

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Microcomputadoras

Proyecto Final

Profesor: Rubén Anaya García

Alumnos:

- Kennedy Villa Carolina
- Murrieta Villegas Alfonso
- Reza Chavarría, Sergio Gabriel
- Valdespino Mendieta Joaquín

Grupo: 01

Semestre: 2021-2

Práctica Final

Objetivo

Realizar un programa que reciba comandos a través de puerto serie y controle el funcionamiento de 2 motores de CD, que están distribuidos en un robot móvil de tipo diferencial.

Características Generales

Programa

- a) Recepción asíncrona (interrupciones)
- b) Control de velocidad PWM
- c) En un display LCD I2C mostrar el comando que realiza el móvil.
- d) El programa funcionará durante 2 minutos únicamente; enviará un mensaje al LCD indicando el mensaje adecuado.

Comando	Acciones
P	ROBOT EN PARO
A	ADELANTE
T	ATRÁS
D	DERECHA
I	IZQUIERDA

Funcionamiento

- a) La recepción de datos la realiza a través de Bluetooth (HC05; instalar bibliotecas para Proteus).
- b) La transmisión de datos deberá ser por medio de una aplicación que ejecutará desde su teléfono celular (usar el software preferido para diseñar su aplicación, recomendación app inventor de MIT).
- c) Usar los elementos necesarios en Proteus para ver el funcionamiento del robot.

Desarrollo

Para la realización del proyecto se utilizó como Microcontrolador principal el PIC16F877A y el programa se realizó en el Lenguaje C. En el desarrollo del programa se utilizaron los elementos siguientes.

Elementos internos utilizados:

- Manejo del TIMER1 para la temporización de 2 minutos del programa.
- Implementación del convertidor A/D y el manejo de CCP1 para obtener el PWM, esto para modificar la velocidad.

Elementos Adicionales al manejo del programa:

- Manejo de la biblioteca I2C_LCD.c para la implementación de display LCD por el manejo por medio de Inter Circuito Integrado.
- Implementación de la biblioteca para HC05 Bluetooth.
- Implementación de turtle para la simulación del control del robot.
- Aplicación realizada en App Invertor del MIT.

I2C

Para la implementación en el código se obtuvo el archivo `i2c_LCD.c`, el cual tiene el manejo del display LCD a partir del uso del Inter Circuito Integrado. El archivo se agregó a la carpeta de “Drivers” de la carpeta principal del CCS Compiler (Carpeta PICC).

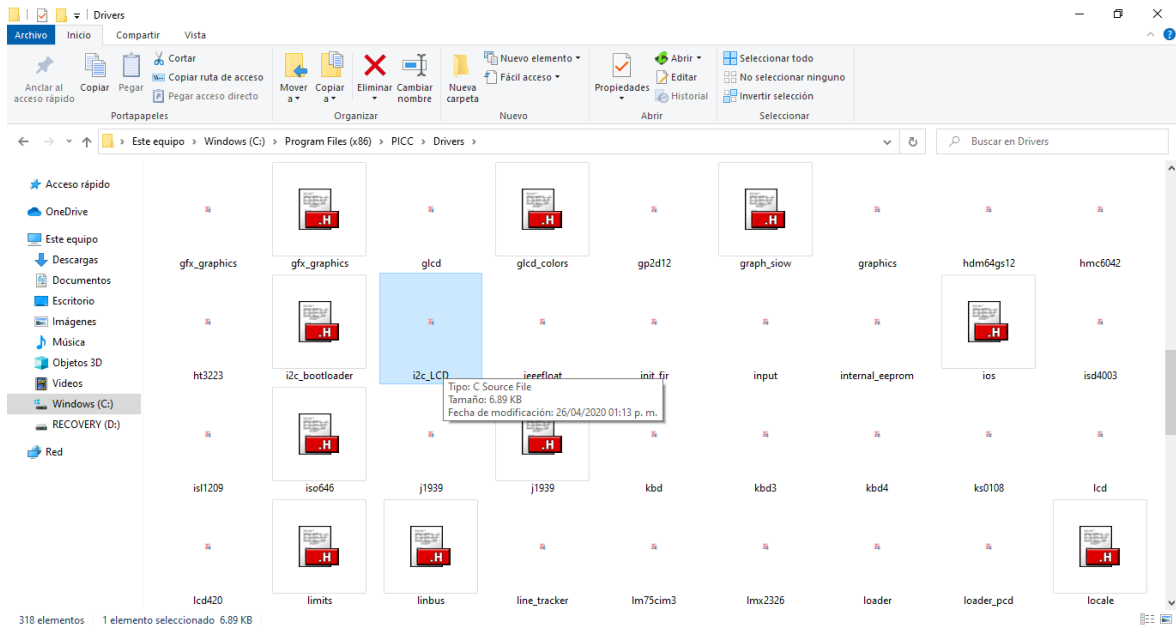


Ilustración 1: i2c_LCD.c en DRIVERS de CCS Compiler

El manejo de la biblioteca se utilizó de manera similar a la implementación utilizada en clases del display LCD. Lo que se diferencia es en la inicialización del LCD, se debe especificar el tamaño en renglones y columnas.

Nota: Este archivo está anexo en el comprimido del proyecto.

Bluetooth HC-05

Para el manejo de la conexión vía bluetooth se agregaron los archivos “BluetoothTEP.idx” y “BluetoothTEP.lib” en la dirección siguiente “C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\LIBRARY”, correspondiente a los drivers de Proteus Profesional.

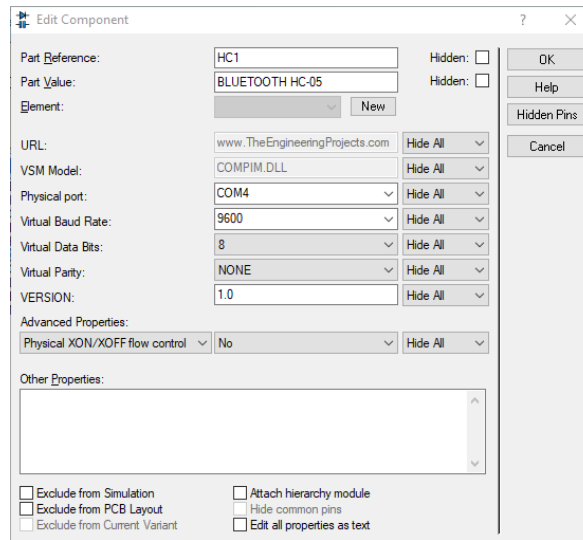


Ilustración 4: Configuración del HC-05

Nota: Verificar que el puerto COM a utilizar este disponible. Esto puede verse en la configuración de Bluetooth.

