M16

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Semestre 2023-2

Laboratorio de Microcomputadoras

Proyecto Final de Laboratorio

Profesor:

M.I. Ruben Anaya García

Alumnos:	Número de cuenta:
Barreiro Valdez Alejandro	317520888
Zepeda Baeza Jessica	317520747

Grupo Laboratorio: 4

Grupo Teoría: 1

Fecha de realización: 3 de junio de 2023

Fecha de entrega: 7 de junio de 2023

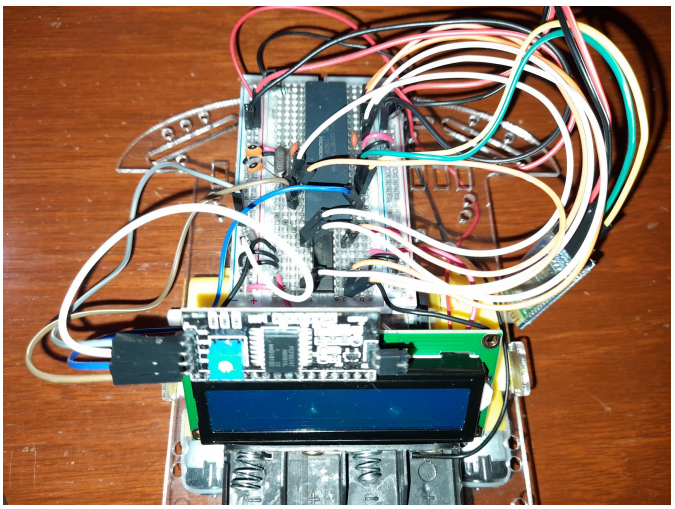
Desarrollo

En este proyecto final, hemos construido un coche funcional que abarca la mayoría de los conceptos aprendidos durante nuestro curso de laboratorio. Utilizando la comunicación en serie asíncrona, creamos un coche capaz de recibir datos y controlar sus movimientos. Inspirados en uno de nuestros ejercicios anteriores, implementamos un sistema para modificar los valores de dos motores y lograr que el coche se mueva hacia delante, hacia atrás, hacia la izquierda y hacia la derecha. Todo esto fue posible gracias al módulo L293 y al puerto de salida B.

Además, reescribimos el código en lenguaje C para aplicar los conocimientos más recientes. Agregamos dos características adicionales relacionadas con nuestras últimas prácticas. Incorporamos un circuito PCF8574 para utilizar el protocolo I2C en una pantalla de cristal líquido. Esto nos permitió mostrar la cantidad de metros recorridos por el coche mientras avanza o retrocede. También utilizamos interrupciones por Timer 0 para llevar un seguimiento del tiempo y realizar operaciones específicas.

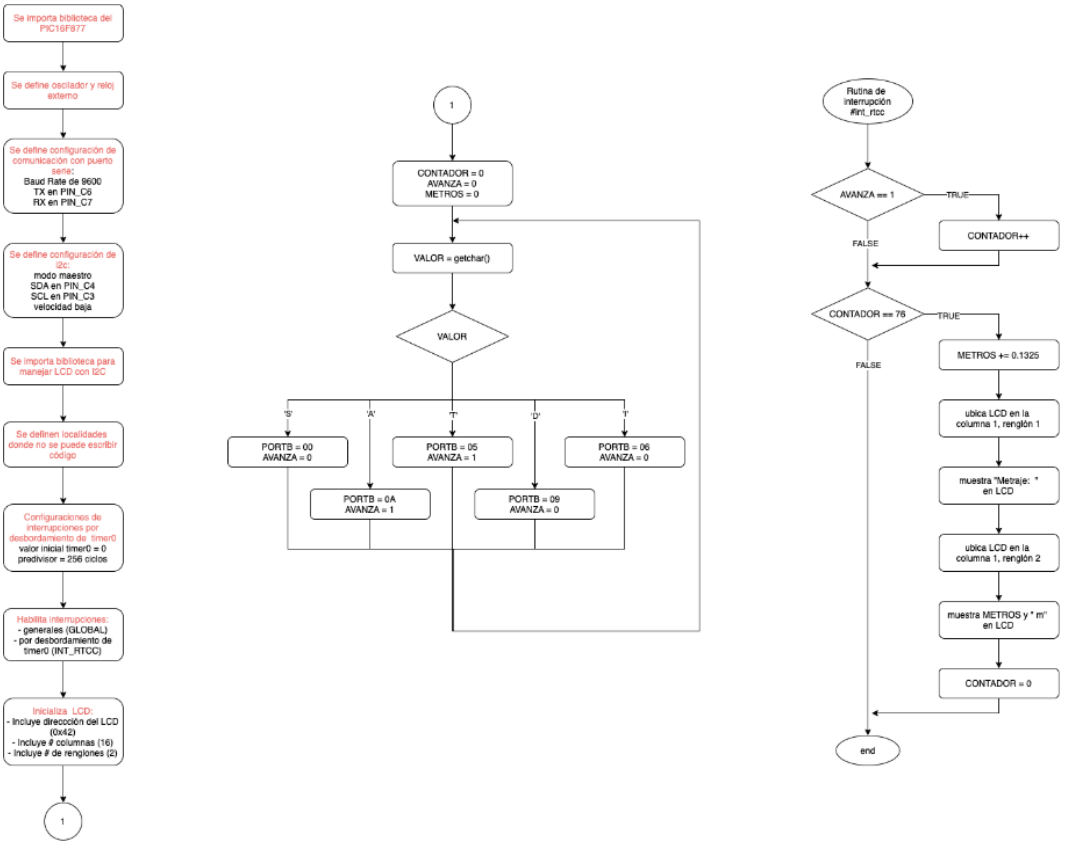
Finalmente, utilizamos la comunicación en serie asíncrona para probar y depurar el código. Después de las pruebas, integramos un módulo Bluetooth para controlar el coche a través de una aplicación que desarrollamos. Con todas estas características, logramos crear un coche funcional que registra la distancia recorrida utilizando el protocolo I2C y las interrupciones de tiempo.

