBee-PRO® RF Modules from Digi Intenational, Inc. (the "Product") are warranted against defects in materials and workmanship under normal use, for a period of 1-year from the date of purchase. In the event of a product failure due to materials or workmanship, Digi will repair or replace the defective product. For warranty service, return the defective product to Digi, shipping prepaid, for prompt repair or replacement.

The foregoing sets forth the full extent of Digi's warranties regarding the Product. Repair or replacement at Digi's option is the exclusive remedy. THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, AND DIGI SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL DIGI, ITS SUPPLIERS OR LICENSORS BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVINGS, OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. THEREFORE, THE FOREGOING EXCLUSIONS MAY NOT APPLY IN ALL CASES. This warranty provides specific legal rights. Other rights which vary from state to state may also apply.

Anexo 20: Ejemplo de Interfaz en Android para comunicación con Arduino

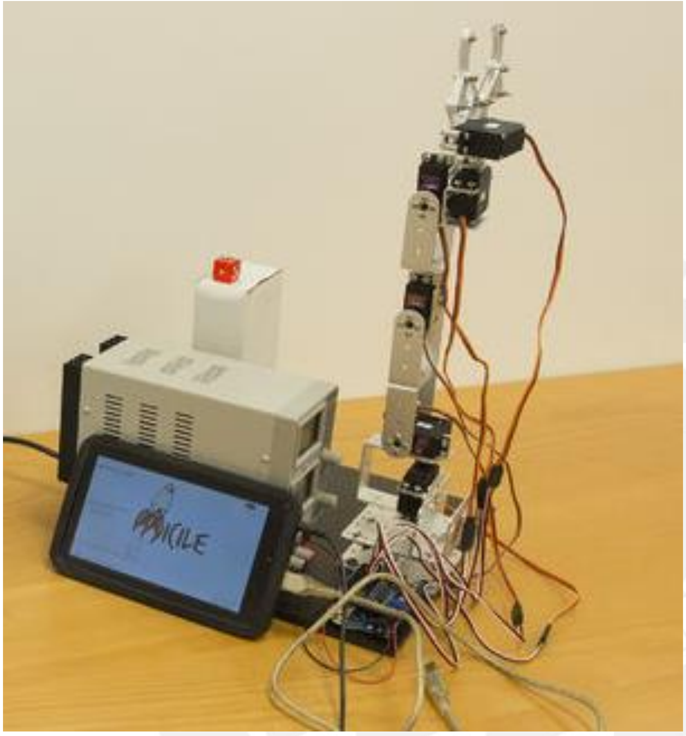


Figura 20-1: Brazo robótico controlado por Arduino Uno+ Tablet con SO Android (Fuente:www.instructables.com)

En este ejemplo se muestra la utilización de una Tablet con sistema operativo Android , para realizar una interfaz de comunicación con el Arduino Uno que se encarga de controlar el brazo robótico.

La Tablet y el Arduino se comunican por el puerto serial.

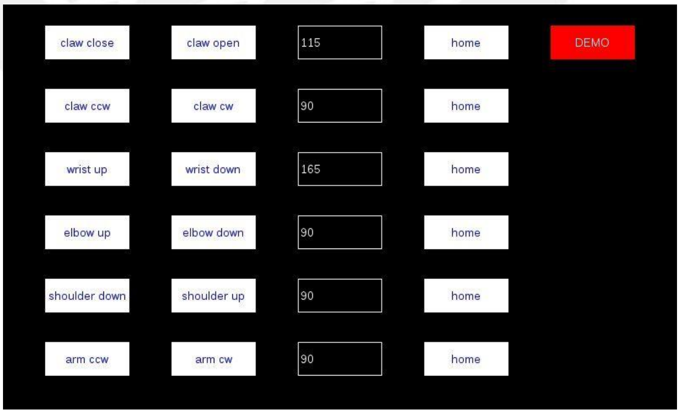


Figura 20-2: Interfaz mostrada en la Tablet (Fuente:www.instructables.com)

En la figura 20-2 se muestra la interfaz creada que se muestra en la Tablet, la cual cumple la función de guardar los movimientos del brazo robótico, los grados, que se deseen y luego al presionar DEMO el robot se moverá. Se puede lograr una interfaz mucho más amigable para obtener mejores resultados de interacción humano robot siguiendo la base de programas como este.

