

# INTRODUCCIÓN

El siguiente circuito consiste en programar el PIC para poder recibir una señal analógica y en base al valor recibido, enviar un PWM y dos

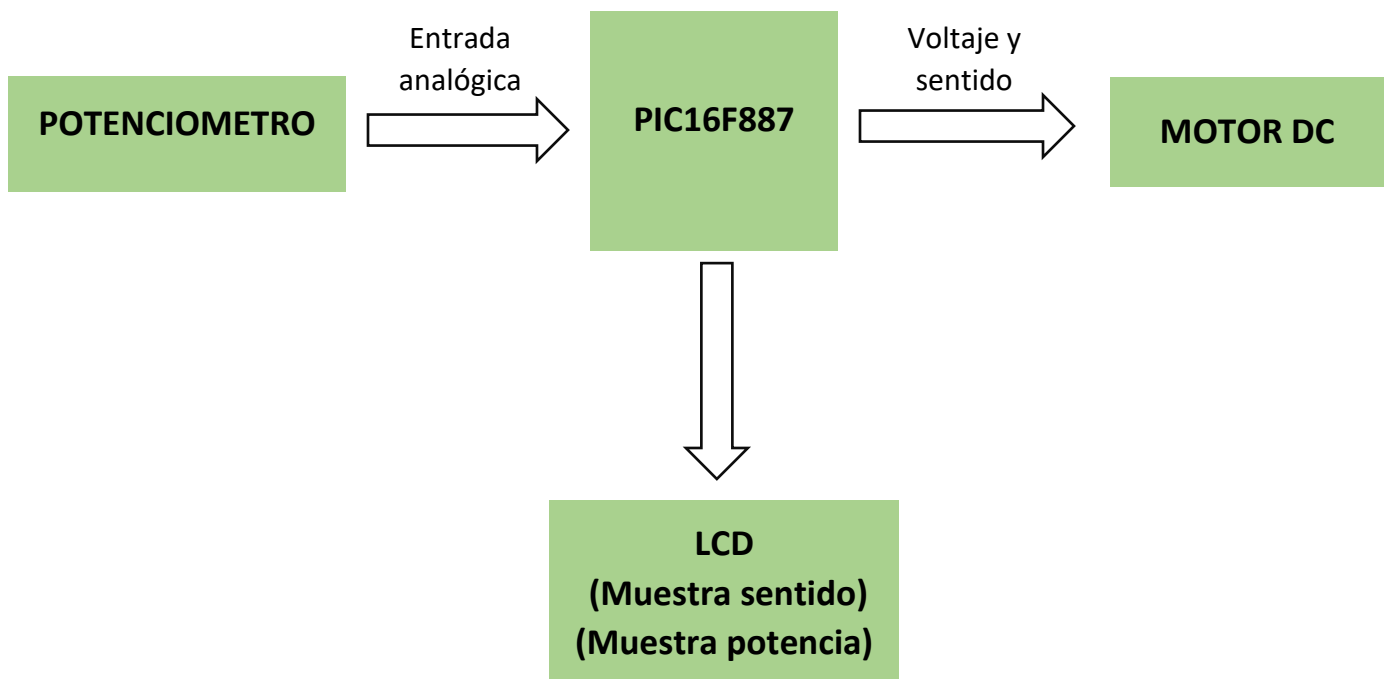
bits que encenderán un motor con sentido y potencia según sea dada por estas señales, la señal analógica será provista por un potenciómetro. El motor girará a la derecha o a la izquierda y se mostrará esto en la pantalla LCD con un mensaje y con la potencia a la que está funcionando el motor.

