



**Universidad Nacional Autónoma de
México
Facultad de Ingeniería**



Semestre 2023-2

Laboratorio de Microcomputadoras

**Proyecto 1:
Sistema mínimo PIC16F877**

Profesor:

M.I. Ruben Anaya García

Alumnos:

Barreiro Valdez Alejandro

Zepeda Baeza Jessica

Número de cuenta:

317520888

317520747

Grupo Teoría: 1

Fecha de realización: 20 de marzo de 2023

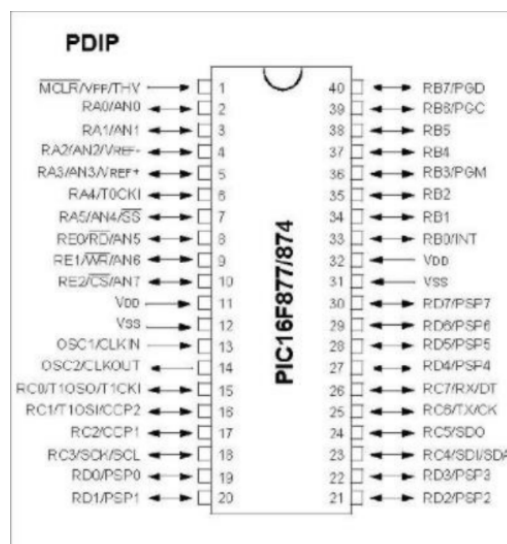
Fecha de entrega: 29 de marzo de 2023

Introducción

En esta práctica se entenderá cómo se utiliza el sistema mínimo del microcontrolador PIC16F877. Se comprenderán cuáles son las partes de este sistema gracias al esquema donde se muestran sus partes. Además, se realizará un programa donde se tengan salidas que sean provenientes de un puerto paralelo. Esta salida será una palabra de tres letras donde se pueda poner en práctica el recorrido de LEDs que se ha realizado en el laboratorio. La gran diferencia con la práctica de laboratorio es que en este proyecto se armará el sistema mínimo y se probará para corroborar su correcto funcionamiento. La salida de este sistema también tendrá que ser armada para generar las diferentes letras de salida utilizando LEDs. Para este proyecto, se armará el sistema mínimo y se generará un código que se cargará en el microcontrolador para comprobar las salidas del microcontrolador.

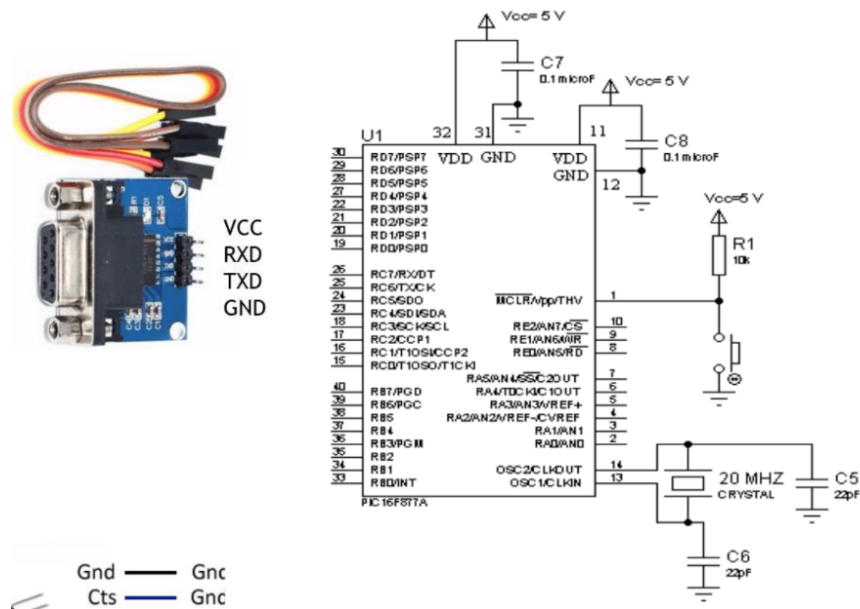
Desarrollo

Para el sistema mínimo se podían elegir distintas maneras de alambrar. La manera que se eligió fue utilizando una interfaz serial. Esta interfaz serial tiene un circuito MAX232 que le permite al microcontrolador comunicarse con una computadora. Lo primero que se hizo para la construcción del sistema mínimo fue cargar el bootloader en el PIC16F877 con ayuda del profesor. Este programa es útil para poder cargar otros programas desde la computadora. Posteriormente se identificaron cada uno de los pines que se tienen en el microcontrolador con el siguiente esquema.