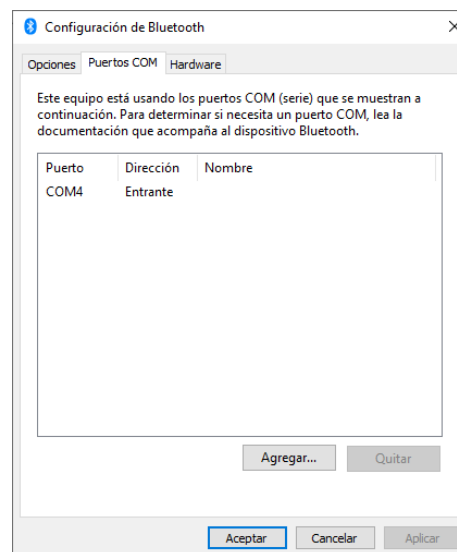


Ilustración 5: Visualización de puertos COM disponibles

Turtle

Para la visualización del manejo del robot se utilizó el elemento Turtle. Este elemento se obtuvo del ejemplo de Proteus del seguidor de línea. Este contiene sensores de color en la parte interna, sensores de distancia en la parte delantera y 2 motores de Corriente directa. Para el proyecto se utilizó solamente los motores CD.

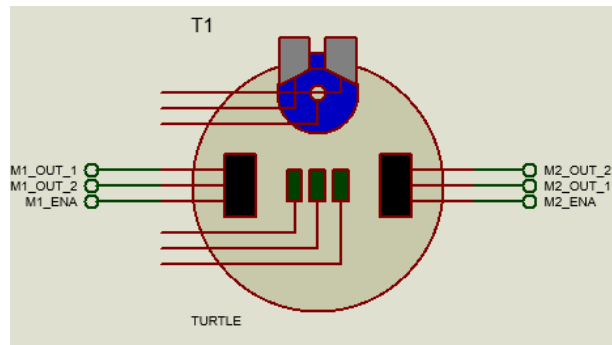


Ilustración 6: Robot turtle

El manejo del turtle se utilizó específicamente para ver la simulación del movimiento del elemento implementada en Proteus.

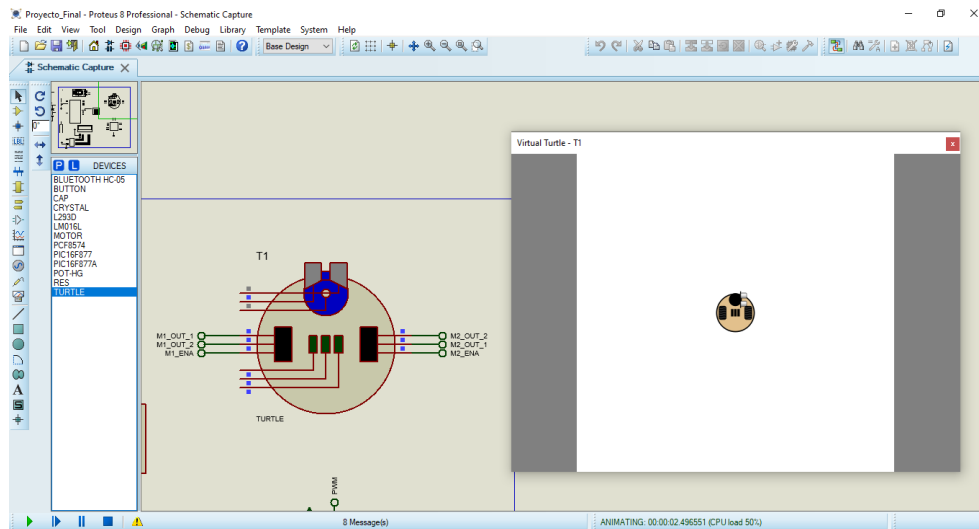


Ilustración 7: Visualización de movimiento

Para modular la velocidad de los motores se utilizó el L293D. La entrada VS es utilizada para la tensión de los motores, esto permite tener control de la velocidad de rotación de los motores. Esto se manejará a partir de un potenciómetro y del convertidor A/D del PIC16F877A. La entrada de VS será manejada por la salida de CCP1, el cual manejará la modulación de ancho de pulso (PWM).

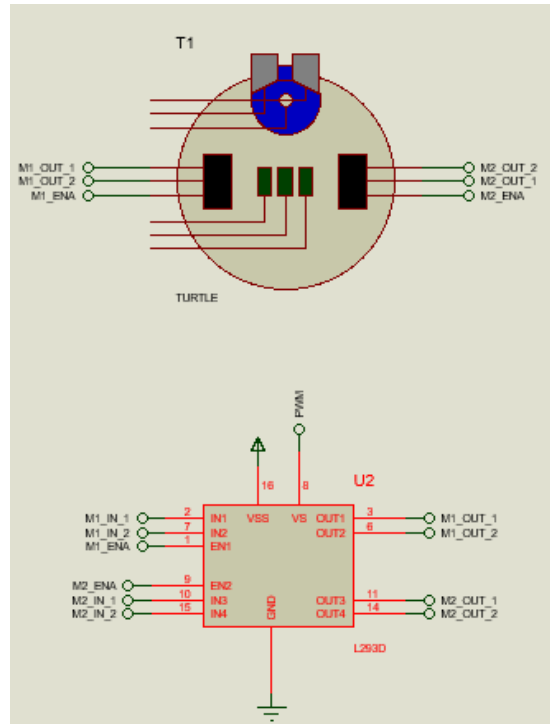


Ilustración 8: Conexión de motores del turtle con el PIC16F877A a partir del L293D

Aplicación

La aplicación para el manejo del robot fue realizada en la aplicación de MIT App Inventor, ya que es una rápida y sencilla plataforma para realizar una aplicación sencilla. El diseño de la aplicación simula un control remoto con las direcciones del robot y el botón de paro.

En la sección de conexión de bluetooth se debe seleccionar la conexión de la computadora. Una vez seleccionado se le notificará al usuario que la conexión fue exitosa con un mensaje debajo de la barra de conexión. Al conectarse ya se podrá manipular el robot por medio de la conexión con el Bluetooth HC-05.