Anexo 21: Batería Yuasa 12V 7Ah y Cálculos del Consumo de Enegía

Arduino MEGA ADK	
roboclaw	40mA
74LS241	27mA
EasyVR	180mA
Matriz	275mA
Xbee-Pro	200uA
IMU	40mA

Tabla 21-1

Arduino Nano	
MB 1000	2mA*8
HCSR04	15mA*2

Tabla21-2

BeagleBone	
Levelshifter	4uA
cámara	40mA

Tabla21-3

En las tablas 21-1, 21-2 y 21-3 (Fuente: Propia) se muestran los gastos de corriente de todos los componentes que se alimentan a 5V.

Los componentes que consumen 12V están los motores:

Motores Pioneer: 2.5A

Sermotores Dynamixel: 1.5A

Estas cifras es el peor de los casos.

Entonces se tiene:

153.1 W en total

La batería proporciona 12 V 7 Ah (se conectarán dos baterías en paralelo)

E=12*7*3600*2= 604800 J (CV)

El tiempo de uso t=604800/(153.1*3600)

En el peor de los casos la batería duraría casi 2 horas

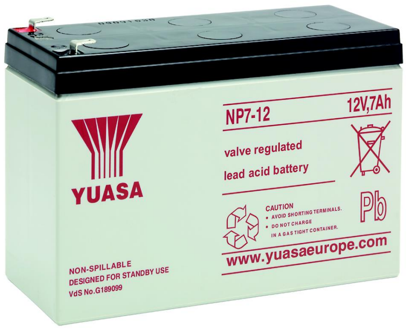
Pero los motores no trabajarán constantemente ni a toda su potencia por lo que esta duración se podría cuadruplicar.

NP-Series - Valve Regulated Lead Acid Battery

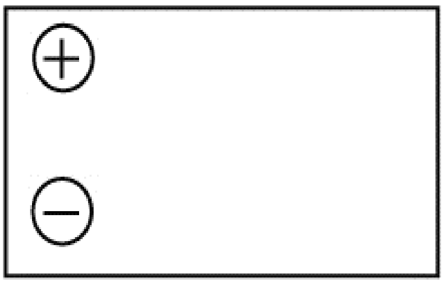
Data Sheet

NP7-12

SPECIFICATIONS		
Nominal voltage	12	V
20-hr rate Capacity to 1.75VPC at 20°C	7	Ah
10-hr rate Capacity to 1.75VPC at 20°C	6.4	Ah
DIMENSIONS		
Length	151 (±1)	mm
Width	65 (±1)	mm
Height		mm
(height over terminals)	97.5 (±2)	mm
Mass (typical)	2.2	kg
TERMINAL TYPE		
FASTON (Quickfit / release)	4.75	mm
OPERATING TEMPERATURE RANGE		
Storage	-20°C to +60°C	
Charge	-15°C to +50°C	
Discharge	-20°C to +60°C	
STORAGE		
Capacity loss per month at 20°C (approx)	3	%
CASE MATERIAL		
Standard Option	ABS (UL.94:HB)	
Flame retardant option (FR)	ABS (UL94:V0)	
CHARGE VOLTAGE		
Float charge voltage at 20°C	13.65 (±1%) 2.275 (±1%)	V V/cell
Float Charge voltage temperature correction factor (for variations from the standard 20°C)	-3	mV/cell/°C
Cyclic (or Boost) charge at 20°C	14.5 (±3%) 2.42 (±3%)	V V/cell
Cyclic Charge voltage temperature correction factor (for variations from the standard 20°C)	-4	mV/cell/°C
CHARGE CURRENT		
Float charge current limit	No limit	A
Cyclic (or Boost) charge current limit	1.75	A
MAXIMUM DISCHARGE CURRENT		
1 second	210	A
1 minute	48	A
SHORT-CIRCUIT CURRENT & INTERNAL RESISTANCE		
(according to EN IEC 60896-21)		
Internal resistance	N/A	mq
Short-Circuit current	N/A	A
IMPEDANCE		
Measured at 1 kHz	23	mq
PERFORMANCE & CHARACTERISTICS		
Refer to the technical manual	NP	
DESIGN LIFE		
EUROBAT Classification: Standard Commercial	3 to 5	years
Yuasa design life @ 20°C	up to 5	years
SAFETY		
Installation		
Can be installed and operated in any orientation except permanently inverted		
Handles		
Batteries must not be suspended by their handles (where fitted)		
Vent valves		
Each cell is fitted with a low pressure release valve to allow gasses to escape and then reseal.		
Gas Release		
VRLA Batteries release hydrogen gas which can form explosive mixtures in air. Do not place inside a sealed container		
Recycling		
YUASA's VRLA batteries must be recycled at the end of life in accordance with local and national laws and regulations		