**Nombre: Barrios García Aldo**

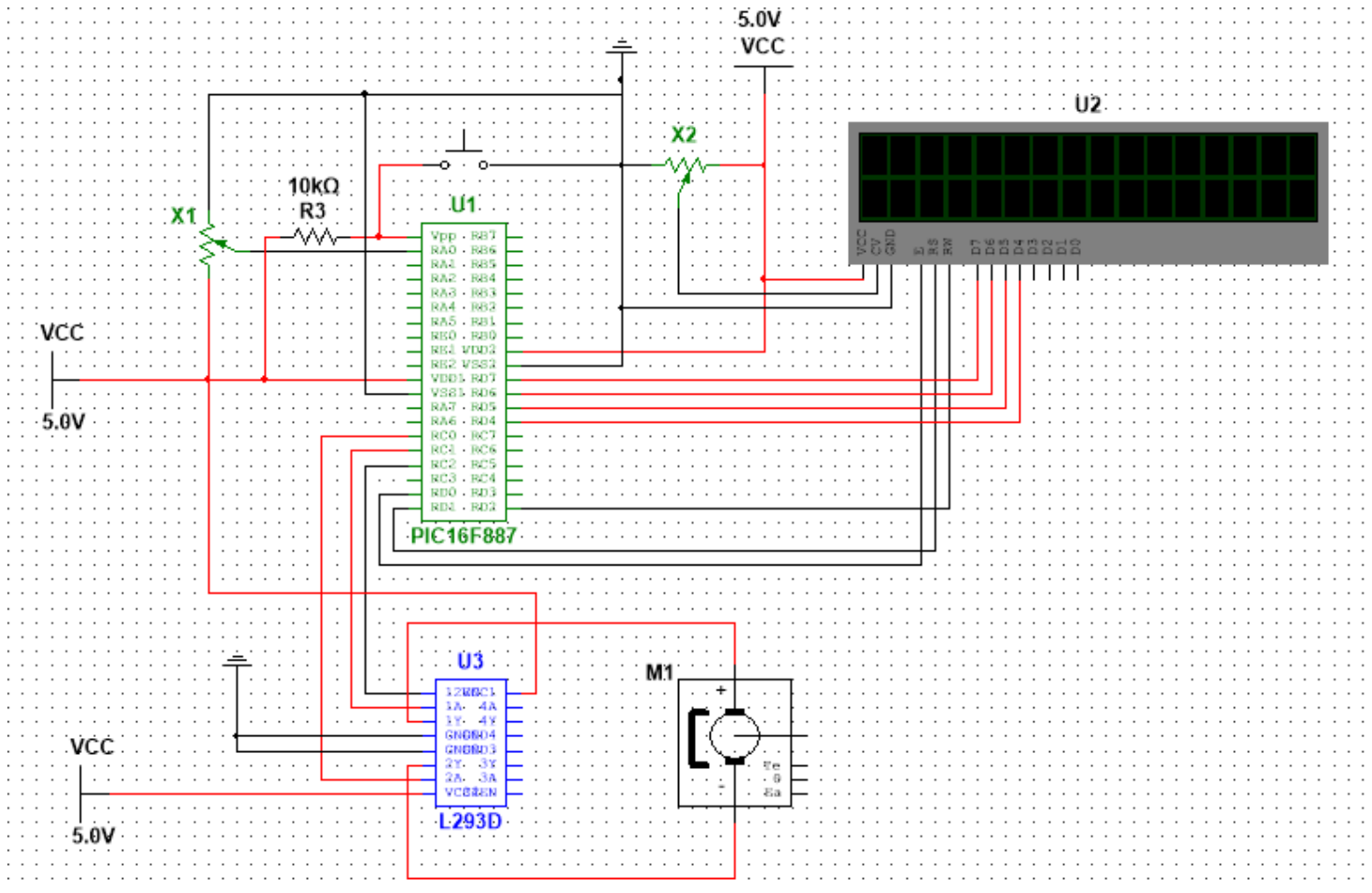
**Grupo: 1758**

**Circuito: 06**

## DIAGRAMA DE BLOQUES



## DIAGRAMA ELÉCTRICO



**CÓDIGO**

```
#include<16f887.h>
#fuses INTRC_IO,NOBROWNOUT,PROTECT,NOLVP,MCLR,NOBROWNOUT
#device ADC=8
#use delay (INTERNAL=4000000)
#include<lcd.c>
#include <math.h>

void main()
{
    signed int16 transf, valor;

    lcd_init();
    setup_ccp1(CCP_PWM);    //Se configura ccp1 para usar pwm

    setup_timer_2(T2_DIV_BY_16,255,1); // Timer a usar

    set_pwm1_duty(0);    //Fijado de ciclo de trabajo en 0%

    setup_adc_ports(sAN0); //Puerto analogico para muestreo
    setup_adc(ADC_CLOCK_INTERNAL); // Reloj interno
```

```

while(TRUE)
{
    set_adc_channel(0);
    valor=read_adc();

    //ESTADO APAGADO

    if(valor<129 && valor>126){ //Rango muersto es de 127 a 128
        output_low(PIN_C0); //PIN C0 en estado bajo
        output_low(PIN_C1); //PIN C1 en estado bajo

        printf(lcd_putc, "\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc, "Motor Apagado");
        lcd_gotoxy(1,2);
        printf(lcd_putc, "Velocidad: 0%%");
        delay_ms(1000);
        set_pwm1_duty(0);
    }

    //GIRO A LA IZQUIERDA

    else if(valor<127){ //Rango menor a 127
        output_high(PIN_C0); //PIN C0 en estado alto
        output_low(PIN_C1); //PIN C1 en estado bajo

        transf=fabs((valor*2)-255); //Conversión para poder sacar PWM
        set_pwm1_duty((transf)*4); //Aqui transf tiene valor de 0 a 255 y se multiplica para alcanzar los 5V
        transf = transf*100/255; //Conversión para poder mostrar porcentaje

        printf(lcd_putc, "\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc, "Motor Izquierda");
        lcd_gotoxy(1,2);
        printf(lcd_putc, "Velocidad: %lu %%", transf);
        delay_ms(1000);
    }

    //GIRO A LA DERECHA

    else if(valor>128){ //Rango mayor a 128
        output_high(PIN_C1); //PIN C1 en estado alto
        output_low(PIN_C0); //PIN C0 en estado bajo

        transf=fabs((valor*2)-255); //Conversión para poder sacar PWM
        set_pwm1_duty((transf)*4); //Aqui transf tiene valor de 0 a 255
        transf = transf*100/255; //Conversión para poder mostrar porcentaje

        printf(lcd_putc, "\f");
        lcd_gotoxy(1,1);
        printf(lcd_putc, "Motor Derecha");
        lcd_gotoxy(1,2);
        printf(lcd_putc, "Velocidad: %lu %%", transf);

        delay_ms(1000);
    }

}
}

```

## IMÁGENES DEL CIRCUITO