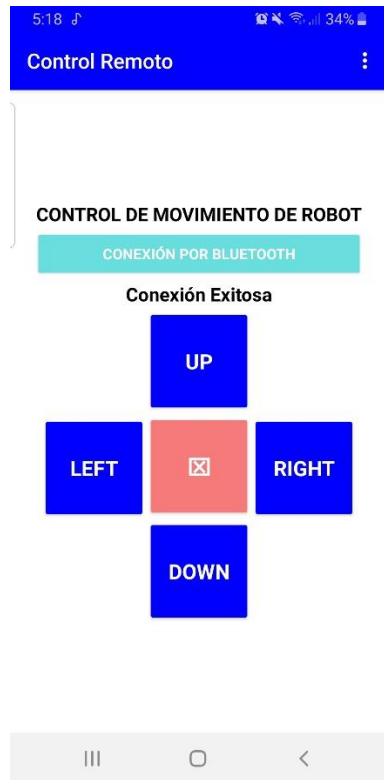


Ilustración 9: Visualización de Aplicación

*Nota: El archivo *.apk también se adjuntará a los*

Temporizador

Para el temporizador de los 2 minutos del programa se realizó la implementación con el TIMER1. Esta interrupción cuenta con el manejo de 0.1041ms, por lo cual se puede obtener el valor de 120 segundos que necesita el programa.

$$T_{TMR1IF} = (0.2\mu s)(8)(65535 - 0) = 104.8856ms$$

La interrupción se activa aproximadamente 9 veces por cada segundo. Para cada 9 veces se aumentará la cuenta de segundos. Una vez que el contador de segundos llegue a 114, debido a un desfase de tiempo, se habrá llegado a los 2 minutos.

$$Cant = \frac{1}{T_{TMR1IF}} = \frac{1}{0.1048857} = 9.53 \approx 9$$

Al llegar a los 2 minutos se cambiará la bandera del ciclo del programa, evitando que se continúe el ciclo de este, los motores se detendrán. También se mandará una notificación del tiempo transcurrido en el display LCD.

Nota: Para llegar a utilizar la interrupción se activó en el main la interrupción “INT_TIMER1” y las interrupciones globales.

Conversión A/D y Manejo de la Modulación de Ancho de Pulso (PWM)

La configuración inicial para el convertidor se realiza la configuración con respecto al reloj interno y se asigna como analógico el puerto A. Solo se utilizará el PIN A0 para obtener la entrada del potenciómetro.

Para la configuración del CCP1, se especifica el uso para la modulación del ancho del pulso. Se debe de configurar el TIMER2, ya que se utiliza para generar la señal.

Cada vez que se actualice el direccionamiento del robot, se dará la conversión a la señal del potenciómetro. Este valor se le asignará al ciclo duty, para la variación de la salida, al pwm1.

La señal obtenida se enviará al L293D para variar la velocidad de los motores del robot.

Selección de direccionamiento

A partir de la obtención de datos, por medio de la conexión vía bluetooth, se realizará la impresión del mensaje, dependiendo de la acción, y el cambio de direccionamiento correspondiente (Entradas 1, 2 y el habilitador). Al finalizar las instrucciones se manda a la salida el direccionamiento por el puerto B.

Esta sección del código está dependiendo de la bandera del tiempo, esta será desactivada cuando se llegue a los 2 minutos (Interrupción de Timer1). Ya no será posible la elección de movimientos por parte del usuario.

Código

A continuación, se anexará el código en el documento, al igual se agregará en el proyecto comprimido.

```

//Proyecto Final
#include <16F877A.h>
#define HS,NOWDT,NOPROTECT
#define device ADC=10
#define use delay(clock=2000000)
//Manejo I2C
#define use i2c(MASTER,SDA=PIN_C4,SCL=PIN_C3,SLOW,NOFORCE_SW)
//Comunicación serial asincrona
#define use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
//Manejo del LCD por I2C
#include <i2c_LCD.c>

#define org 0x1F00, 0x1FFF void loader16F877(void) {}

char option='P';
int tempo=1;
int contador=0;
long seg=0;
long duty=0;
int vel=0;

//Interrupción de desbordamiento de TIMER 1 para contar 2 minutos
#define int_TIMER1
void TIMER1_isr(){
    if(seg==114){
        //Al obtener 2 mins (120 seg) el ciclo del programa terminará
        tempo=0; //Fin del ciclo
        lcd_gotoxy(1,1);

        //Mensaje en display cuando se cumplan los 2 minutos
        printf(lcd_putc,"2 MINUTOS\n");
        lcd_gotoxy(1,2);
        printf(lcd_putc,"TRANSCURRIDOS\n");
        output_b(0x00);
    }else{

        if(contador==9){//Contador para obtener 1 segundo
            contador=0;
            seg++;
            //printf("%ld\n",seg);
        }else{
            contador++;
        }
    }
}

//Función para obtener el PWM para variar la velocidad.
void cambio_vel(){
    set_adc_channel(0); //Selección canal 0
    duty=read_adc(); //Conversión A/D
    set_pwm1_duty(duty); //Asignar ciclo duty a PWM1
}

void main(){

```

```

//TIMER 1

set_timer1(0);
setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
enable_interrupts(INT_TIMER1);
enable_interrupts(GLOBAL);

//Configuración para PWM
setup_adc(ADC_CLOCK_INTERNAL); //Configuración Convertidor A/D
setup_port_a(ALL_ANALOG); //Puerto A como convertidores

setup_ccp1(CCP_PWM); //Uso de CCP1 para el manejo de PWM
setup_timer_2(T2_DIV_BY_4, 255, 1); //Manejo de timer 2 para realizar la
comparación
delay_ms(100);

//Configuración I2C con LCD
i2c_start(); //Inicialización I2C
lcd_init(0x44, 16, 2); //Manejo de LCD por I2C (Dirección 0x44)
                        //Manejo de PCF8574, A2=0, A1=1 A0=0, Escritura=0
                        //Display de 16 x 2

lcd_clear();
lcd_gotoxy(1, 1); //Impresión en (1,1)

printf(lcd_putc, "A CORRER\n\r");
delay_ms(250);

set_tris_b(0x00);
output_b(0x00);

while(tempo==1){
    option=getchar();
    if(tempo==1){
        switch(option){
            case 'A': //Opción de avanzar
                lcd_clear();
                lcd_gotoxy(1, 1);
                printf(lcd_putc, "AVANZANDO");
                //00 101 101
                vel=0x2D;

                break;
            case 'T': //Opción de retroceder
                lcd_clear();
                lcd_gotoxy(1, 1);
                printf(lcd_putc, "RETROCEDIENDO");
                //00 110 110
                vel=0x36;

                break;
            case 'D': //Opción de giro a la derecha
                lcd_clear();
                lcd_gotoxy(1, 1);
                printf(lcd_putc, "GIRO A LA\n\r");
        }
    }
}

```

```

        lcd_gotoxy(1,2);
        printf(lcd_putc,"DERECHA\n");
        //00 000 101
        vel=0x05;
    break;
    case 'I':    //Opción de giro a la izquierda
        lcd_clear();
        lcd_gotoxy(1,1);
        printf(lcd_putc,"GIRO A LA\n");
        lcd_gotoxy(1,2);
        printf(lcd_putc,"IZQUIERDA\n");
        //00 101 000
        vel=0x28;
    break;
    default:    //Opción de parar
        lcd_clear();
        lcd_gotoxy(1,1);
        printf(lcd_putc,"DETENIDO");
        //00 000 000
        vel=0x00;
    break;
}
(PWM) cambio_vel(); //Realiza la lectura del convertidor y cambia el duty
        output_b(vel); //Asigna la dirección a la salida B.
    }
}
}