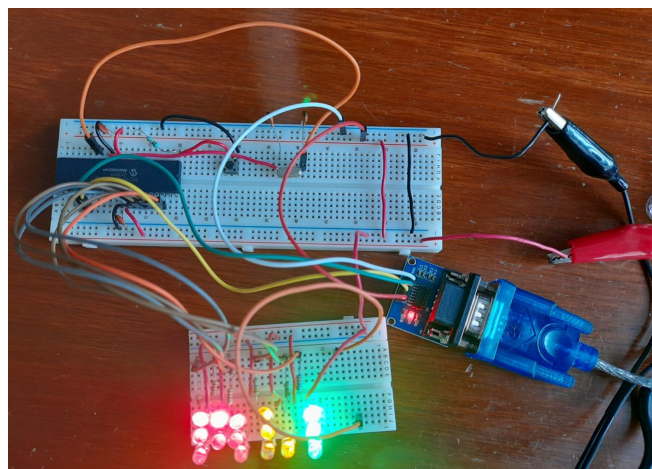


Se requiere identificar los pines que se utilizarán para el sistema mínimo. Además, antes de empezar a alambrar, también se compraron los componentes que se necesitan para el

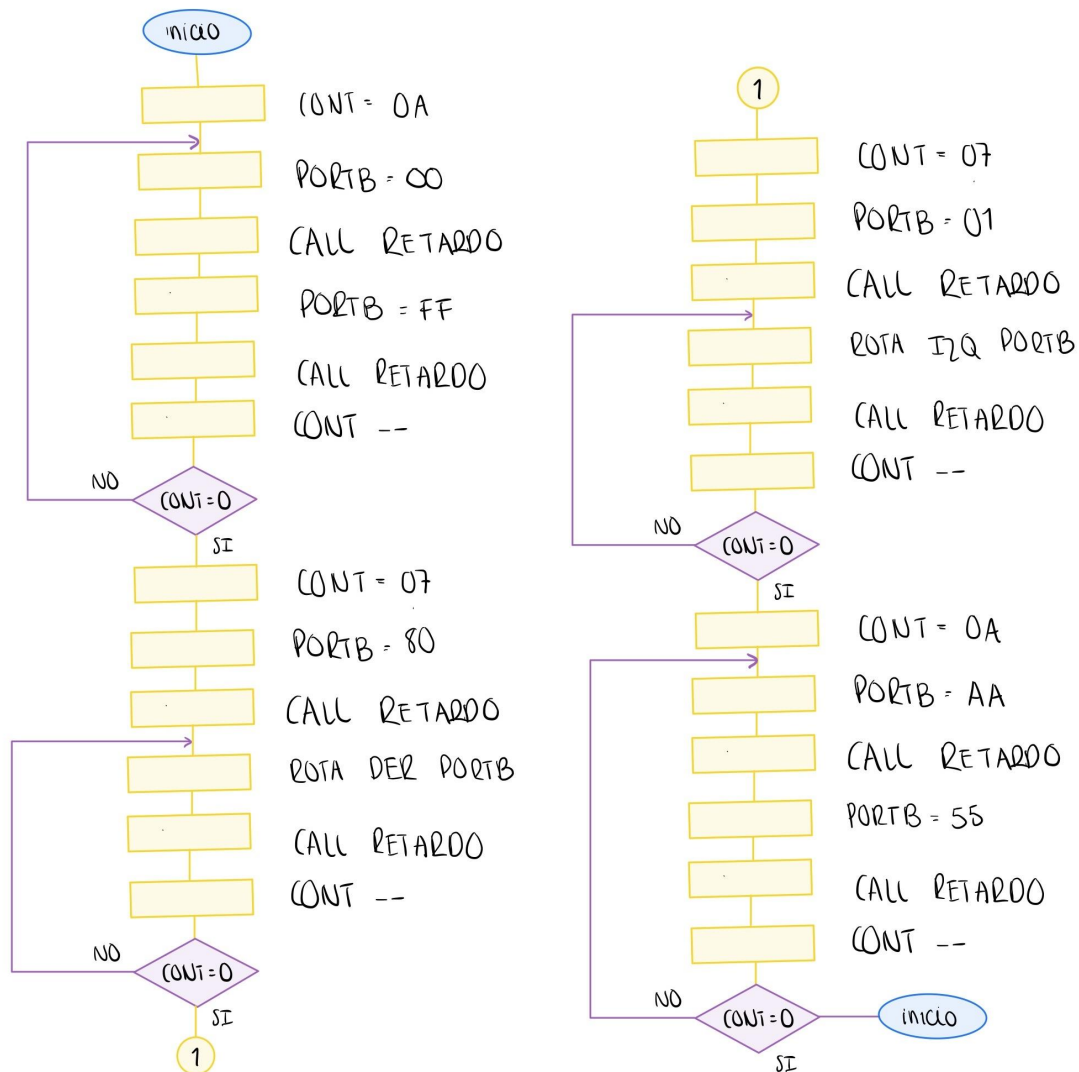
sistema. Estos componentes fueron enlistados por el profesor por lo que lo único que se hizo fue conseguir cada uno. Teniendo todos los componentes que forman parte del sistema mínimo, se pasó a alambrear este utilizando el siguiente diagrama donde se indican los números de pines y a qué componente van conectados.