Para la comunicación entre el robot y una aplicación móvil, se utilizará un módulo Bluetooth. Este módulo se conectará a los pines de transmisión (TX) y recepción (RX) del microcontrolador, que se configurará como receptor de los comandos enviados desde la aplicación móvil. Además, es importante utilizar baterías adecuadas para alimentar eficientemente el sistema, incluyendo el microcontrolador, los motores y los demás componentes del robot. Por último, se agregó el display con el circuito mencionado y se conectan los pines de SDA y SCL a C3 y C4. El resultado fue el siguiente.



Algoritmo

La carta ASM del programa se muestra a continuación:



Primero se incluyeron distintas directivas e instrucciones iniciales para configurar el microcontrolador. Para empezar, se importa la biblioteca que maneja el PIC16F877, se define el oscilador y el reloj interno, se configura la recepción y transmisión de datos por el puerto serie, se configura la comunicación i2c, se importa la biblioteca que maneja el LCD

por i2c y se inicializa, se definen las localidades donde no se puede escribir código y por último se hace la configuración de la interrupción por TIMER0 (definiendo predivisor y valor inicial) así como también se habilitan las interrupciones generales y la de desbordamiento por TIMER0.

Después se incializan tres variables en 0: CONTADOR, AVANZA y METROS. También se tiene otra variable VALOR que toma el valor del caracter recibido por el puerto serie mediante la función *getchar()*. Dependiendo el caracter en VALOR se modifica el valor del Puerto B para mover al coche hacia delante (‘A’, Puerto B = 0A), hacia atrás (‘T’, Puerto B = 05), girar a la derecha (‘D’, Puerto B = 09), girar a la izquierda (‘D’, Puerto B = 06) y detenerse (‘S’, Puerto B = 00). También, dependiendo VALOR se activa (1) o desactiva (0) la bandera AVANZA para contar los metros que ha recorrido el coche. Solamente al momento de ir para delante o atrás AVANZA es igual a 1, en los demás casos es 0. Después de modificar estas variables el programa regresa a recibir otro caracter.

Se implementa también la interrupción de TIMER0 en una rutina que primero verifica si AVANZA es 1 para incrementar un contador. El contador cuenta hasta 76 (1 segundo) para aumentar los metros recorridos ya que se asume que el coche recorre 0.1325 m/s. Entonces en caso de que CONTADOR sea igual a 76, se aumenta 0.1325 a la variable METROS, se imprime “Metraje: “ en el primer renglón del LCD y METROS en el segundo renglón y para finalizar, CONTADOR se vuelve 0 de manera que puede volver a contar 1 segundo.