```

Diagramas de Flujo

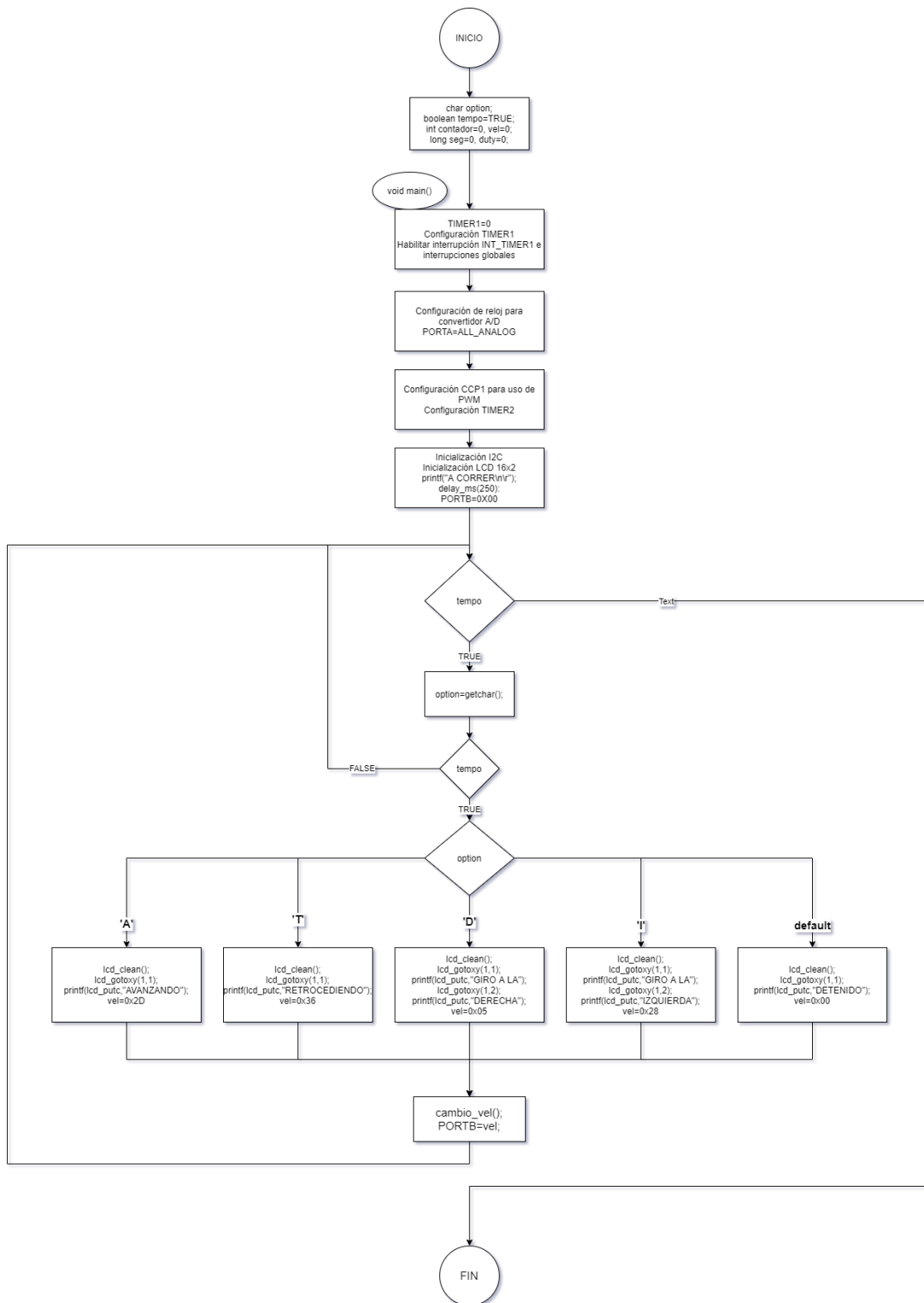


Diagrama 1: Programa principal

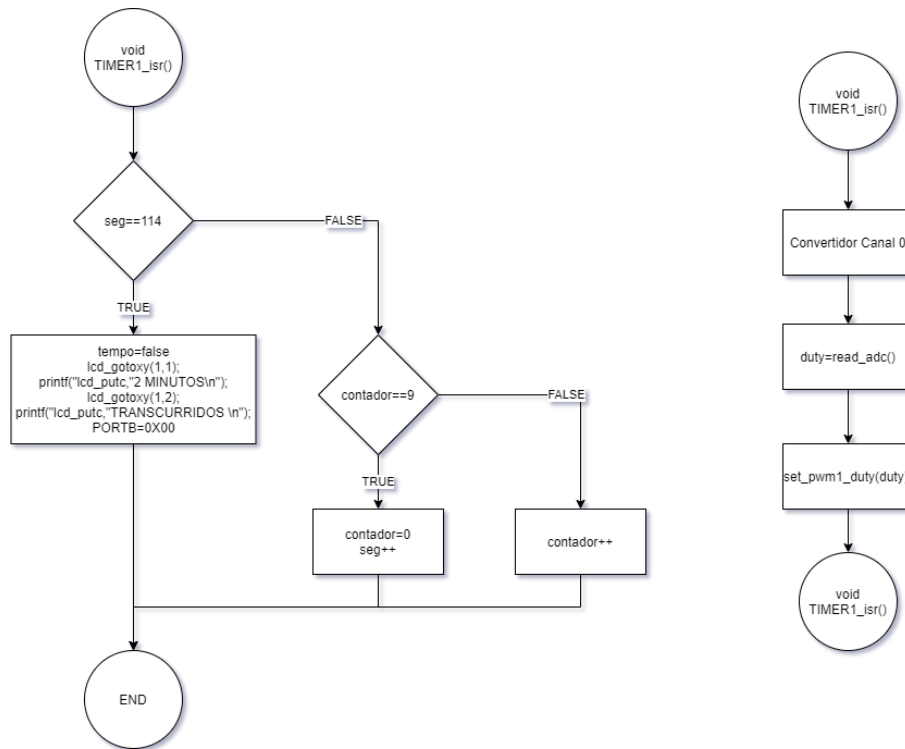