LAYOUT



3RD PARTY CERTIFICATIONS

ISO 9001 - Quality Management Systems
ISO 14001 - Environmental Management Systems
EN 18001 - OHSAS Management Systems
UNDERWRITERS LABORATORIES Inc.
VdS (Germany) - VdS No: G189099

STANDARDS

IEC61056

ALL DATA IS SUBJECT TO CHANGE WITHOUT NOTICE

Issue No.: V.4 / Issue Date: March 2014

YUASA BATTERY SALES UK LTD.
Unit 13, Hunts Rise
South Marston Industrial Estate
Swindon
SN3 4TG
UK

Anexo 22: Gestos

El robot contará con dos matrices de leds como se mencionó y estos se utilizarán para reproducir gestos. Estos gestos están basados en el Superintendente del videojuego Halo (figura 22-1), que con sólo dos ojos muestra diferentes estados de ánimo. Esto ayuda a simplificar la programación y la utilización de más recursos para mostrar emociones, como una boca, nariz, cejas, etc.

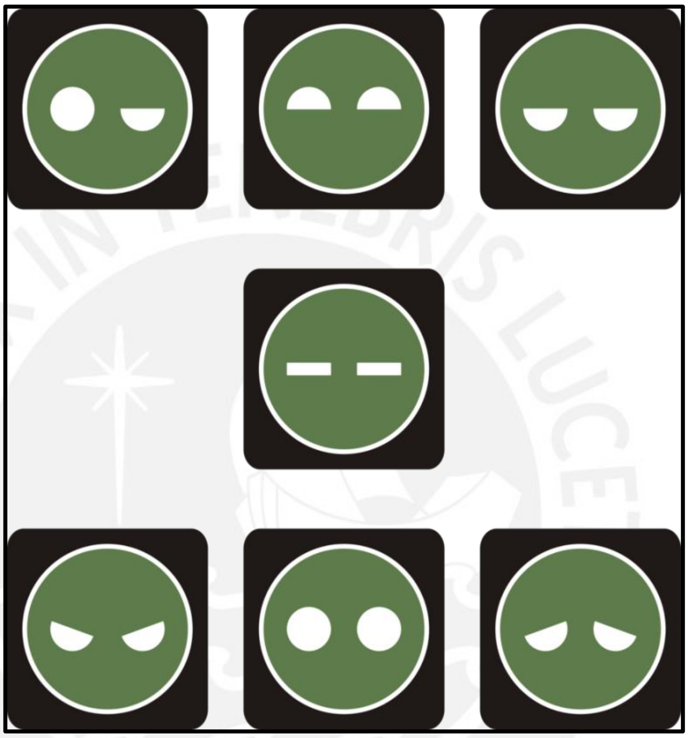


Figura 22-1:Estados de ánimo del Superintendente de Halo(Fuente:halo.wikia.com)

Para los movimientos del robot se puede observar en las figuras 22-2, 22-3 y 22-4 los diferentes gestos que se realizarán con los brazos para luego poder preguntar a los usuarios cómo han percibido al sistema mecatrónico.

