INFORME DE LABORATORIO #1

ENTRADAS Y SALIDAS DIGITALES

Jeisson Alexander Giraldo Tique
e-mail: jagiraldot@itc.edu.co
Johan Sebastián Martínez García
e-mail: jsmartinezg@itc.edu.co
Escuela Tecnológica Instituto Técnico Central (ETITC)

RESUMEN **- En la actualidad, existe una gran aplicación de los microcontroladores en la sociedad, tales como en electrodomésticos, automóviles o alarmas, por tal motivo este informe de laboratorio consiste en la configuración y programación de las entradas y salidas digitales del microcontrolador PIC 18F4550 con la intención de comprender el manejo de este componente y los elementos que sean necesario para su buen funcionamiento ante un problema determinado.**

PALABRAS CLAVE **-** ***Microcontrolador, PIC 18F4550, configuración, programación, simulación, entradas y salidas digitales***.

I. INTRODUCCIÓN

Este informe de laboratorio se desarrolló con el fin de realizar los procedimientos seguidos y los resultados obtenidos tanto teóricos como prácticos en la materia de microcontroladores, dictada por el ing. Jorge Cote; el laboratorio tuvo como tema principal programar, configurar y simular las entradas y salidas digitales de un microcontrolador PIC 18F4550, por medio de programas y lenguajes de programación tales como proteus, MPLAB, XC8, Lenguaje C.

OBJETIVOS:

- Configurar el entorno de programación de MPLAB XC8 para la programación del PIC 18F4550.
- Inicializar los puertos de entradas y salidas digitales para el PIC 18F4550.
- Utilizar los puertos digitales de entrada y salida para recibir o enviar información a través de estos.

II. PROCEDIMIENTO

El procedimiento para la práctica de laboratorio inicio con la configuración del programa MPLAB para el manejo del microcontrolador PIC18F4550 donde se realizó la programación de cada ejercicio dejada por el Ing. Jorge cote.

El primer código de programación que se realizó fue mover 2 unos lógicos de izquierda a derecha y de derecha a izquierda a lo largo del puerto D del PIC 18F4550, de tal manera que se crucen justo en la mitad del puerto, así mismo se utilizó leds para visualizar el desplazamiento de los bits.
En las siguientes figuras se podrá Visualizar unas partes del código del primer ejercicio planteadas en la guía de laboratorio.

```
#include <xc.h>
#define _XTAL_FREQ 4000000


void main(void) {
    TRISD = 0b00000000;
    TRISA=0b00000001;
    PORTD=0b00000000;
    PORTA=0b00000001;
    ADCON1 = 0b00001111;
    if (PORTAbits.RA2==1)
    {
```

Figura N.1 Entrada y Salida del puerto A y D

```
ADCON1 = 0b00001111;
if (PORTAbits.RA2==1)
{
    PORTDbits.RD0=0b00000001;
    PORTDbits.RD7=0b00000001;
    __delay_ms(500);
    PORTDbits.RD0=0b00000000;
    PORTDbits.RD7=0b00000000;
    __delay_ms(500);
    PORTDbits.RD1=0b00000001;
    PORTDbits.RD6=0b00000001;
    __delay_ms(500);
    PORTDbits.RD1=0b00000000;
    PORTDbits.RD6=0b00000000;
    __delay_ms(500);
    PORTDbits.RD2=0b00000001;
```

Figura N.2 Inicializar y apagar el Puerto D

En la figura N.1 se puede observar que se inicializa el puerto D como salida y el puerto A como entrada, donde específicamente en el pin A2 tendrá un Pulsador que mande un 1 lógico por la cual el código analice si cumple la condición " if (PORTAbits.RA2==1)", si la cumple se inicializa el movimiento de bits que se puede visualizar en la figuras N.2, por otro lado se puede observar la funcionalidad del código por medio de un software de simulación "proteus 8.6" donde se observara los movimiento de bits por medio de led que se puede ver en la siguientes figuras:

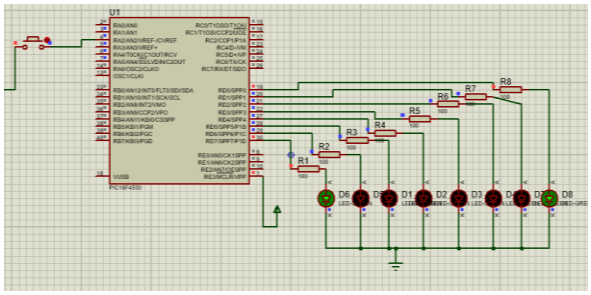

Figura N.3 Simulación en proteus 8.6 movimiento de 2 bits

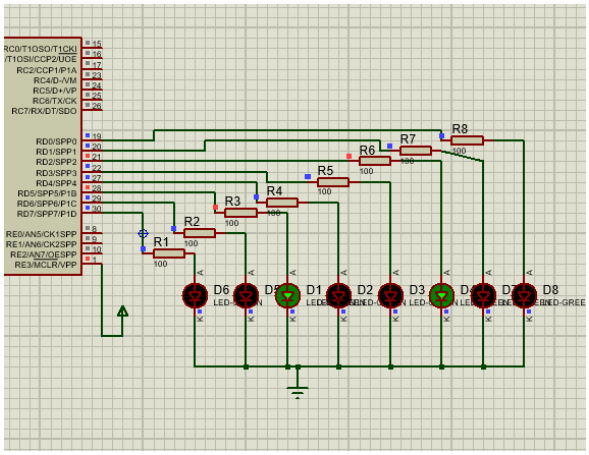

Figura N.4 Simulación en proteus 8.6 movimiento de 2 bits

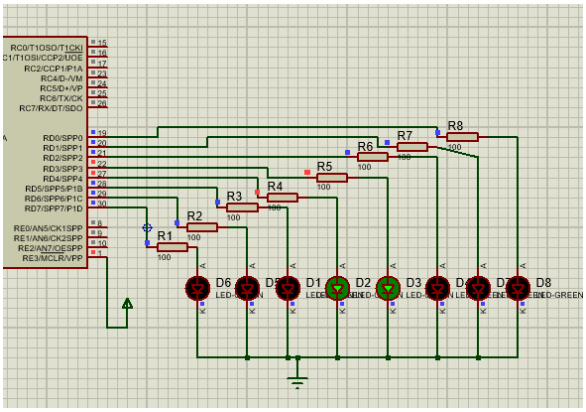

Figura N.5 Simulación en proteus 8.6 movimiento de 2 bits

En el segundo ejercicio el código de programación que se realizó fue Encender un led del puerto cada 3 segundos de forma consecutiva de tal manera que al finalizar el bit 8 (más significativo) del puerto B se tenga una línea de leds encendidos, luego posteriormente debe empezar a apagarse de a un led a la vez comenzando por el bit menos significativo hasta apagar completamente el puerto, todo el código se ejecuta al oprimir un pulsador que está conectado en el pin A2.

Para inicializar Para el conteo de tiempo se utilizó la función" _delay_ms()"que por consiguiente también debe incluir el comandó XTAL_FREQ_4000000 para que funcione adecuadamente, además en la función de delay se le coloca 3000 milisegundos que equivale a 3 segundos esto se puede observar en la siguientes figuras:

```
#include <xc.h>
#define _XTAL_FREQ 4000000


void main(void) {
    TRISB=0b00000000;
    TRISA=0b00000001;
    PORTB=0b00000000;
    PORTA=0b00000001;
    ADCON1=0b00001111;

    if (PORTAbits.RA2==1)
    {
```

Figura N.6 función #define __XTAL_FREQ 400000

```
ADCON1=0b00001111;

if (PORTAbits.RA2==1)
{
    PORTBbits.RB0=0b00000001;
    __delay_ms(3000);
    PORTBbits.RB1=0b00000001;
    __delay_ms(3000);
    PORTBbits.RB2=0b00000001;
    __delay_ms(3000);
    PORTBbits.RB3=0b00000001;
    __delay_ms(3000);
    PORTBbits.RB4=0b00000001;
    __delay_ms(3000);
    PORTBbits.RB5=0b00000001;
    __delay_ms(3000);
    PORTBbits.RB6=0b01000001;
    __delay_ms(3000);
    PORTBbits.RB7=0b10000001;
    __delay_ms(3000);
    PORTBbits.RB0=0b00000000;
    __delay_ms(3000);
   PORTBbits.RB1=0b00000000;
   __delay_ms(3000);
   PORTBbits.RB2=0b00000000;
   __delay_ms(500);
   PORTBbits.RB3=0b00000000;
   __delay_ms(3000);
```