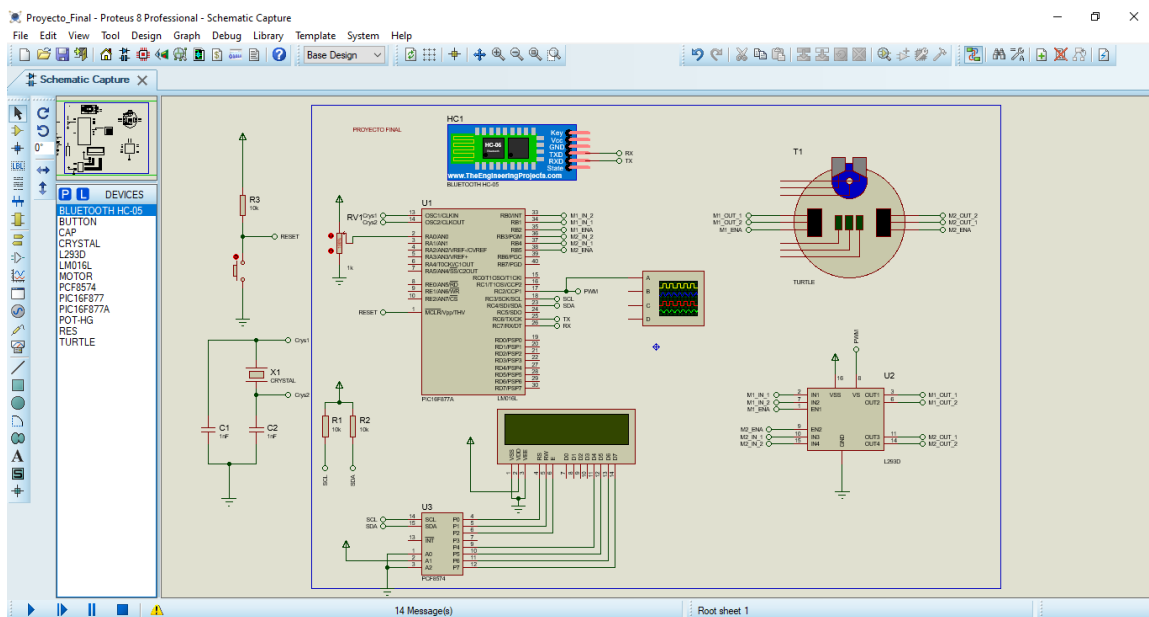
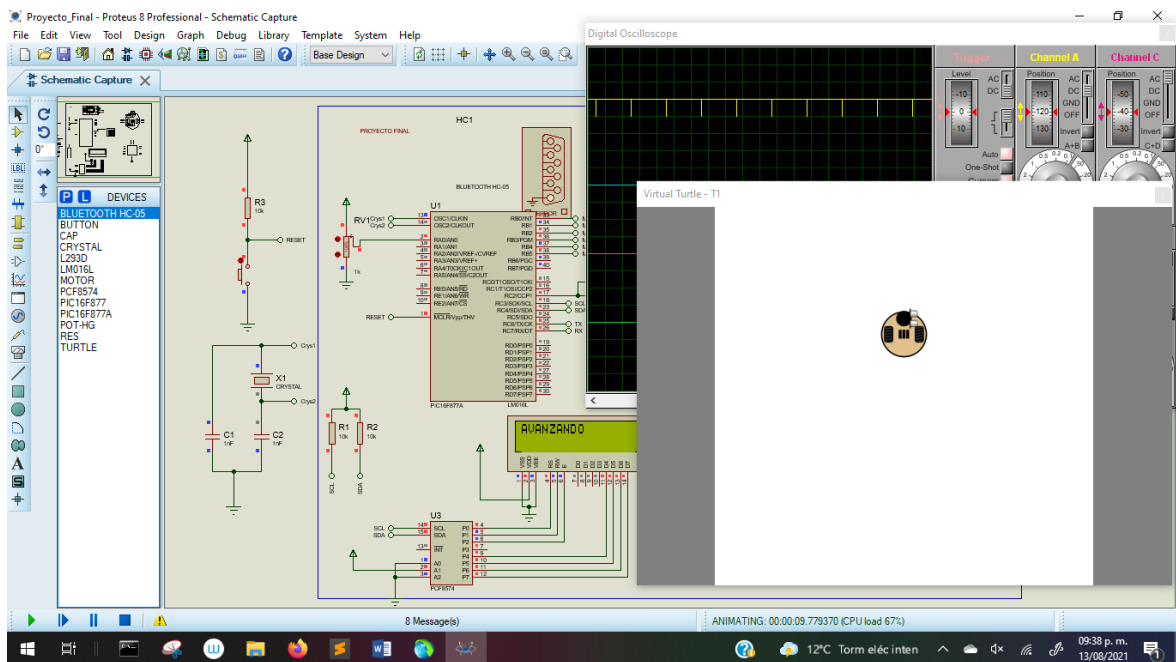
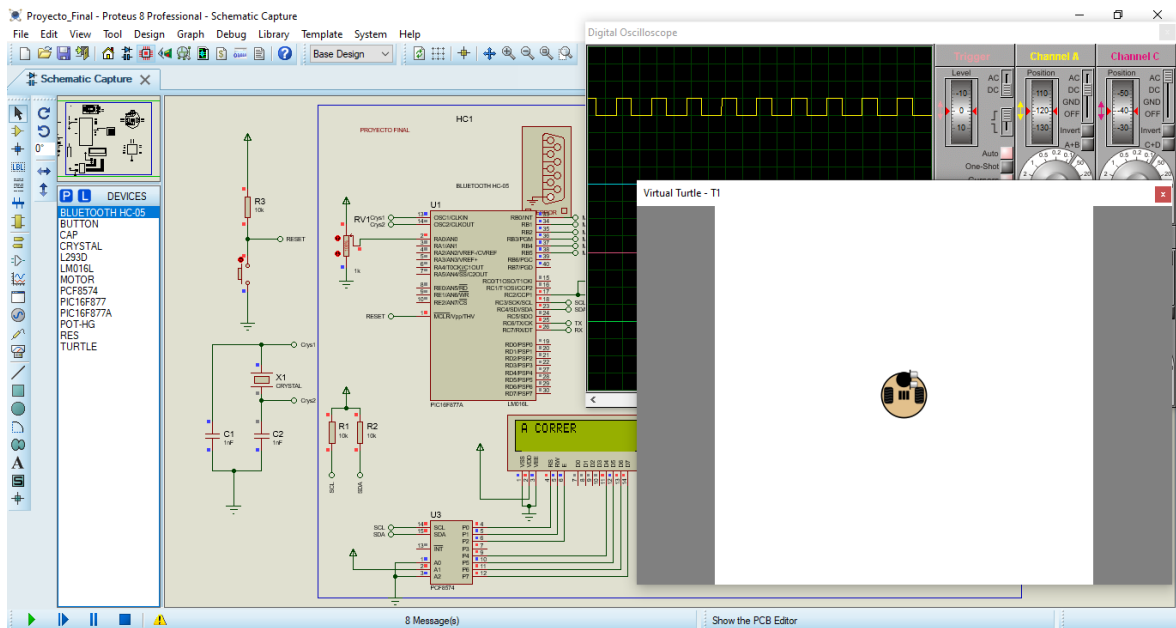


