

## CODIGO FUENTE DEL PROGRAMA

```
/*
 * Registro.c
 *
 * Created: 21/08/2013 08:56:03 p.m.
 * Author: JORGE
 */

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

int programa_adc(unsigned char);
void descomponer(unsigned int);
void conversion      (unsigned int,unsigned char);
unsigned char detectar_cambio(unsigned char);
void reseteo(void);
void motor_up(void);
void motor_down(void);

int main(void)
{
    unsigned int z;
    unsigned char temp;
    unsigned char e_ant=0b11100000;
    unsigned char e_act;

    DDRA=0x00;
    DDRB=0xFF;
    DDRC=0xFF;
    DDRD=0xF3;

    MCUCR=(1<<ISC01) | (1<<ISC00) | (1<<ISC11) | (1<<ISC10);
    GICR=(1<<INT0) | (1<<INT1);
    sei();

    while(1)
    {
        temp=PINA;
        temp=PINA&0b11100000;

        e_act=detectar_cambio(e_ant);
        e_ant=e_act;

        z = programa_adc(temp);
        conversion(z,temp);
    }
    return 0;
}

int programa_adc(unsigned char interruptor)
{
    unsigned char VH;
    unsigned char VL;
    unsigned int registro;

    ADCSRA=(1<<ADEN) | (1<<ADPS2) | (1<<ADPS1);
    ADMUX=(1<<REFS0);
```

```

    if (interruptor==0b10000000)
    {

    }

    else if((interruptor==0b01000000) || (interruptor==0b00100000))
    {
        ADMUX|=(1<<MUX0);
    }

    else
    {
        return 0;
    }
    _delay_ms(50);

    while(1)
    {
        ADCSRA|=(1<<ADSC);
        while(!(ADCSRA&(1<<ADIF)));
        ADCSRA|=(1<<ADIF);
        VL=ADCL;
        VH=ADCH;

        registro=VH;                //Guardamos el valor del ADC en un registro
        registro=registro<<8;
        registro+=VL;

        return(registro);
    }
}

void descomponer(unsigned int numero)
{

    unsigned int temp;
    unsigned int uni;
    unsigned int dec;
    unsigned int cen;

    uni=numero%10;
    temp=numero-uni;
    temp=temp%100;
    dec=temp/10;
    temp=numero-dec*10-uni;
    cen=temp/100;

    temp=0;
    temp=cen<<4;
    temp+=dec;

    PORTB=uni;
    PORTC=temp;

}

void conversion(unsigned int val,unsigned char cual)
{
    float valor;
    unsigned int num;
    if (cual==0b10000000)

```

```

{
    valor= (float)val;
    valor=valor*2000/1023;           // 0-5VDC --- 0-1024bits --- 0-2000VAC
    valor=round(valor);
    num=(unsigned int)valor;

    if (num<1000)
    {
        descomponer(num);
    }
    else
    {
        PORTC=0xCC;
        PORTB=0X0C;
    }

}

else if((cual==0b01000000)|| (cual==0b00100000))
{
    valor= (float)val;
    valor=valor*400/1023;           // 0-5VDC --- 0-1024bits --- 0-400AAC
    valor=round(valor);
    num=(unsigned int)valor;
    descomponer(num);
}
else
{
    PORTB=0x0F;
    PORTC=0xFF;
    return;
}

}

unsigned char detectar_cambio(unsigned char e_ant)
{
    unsigned char e_act;

    e_act=PINA&0b11100000;

    if (e_act!=e_ant)
    {
        reseteo();
    }
    return(e_act);
}

void reseteo()
{
    unsigned char temp;
    unsigned int y;
    do
    {
        temp=PINA;
        temp=PINA&0b11100000;
        y = programa_adc(temp);
        PORTC=0xCC;
        PORTB=0X0C;
        PORTD|= 0b01000000;
    } while (y!=0);

    PORTD=0x00;
}

```

```

ISR(INT0_vect)
{
    motor_up();
    GIFR=(1<<INTF0);
}

ISR(INT1_vect)
{
    motor_down();
    GIFR=(1<<INTF1);
}

void motor_up(void)
{
    unsigned char temp;
    unsigned int valor;
    temp=PINA;
    temp=PINA&0b11100000;
    valor = programa_adc(temp);

    if (temp==0b10000000)
    {

        if (valor>=512)
        {
            return;
        }
        else
        {
            PORTD|= 0b10000000;
            _delay_ms(50);
            PORTD&= 0b01111111;
        }
    }
    else
    {

        if (valor==1023)
        {
            return;
        }
        else
        {
            PORTD|= 0b10000000;
            _delay_ms(50);
            PORTD&= 0b01111111;
        }
    }
}

void motor_down(void)
{
    unsigned char temp;
    unsigned int valor;
    temp=PINA;
    temp=PINA&0b11100000;
    valor = programa_adc(temp);
    if (valor==0)
    {
        return;
    }
    else

```

```
{  
    PORTD|= 0b01000000;  
    _delay_ms(50);  
    PORTD&= 0b10111111;  
}  
  
}
```