Figura 22-2 (Fuente:Propia)

Figura 22-3 (Fuente:Propia)

Figura 22-4 (Fuente:Propia)

Anexo 23: Programa para Detección de Personas

El siguiente ejemplo se realizó en Open CV y es la manera en la que se detectará a las personas.

El código primero quitará el fondo dejando la imagen, luego se halla el contorno de la imagen, con lo que podrá hallar el área de la imagen y si esta es suficientemente grande es que hay una persona al frente y se interactuará con ella.

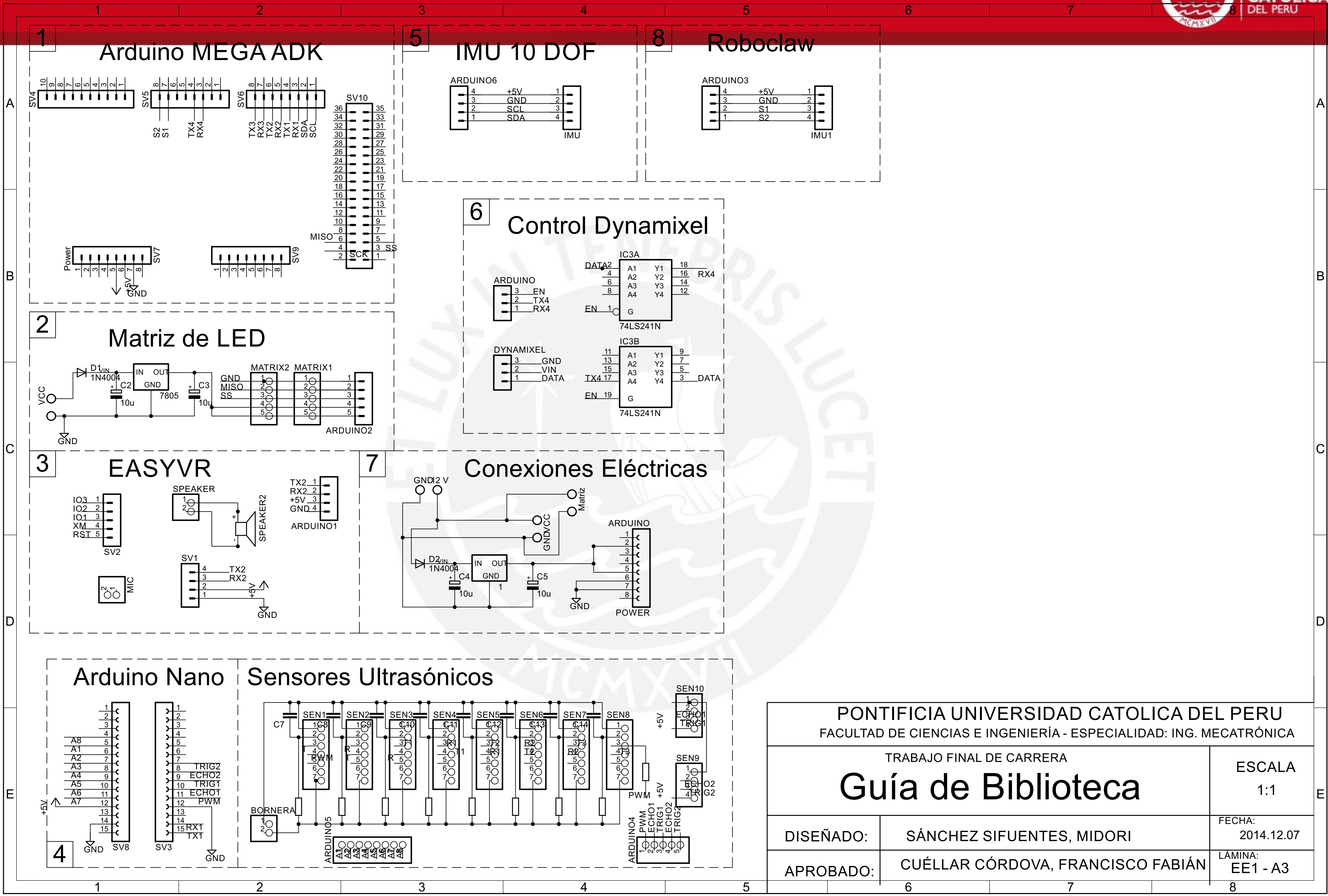
```
#include "opencv2\imgproc\imgproc.hpp"
#include "opencv2\nonfree\nonfree.hpp"
#include <opencv2\video\background_segm.hpp>
#include "opencv2\highgui\highgui.hpp"
#include "opencv2\features2d\features2d.hpp"
#include "opencv2\core\core.hpp"
#include <opencv2/core/core.hpp>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <cv.h>
#include <highgui.h>
#include <iostream>
#include <sstream>

using namespace std;
using namespace cv;

Mat frame;
Mat src_gray;
RNG rng(12345);
Mat frames;
Mat imgrect;
Mat fgMaskMOG;
Ptr<BackgroundSubtractor> pMOG; //MOG Background subtractor
int keyboard;
int areaM;
int contourM;
int cim=1;
int thresh=50;
int max_thresh=255;
Mat img_matches;
void Sift_images(Mat);
int minHessian = 400;
int cc;
```

```
int main(int arc,char *argv[1])
{
	VideoCapture cap(0);
	if(!cap.isOpened())
	{
		cout<<"Cannot initialize video"<<endl;
		return -1;
	}
	vector<vector<Point> > contours;
	vector<Vec4i> hierarchy;

	namedWindow("MyVideo",CV_WINDOW_AUTOSIZE);
	moveWindow("MyVideo",0,0);