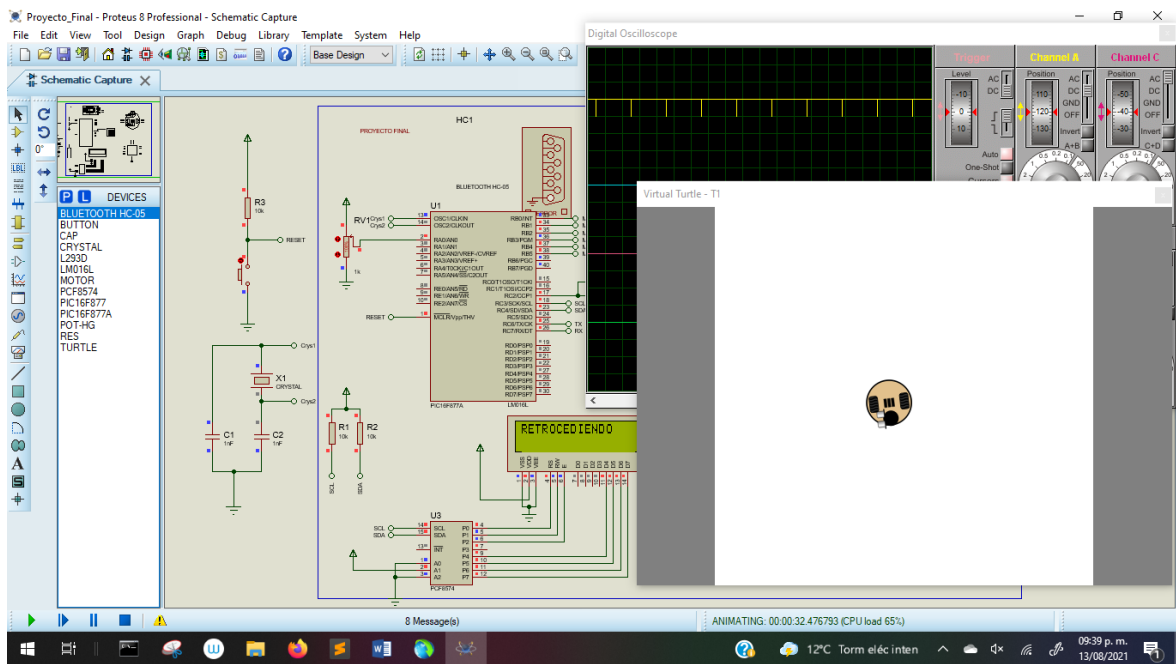
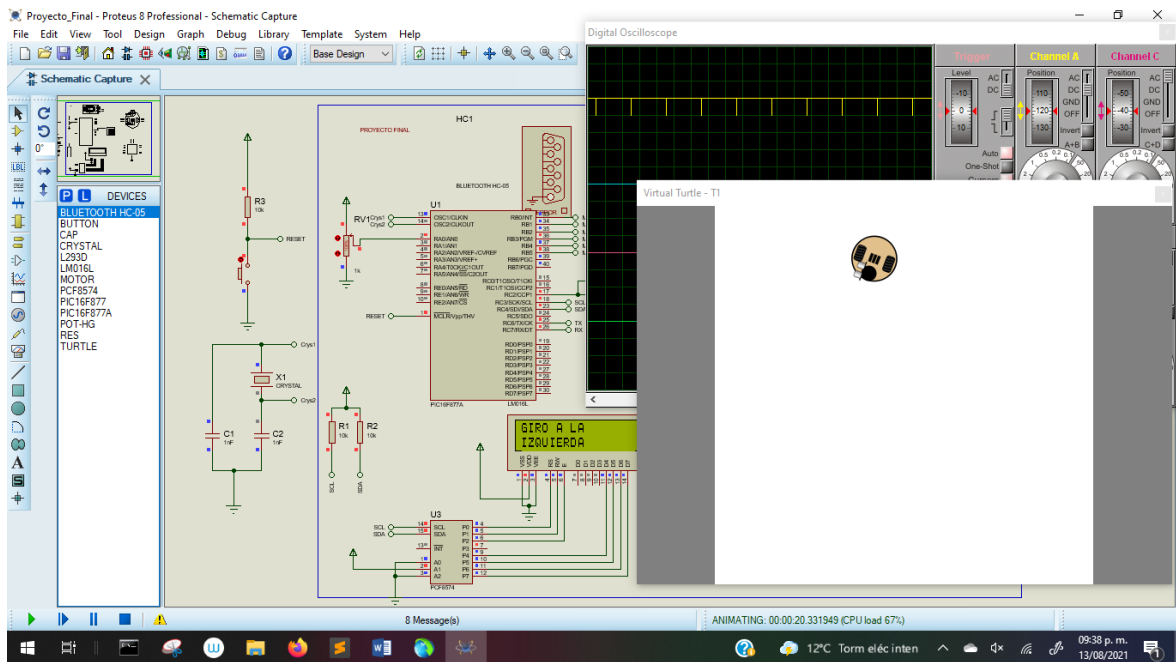
Diagrama 2: Interrupción de TIMER1 y obtención del PWM por medio del convertidor A/D