Figura N.7 función __delay_ms(3000).

por otro lado se puede observar la funcionalidad del segundo ejercicio por medio de un software de simulación “proteus 8.6” donde se visualizara la secuencia de encendido de los leds del menos significativo al más significativo y luego el apagado del menos significativo al más significativo que están conectados al puerto B obstante se puede observar en la siguientes figuras:

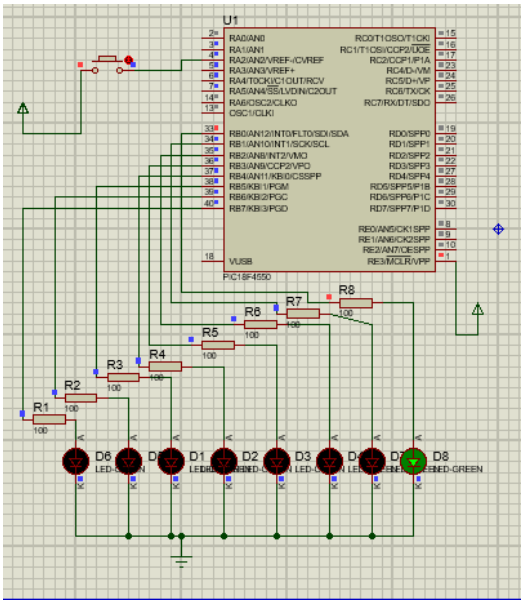

Figura N.8 Encendido del led del menos significativo al más significativo.

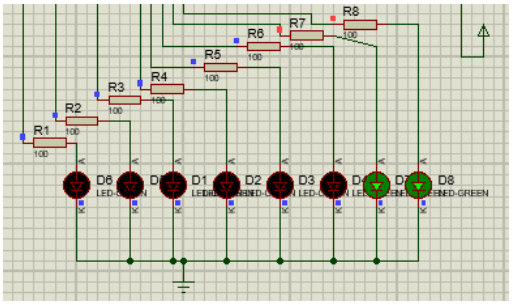

Figura N.9 Encendido del led del menos significativo al más significativo.

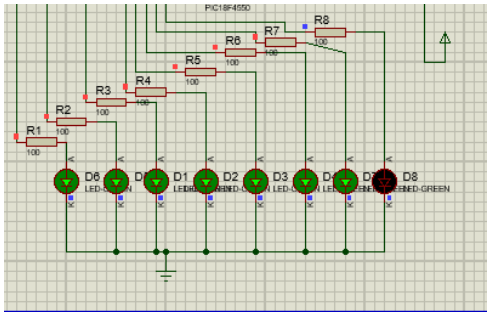

Figura N.10 apagado del led del menos significativo al más significativo.

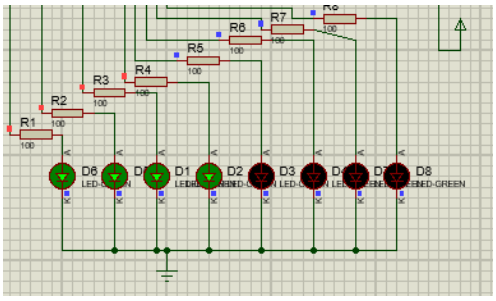

Figura N.11 apagado del led del menos significativo al más significativo.