Programa comentado

```
#include <16F877.h>                                          //Biblioteca PIC16F877A
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)                                   //Configurando el reloj
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)               //Configuración puerto en serie
#use i2c(MASTER,SDA=PIN_C4,SCL=PIN_C3,SLOW,NOFORCE_SW)
/*
Directiva para protocolo I2C, se configura:
-Modo maestro
-SDA en Pin C4
-CSL en Pin C3
-Velocidad baja
*/
#include <i2c_LCD.c>                                         //Biblioteca LCD para I2C
#org 0x1F00,0x1FFF void loader16F877(void){}

int contador,avanza;
/*
contador:   Contador de Timer0
avanza:     Bandera de si avanza o no
*/
float metros;                                                //Metros recorridos
char valor;                                                  //Valor de control

#int_rtcc
clock_isr(){
   if(avanza == 1)                                           //Si avanza se incrementa el contador
      contador++;                                            //Incrementa el contador

   if(contador == 76){                                       //Cada segundo
      metros += 0.1325;                                      //Se aumentan los metros en .13
      lcd_gotoxy(1,1);
      printf(lcd_putc,"Metraje: \n");
      lcd_gotoxy(1,2);
      printf(lcd_putc,"%f m\n",metros);                      //Imprimir en LCD: "Metraje: metros"
      contador = 0;                                          //Se reinicia el contador
   }
}

```

```c
void main() {
   set_timer0(0);                                          //Inicia Timer en 00h
   setup_counters(RTCC_INTERNAL,RTCC_DIV_256);             //Fuente de reloj y pre-divisor
   enable_interrupts(INT_RTCC);                            //Habilita interrupción de Timer0
   enable_interrupts(GLOBAL);                              //Habilita interrupciones globales

   lcd_init(0x4E, 16, 2);                                  //Incializa LCD en 4E

   contador = 0;                                           //Inicializa contador interrupción en 0
   avanza = 0;                                             //Bandera por si avanza
   metros = 0;                                             //Variable de metros recorridos

   while( TRUE ) {
      valor = getc();                                      //Recepción de un caracter
      if(valor == 'S'){                                    //Si es S
         avanza = 0;                                       //Bandera en no avanza
         output_b(0x00);                                   //Se para el coche
      }
      if(valor == 'A'){                                    //Si es A
         avanza = 1;                                       //Bandera en avanza
         output_b(0x0A);                                   //Avanza el coche
      }
      if(valor == 'T'){                                    //Si es T
         avanza = 1;                                       //Bandera en avanza
         output_b(0x05);                                   //Retrocede el coche
      }
      if(valor == 'D'){                                    //Si es D
         avanza = 0;                                       //Bandera en no avanza
         output_b(0x09);                                   //Gira a la derecha
      }
      if(valor == 'I'){                                    //Si es I
         avanza = 0;                                       //Bandera en no avanza
         output_b(0x06);                                   //Gira a la izquierda
      }
      printf("%c", valor);                                 //Imprime valor leído en terminal
   }
}
```

Conclusiones y/o comentarios

Zepeda Baeza Jessica:

En este proyecto final, se implementaron prácticamente todos los conceptos, periféricos y protocolos vistos a lo largo del curso en un programa en C que hace uso del microcontrolador PIC16F877. Además de ello, se logró construir el prototipo de un coche que utiliza motores como llantas y otros dispositivos electrónicos como el driver L293 para poder controlar los motores, el PCF8574 para la comunicación i2c y el H05 para la comunicación por Bluetooth. En general es un proyecto que implementa la comunicación asíncrona para controlar la dirección de un coche asignando un cierto valor a uno de sus puertos paralelos y que al hacer esto, el microcontrolador genera interrupciones que van calculando el metraje recorrido y que lo muestran en un display LCD mediante la comunicación i2c desde el microcontrolador. Aunque el código en sí es sencillo debido a que en su mayoría está compuesto por ciclos y asignaciones sencillas, considero que es un sistema bastante completo. Lo que me hace concluir que no se necesitan muchas instrucciones ni código para formar un sistema funcional en ensamblador además de que se tienen todos los registros y puertos y demás cercanos para poder generar el código más eficiente y saber exactamente de dónde a dónde se manda la información.

Barreiro Valdez Alejandro:

En conclusión, este proyecto final ha culminado exitosamente con la creación de un coche funcional que incorpora una amplia gama de conceptos y técnicas abordadas

durante el curso. Mediante la implementación de la comunicación en serie asíncrona, la utilización de módulos como el L293 y el PCF8574, y la aplicación de interrupciones por Timer 0, hemos logrado desarrollar un vehículo capaz de realizar movimientos controlados y llevar un registro preciso de la distancia recorrida. La integración del módulo Bluetooth para el control del coche mediante una aplicación personalizada ha demostrado la aplicabilidad práctica de los conocimientos adquiridos. En definitiva, este proyecto representa un hito importante en la aplicación práctica de los conceptos aprendidos, destacando la capacidad de innovación y resolución de problemas de nuestro equipo.