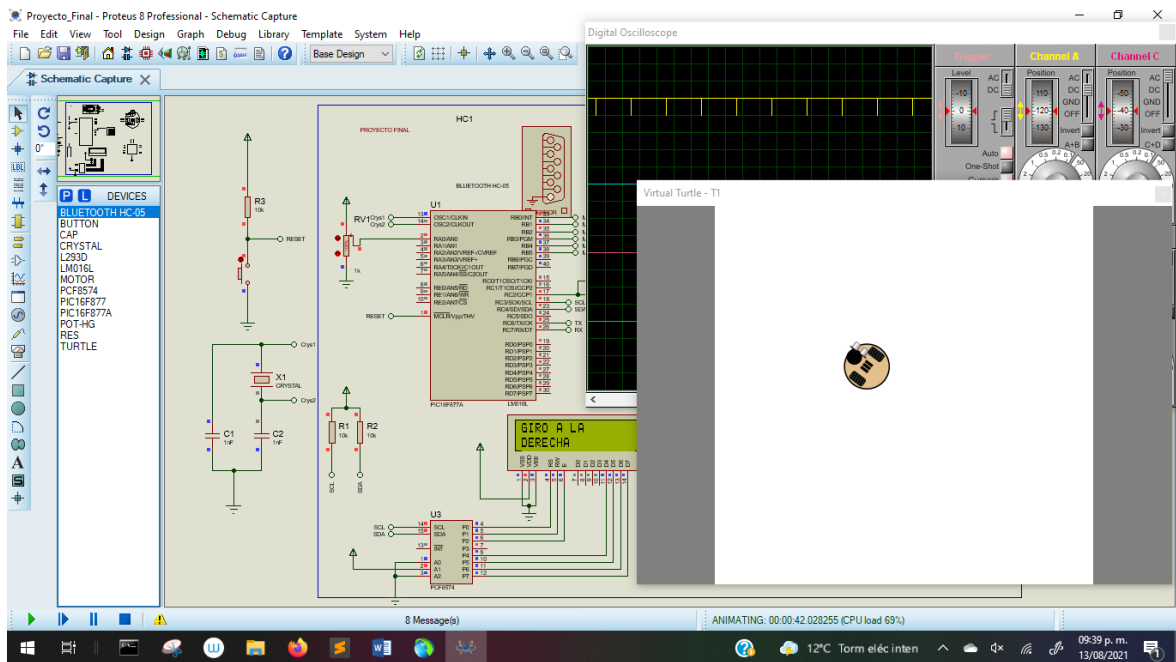
Ejecuciones



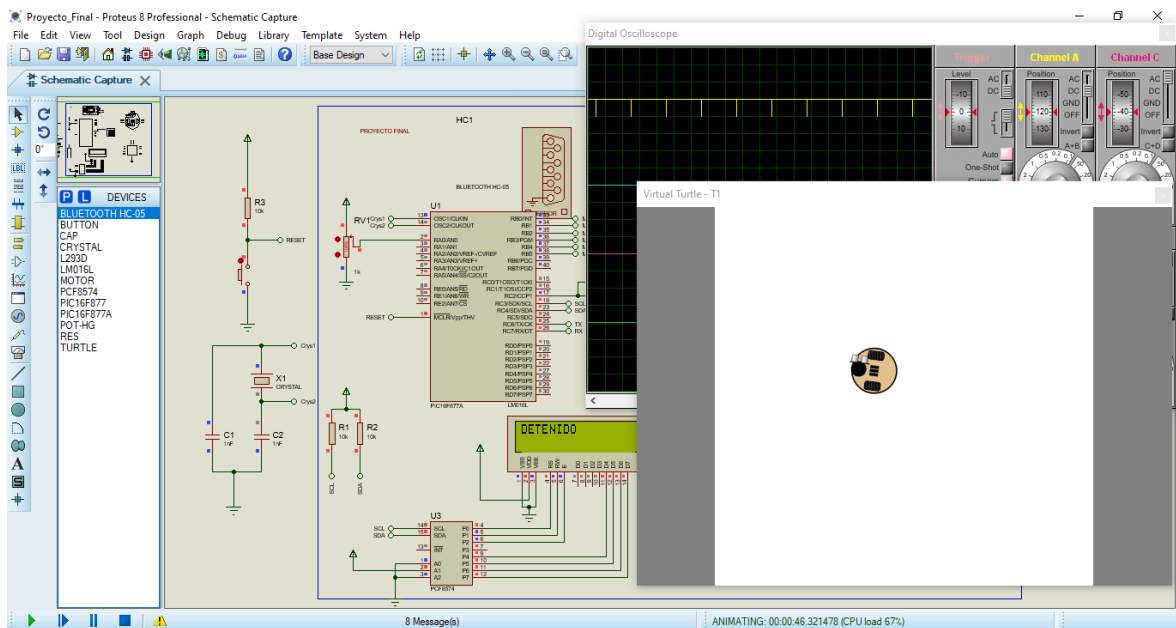
Simulación 1: Simulación base del programa