En el tercer ejercicio el código de programación que se realizó fue Utilizar 4 pines del puerto A para mostrar un conteo en código gray que debe ir cambiando cada 1 segundo, todo el código se ejecuta al oprimir un pulsador que está conectado en el pin A2, y como cuenta con 4 pines se podrá contar de 0 a 15 en número decimal.
Una parte del código se podrá observar en las siguientes figuras:

```
#include <xc.h>
#define _XTAL_FREQ 4000000


void main(void) {
    TRISA=0b00000000;
    ADCON1=0b00001111;
    PORTA=0;
    if (PORTAbits.RA2==1){
        __delay_ms(500);
        PORTA=0b0000;
        __delay_ms(1000);
        PORTA=0b0001;
        __delay_ms(1000);
```

Figura N.12 código gray.

```
ORTA=0;
f (PORTAbits.RA2==1){
    __delay_ms(500);
    PORTA=0b0000;
    __delay_ms(1000);
    PORTA=0b0001;
    __delay_ms(1000);
    PORTA=0b0011;
    __delay_ms(1000);
    PORTA=0b0010;
    __delay_ms(1000);
    PORTA=0b0110;
    __delay_ms(1000);
    PORTA=0b0111;
    __delay_ms(1000);
    PORTA=0b0101;
    __delay_ms(1000);
    PORTA=0b0100;
    __delay_ms(1000);
    PORTA=0b1100;
    __delay_ms(1000);
    PORTA=0b1101;
```

Figura N.13 código gray.

En la figura N.12 se puede observar que se inicializa el puerto A de tal manera que se pueda visualizar el código gray en los 4 pines que van del pin A0 hasta el pin A3 y su secuencia de activación se puede observar una parte del código en la figura N.13, Además también se utilizó la función __delay_ms configurada en 1 segundo.
por otro lado, se puede observar la funcionalidad del tercer ejercicio por medio del software ya usado anteriormente donde se visualizará el conteo en código gray que cambia cada 1 segundo del 0 hasta 15 número decimal no obstante se puede observar en las siguientes figuras:

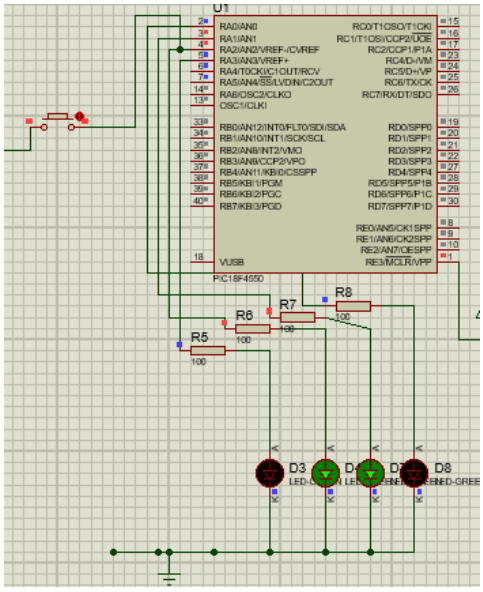

Figura N.14. visualización del 4 decimal en código gray.

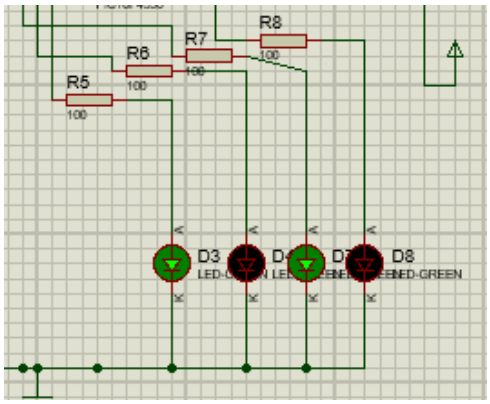

Figura N.15. visualización del 12 decimal en código gray.