	namedWindow("Contours",CV_WINDOW_AUTOSIZE);
	namedWindow("FG Mask MOG",CV_WINDOW_AUTOSIZE);
	namedWindow("Rect",1);
	namedWindow("Matches",1);
	moveWindow("Matches",1000,0);
	pMOG= new BackgroundSubtractorMOG();

	while(1)
	{ cap>>frame;
	imshow("MyVideo",frame);

	pMOG->operator()(frame, fgMaskMOG);

  imshow("FG Mask MOG", fgMaskMOG);
  keyboard = waitKey( 30 );
  findContours(fgMaskMOG, contours, hierarchy, CV_RETR_TREE, CV_CHAIN_APPROX_SIMPLE,
Point(0, 0) );// Find contours
	Mat drawing = Mat::zeros( fgMaskMOG.size(), CV_8UC3 );// Draw contours
	for( int i = 0; i< contours.size(); i++ ) {
    Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
    drawContours( drawing, contours, i, color, 2, 8, hierarchy, 0, Point() );
   }
	 imshow( "Contours", drawing );// Show in a window

	 //calculate the area of the contour, to find the largest area and detect the motion
	for( int i = 0; i< contours.size(); i++ ) // iterate through each contour
		{	double a=contourArea( contours[i],false);  //  Find the area of contour
			if(a>areaM){
				areaM=a;
				contourM=i;//Store the index of largest contour
				waitKey(2000);//wait 2 seconds to take only the object
				Rect rect = boundingRect(contours[i]); // Find the bounding
rectangle for biggest contour
				imgrect=frame(rect);
				imshow("Rect",imgrect);//show the object in another window
			}
```

```
			}

			if (waitKey(30)==27){
			cout<<"esc key is pressed by user"<< endl;
			break;
			}
		}
return 0;
}
```

(Fuente: Propia)