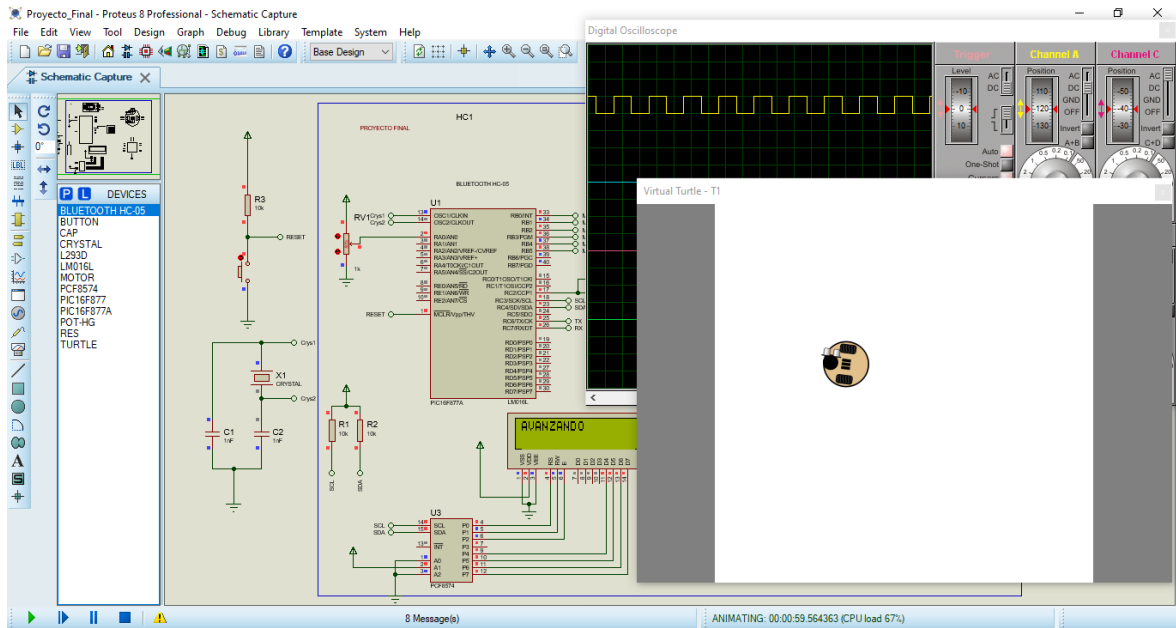




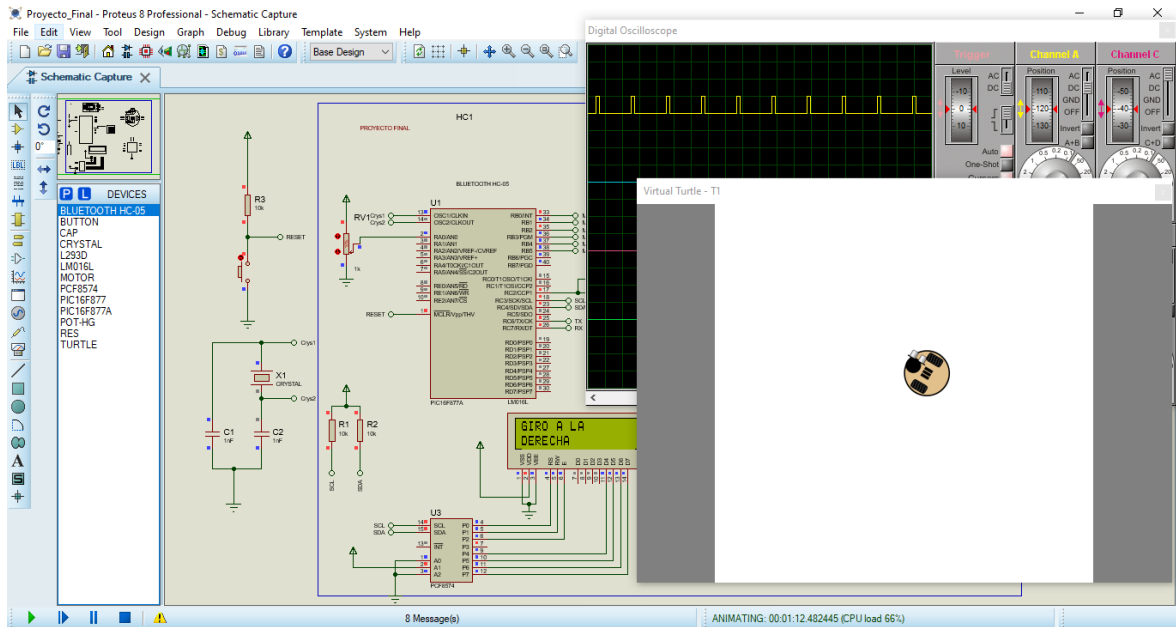
Simulación 6: Giro a la Derecha



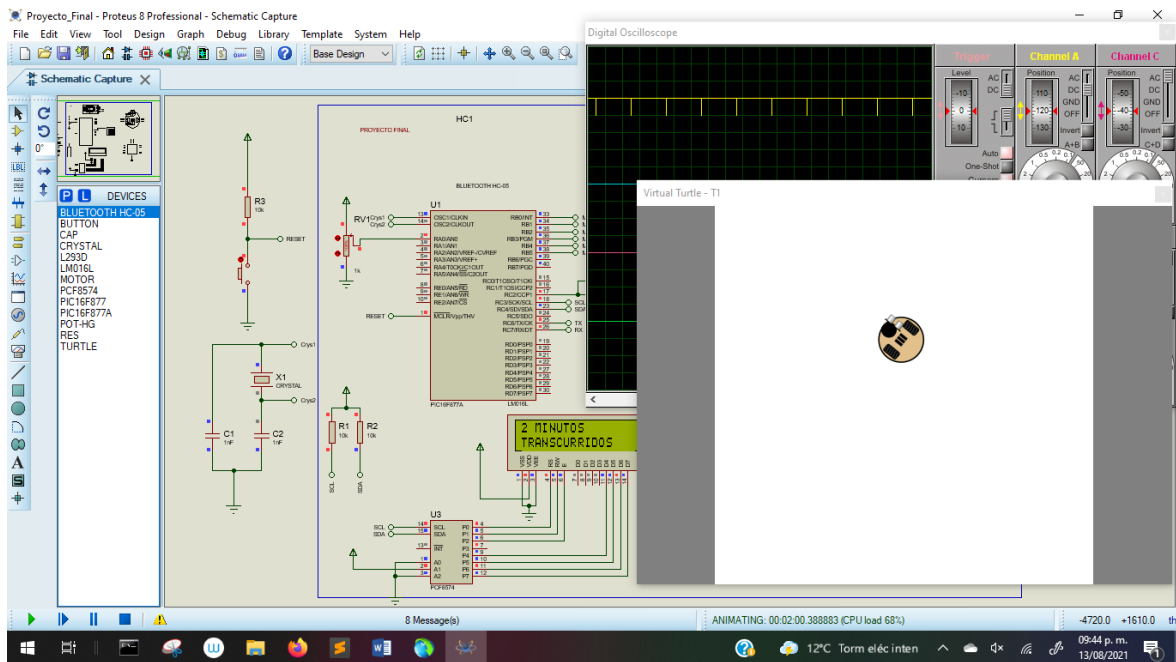
Simulación 7: Paro del robot



Simulación 8: Cambio de velocidad (50%)



Simulación 9: Cambio de velocidad (10%)



Simulación 10: Finalización del programa a los 2 min

Conclusiones

Kennedy Villa Carolina

En este proyecto, aplicamos lo aprendido en esta materia como el uso de los puertos de entrada y salida, dispositivos como LCD y motores, interrupciones, convertidor A/D, entre otros; además de aplicar otros conocimientos de otras materias (como la programación en lenguaje C), para poder crear proyectos de gran alcance. Con ésto, comprendimos las mayores ventajas de un microcontrolador pues podemos diseñar cualquier sistema a bajo costo y grandes aplicaciones.

Fue interesante utilizar otras herramientas para complementar el proyecto, como la aplicación que se desarrolló para controlar el dispositivo del simulador. Sin duda, se añade más valor al proyecto y se puede reutilizar para aplicaciones futuras.

Murrieta Villegas Alfonso

El presente proyecto final es el resultado de la comprensión y uso de varios conceptos que se vieron a lo largo del semestre, a priori el uso de puertos de entrada y salida como medio para obtener información e interactuar con ella, además y debido a la relación con

estos se empleó comunicación serie para trabajar con hardware externo, como es el caso de motores de corriente directa mediante pwm, además de pantallas LCD y el uso de terminales, también se trabajó con otros aspectos de esta materia como el uso de DAC's al trabajar con distintos tipos de señales.

Por último, cabe destacar que este proyecto se caracterizó por relacionar lo visto en esta materia con otras como es dispositivos electrónicos e incluso el programar una aplicación móvil para controlar remotamente a nuestro robot.

Reza Chavarria Sergio Gabriel

A partir de los conceptos de la materia como lo fueron el uso de puertos de entrada y salida, convertidor A/D, interrupciones, los tipos de comunicación serie y del uso de elementos externos como motores, displays y terminales se puede dar una implementación compleja, como lo fue el proyecto, y darle un uso en conjunto.

Esta materia implica el entendimiento del manejo de estos dispositivos para llevarlos a un uso bastante práctico. Bastantes elementos que utilizamos actualmente tienen implementado el uso de microcomputadoras.

Valdespino Mendieta Joaquín

En el presente proyecto pudimos reforzar, aplicar e integrar los conocimientos adquiridos a lo largo del semestre en la materia de microcomputadoras, con el uso de los elementos que contiene nuestro micro, desde los puertos en modo entrada o salida, que permite la interacción con el flujo de información con el exterior, así como el uso conjunto de otros dispositivos que añaden funcionalidad al proyecto o en su caso implementaciones, como los motores aplicados, lcd, convertidor A/D, además de conceptos como interrupciones, comunicación serie etc. Todo ello complementa al llevarlos a un trabajo práctico.

Bibliografía

- Nasir, S. (2016). Bluetooth Library for Proteus. Encontrado en agosto 11, 2021, de The Engineering Projects. Sitio web: <https://www.theengineeringprojects.com/2016/03/bluetooth-library-for-proteus.html>
- Castaño, S. (2020). PWM con PIC – (Modulación por Ancho de Pulso). Encontrado en agosto 11, 2021, de Control Automático Educación. Sitio Web: <https://controlautomaticoeducacion.com/microcontroladores-pic/pwm-modulacion-por-ancho-de-pulso/>
- Castaño, S. (2020). LCD I2C PIC. Encontrado en agosto 11, 2021, de Control Automático Educación. Sitio Web: <https://controlautomaticoeducacion.com/microcontroladores-pic/lcd-i2c-pic/>