En el cuarto ejercicio el código de programación que se realizó fue utilizar el pin A1 como salida para generar una señal cuadrada de que dure el 70% del periodo en nivel alto y el resto en nivel bajo. La frecuencia de la señal debe ser de 1Hz además se verifico con el osciloscopio del simulador la señal cuadrada.
Se calcula el 70 % de 1000 milisegundos que equivale un segundo dando como 700 milisegundos y el otro 30 % seria 300 milisegundos que seria el bajo de la señal cuadrada.
Una parte del código se podrá observar en las siguientes figuras:

```
#include <xc.h>
#define _XTAL_FREQ 1000000
void main(void) {
TRISA=0;
PORTA=0;
ADCON1=0b0001111;
if(PORTAbits.RA2==1){
PORTAbits.RA1=0b00000001;//
__delay_ms(700);
PORTAbits.RA1=0b00000000;
__delay_ms(300);

}
    return ;
```

Figura N.16. Señal cuadrada.

por otro lado, se puede observar la funcionalidad del cuarto ejercicio por medio del software ya usado anteriormente, donde se visualizará la señal cuadrada y además tendrá un led donde también se podrá ver el alto y bajo de la señal, sin embargo, hay que tener muy represente que el osciloscopio este configurado en DC si no la gráfica estará errónea, no obstante, se puede observar en las siguientes figuras:

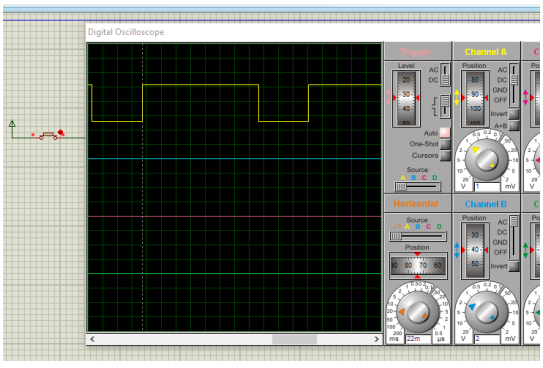

Figura N.17. Simulación de señal cuadrada por osciloscopio "proteus 8.6"

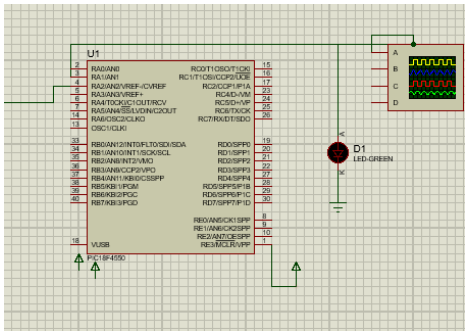

Figura N.18. Simulación de señal cuadrada por osciloscopio "proteus 8.6"

CONCLUSIONES

- Durante la práctica, se pudo analizar y comprender las entradas y salidas digitales del microcontrolador PIC18F4550, además del lenguaje y entorno de programación de MPLAB y XC8, el cual permitió realizar la programación de las acciones del PIC. Durante el desarrollo del laboratorio hubo un inconveniente con respecto al código y se llegó a la conclusión de que fue necesario agregar la función de registro "ADCON1=0b00001111" para activar los pines como digitales de los puertos que comparten pines digitales y análogos , este problema no fue un obstáculo para llevar acabo el laboratorio correctamente, se realizó la configuración adecuada para realizar los ejercicios planteados en la guía.

- Por otro lado es importante resaltar algunas partes que fueron cruciales para el buen funcionamiento del código como incluir la función XTAL_FREQ para el manejo de la función de conteo de tiempo "delay" o la inicialización de la función "PORTDbits…."y su respectivo pin.

- Para finalizar cabe destacar que el software Proteus ayudo a simular prácticamente este laboratorio, mediante el cual se subió la programación hecha en MPLAB con extensión "HEX". El entorno de MPLAB XC8 es un entorno asociado al lenguaje C donde fue más fácil asociarse con sus funciones y parámetros, en ellas se incluyeron condicionales y funciones la cuales ayudaron respectivamente a el buen funcionamiento de las acciones pedidas por la guía. Durante la programación se tuvo que definir la frecuencia de trabajo de nuestro PIC18F4550, en este caso 4MHz, pero es importante definir esto ya que sin ello el parámetro "delay" no funcionaba, , en este caso 4MHz, , así que para todo programa que deba incluirse un "delay" debe definirse correctamente la frecuencia de sincronización del microcontrolador trabajado.

BIBLIOGRAFIAS

- **https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf**
- **https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/puertos-de-entradasalida**