Anexo 24 : Esquemático General



PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU

FACULTAD DE CIENCIAS E INGENIERÍA - ESPECIALIDAD: ING. MECATRÓNICA

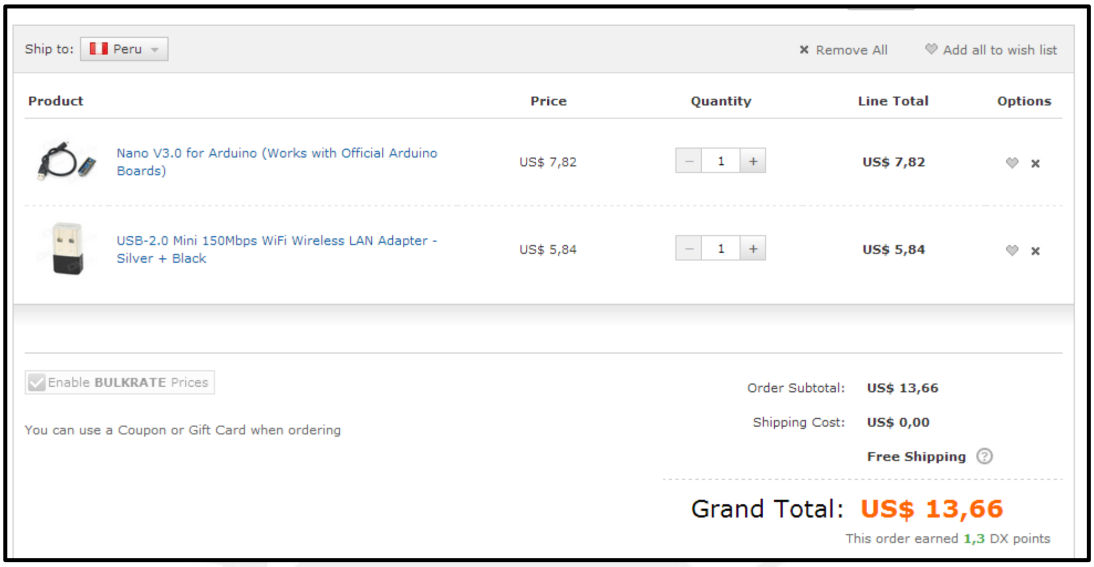
TRABAJO FINAL DE CARRERA

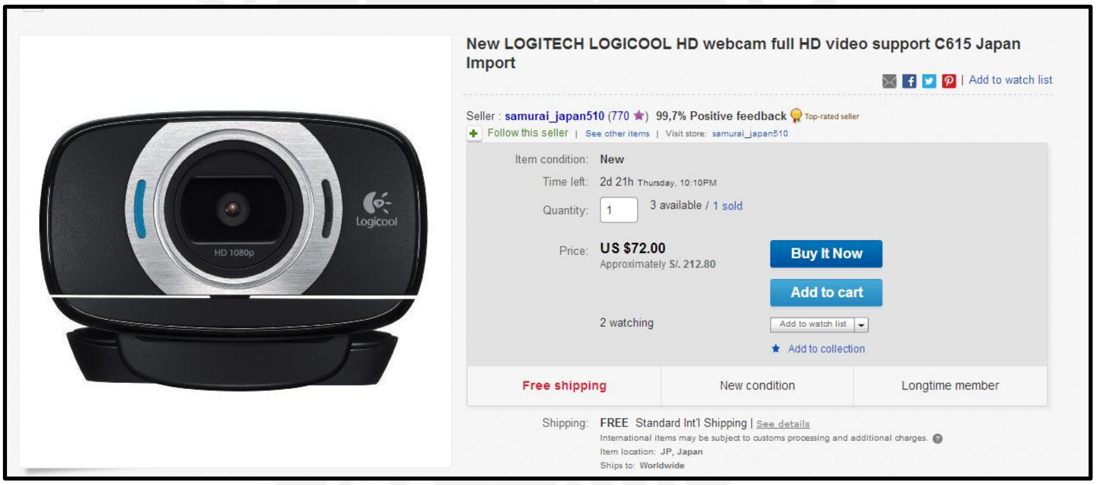
Guía de Biblioteca

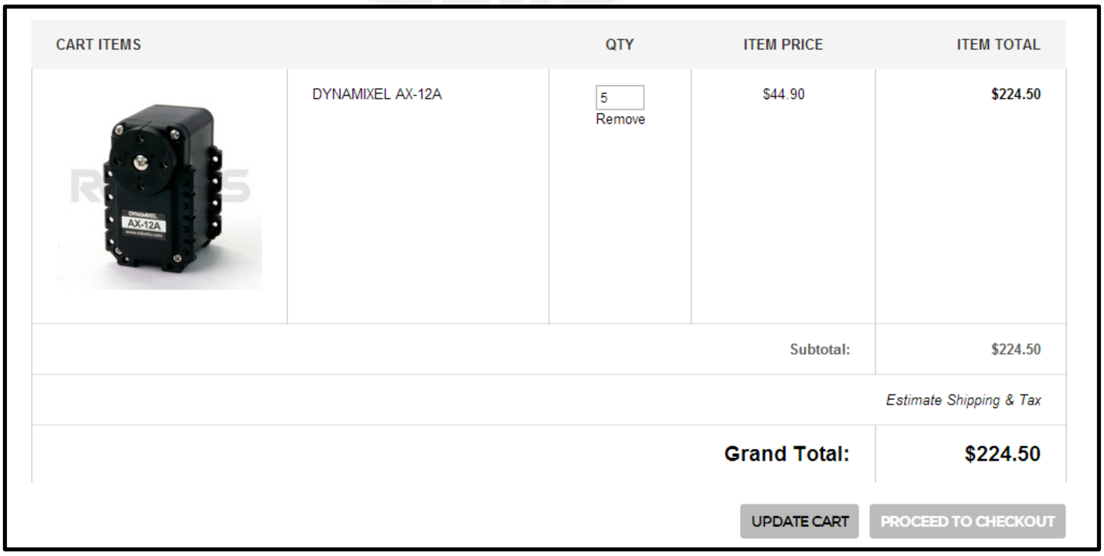
ESCALA	1:1		
DISEÑADO:	SÁNCHEZ SIFUENTES, MIDORI	FECHA:	2014.12.07
APROBADO:	CUÉLLAR CÓRDOVA, FRANCISCO FABIÁN	LÁMINA:	EE1 - A3

08/12/2014 09:35:15 a.m. f=0.70 D:\2014-2\TFC\final\Esquematico general.sch (Sheet: 1/1)

Anexo 25: Proformas







FADIPESA
SOLUCIONES EN IMPRESIÓN 3D

Lima, 08 de diciembre del 2014

Señores:

Pontificia Universidad Católica del Perú
Av. Universitaria 1801
San Miguel - Lima

Cotización: **Carcasa**

Fabricación de modelo de madera para faldón en resina.

Valor de venta: US$ 6,500.00

Fabricación de modelo de madera para torso en resina.

Valor de venta: US$ 1,500.00

Fabricación de modelo de madera para hombro en resina.

Valor de venta: US$ 500.00

Fabricación de modelo de madera para cara en resina.

Valor de venta: US$ 900.00

Fabricación de modelo de madera para collar en resina.

Valor de venta: US$ 1,100.00

Fabricación de molde de madera en dos partes para moño cabeza.

Valor de venta: US$ 1,200.00

Fabricación de molde de madera en dos partes para brazo.

Valor de venta: US$ 2,600.00

Total: US$ 14,300.00
IGV (18%): US$ 2,574.00
Precio: US$ 16,874.00

Condiciones:

- MONEDA: US Dólares
- TIEMPO DE ENTREGA: 45 días
- PAGO: Contra-entrega

Banco de crédito del Peru BCP
Titular de la cuenta: **FABRICACIONES DIGITALES DEL PERU S.A.**
Cuenta Corriente en Nuevos Soles: **194-2139523-0-42**

NOTA: Cotización valida por 15 días

Contacto:
Antonio Moll León
amoll@fadipesa.com
Telf: +51 1 654-3013
Cel: 9 9620-1441
RPM: #930354
www.fadipsa.com
Fabricaciones Digitales del Perú S.A.

FADIPESA
SOLUCIONES EN IMPRESIÓN 3D

Lima, 08 de diciembre del 2014

Señores:

Pontificia Universidad Católica del Perú
Av. Universitaria 1801
San Miguel - Lima

Cotización: **Estructura central**

Fabricación de piezas según planos en aluminio laminado para ensamblaje de estructura central, incluye rolado, corte laser, plegado, mecanizado y soldado.

NOTA: Los cordones son realizados por medio de soldadura TIG

Valor de venta: US$ 1,990.00
IGV (18%): US$ 358.20
Precio: US$ 2,348.20

Condiciones:

- MONEDA: US Dólares
- TIEMPO DE ENTREGA: 30 días
- PAGO: Contra-entrega

Banco de crédito del Peru BCP
Titular de la cuenta: **FABRICACIONES DIGITALES DEL PERU S.A.**
Cuenta Corriente en Nuevos Soles: **194-2139523-0-42**

NOTA: Cotización valida por 15 días

Contacto:
Antonio Moll León
amoll@fadipesa.com
Telf: +51 1 654-3013
Cel: 9 9620-1441
RPM: #930354
www.fadipsa.com
Fabricaciones Digitales del Perú S.A.

FADIPESA
SOLUCIONES EN IMPRESIÓN 3D

Lima, 25 de noviembre del 2014

Señores:

Pontificia Universidad Católica del Perú
Av. Universitaria 1801
San Miguel - Lima

Cotización: **Componentes electrónicos**

Cantidad	Descripción	Precio unitario*	Precio total*
2	Matriz LED	**US$ 115.00**	**US$ 230.00**
8	Ultrasonido tipo MB1000	**US$ 39.00**	**US$ 312.00**
1	Placa Beagleboard	**US$ 190.00**	**US$ 190.00**
1	Controlador de motor 2x15A	**US$ 159.00**	**US$ 159.00**
1	Conversor logico bidireccion de 8 canales	**US$ 25.00**	**US$ 25.00**
1	Adaptador AV500 con enchufe incorporado	**US$ 75.00**	**US$ 75.00**

* Precios NO incluyen IGV

Valor de venta: US$ 991.00
IGV (18%): US$ 178.38
Precio: US$ 1,169.38

Condiciones:

- MONEDA: US Dólares
- TIEMPO DE ENTREGA: 30 días
- PAGO: Contra-entrega

Banco de crédito del Peru BCP
Titular de la cuenta: **FABRICACIONES DIGITALES DEL PERU S.A.**
Cuenta Corriente en Nuevos Soles: **194-2139523-0-42**

NOTA: Cotización valida por 15 días

Contacto:
Antonio Moll León
amoll@fadipesa.com
Telf: +51 1 654-3013
Cel: 9 9620-1441
RPM: #930354
www.fadipsa.com
Fabricaciones Digitales del Perú S.A.

Lima 01 de Diciembre de 2014

SOLICITUD DE COTIZACIÓN

Estimado

Ana Cristina Midori

Ref. Impresión 3d de prototipo en plástico ABS
Lima.-

NOMBRE / EMPRESA	PIEZA	TOTAL	MATERIAL / IMPRESORA
Ana Cristina Midori	acople hombro x5	S/. 420	FORTUS 400/ABS
	agarra camara		
	acople motor cuello		
	agarra matriz x4		
TOTAL		**S/. 420**	

Forma de pago
- Adelantado
- Tesorería de la Pontificia Universidad Católica del Perú

Tiempo de entrega referencial

Atentamente,

Jennifer Wong Poggi
Responsable del área de Impresión 3D
Pontificia Universidad Católica del Perú