Por último, después de alambrear el sistema mínimo, se alambrió la salida utilizando LEDs. Para generar las letras se conectaron varios LEDs en paralelo para generar una columna. Acomodando estas columnas se pudo generar la palabra “ALI”. Además, se tuvo que conectar cada uno de los pines del puerto B a la salida de cada LED y cada LED tuvo que conectarse a tierra. Alambreado todo el sistema con salidas y la interfaz serial, el resultado fue el siguiente.



Ejercicio 1



El programa consiste en poner un contador antes de cada uno de los efectos que se generarán con las salidas e iterar cada una de las operaciones y su retardo que se necesitan para los cuatro efectos que se generarán con los LEDs. El primer estado consiste en apagar y prender los LEDs diez veces. Para ello se asigna la salida como 00 y luego como FF. Posteriormente se asigna como 07 el contador para tener siete iteraciones y la salida como 80 para solo prender un LED. La operación que se realiza en el segundo estado es rotar a la derecha la salida y llamar al retardo. Luego, se vuelve a asignar contador como 07 para siete vueltas pero, la salida se asigna a 01 para prender el LED del bit menos significativo. El efecto del estado 3 consiste en rotar a la izquierda la salida y llamar al retardo. Por último, se asigna el contador como 0A para dar diez vueltas en el efecto del último estado. Este cuarto estado consiste en asignar la salida como AA y

después como 55 llamando a un retardo después de cada asignación. Esto generará que se intercale la manera en que se prenden los LEDs y se prenda uno sí y otro no. Después del último estado se vuelve al inicio del programa para volver a ver todos los estados como salidas.

Programas comentados

Ejercicio 1

```
PROCESSOR 16F877
INCLUDE <P16F877.INC>

valor1 equ h'21'      ;Valor para el retardo
valor2 equ h'22'      ;Valor para el retardo
valor3 equ h'23'      ;Valor para el retardo
cte1 equ 20h          ;Constante para el retardo
cte2 equ 50h          ;Constante para el retardo
cte3 equ 60h          ;Constante para el retardo
CONT equ h'25'        ;Contador para cada estado

ORG 0                 ;Vector de reset
GOTO INICIO

ORG 5
INICIO:
    CLRF PORTA        ;Limpia PORTA
    BSF STATUS,RP0    ;Cambia a banco 1
    BCF STATUS,RP1
    CLRF TRISB        ;Configura el puerto B como salida
    BCF STATUS,RP0    ;Cambia al banco 0
    BCF STATUS,C      ;Limpia el acarreo

LOOP:
    MOVLW 0X0A
    MOVWF CONT        ;CONT = A

ED01:
    CLRW
    MOVWF PORTB       ;Se apaga la salida
    CALL RETARDO      ;Se llama al retardo
    MOVLW 0XFF
    MOVWF PORTB       ;Se prende la salida
    CALL RETARDO      ;Se llama al retardo
    DECF CONT         ;Decrementar el contador
    CLRW
    XORWF CONT,W
```

```

BTFSZ STATUS,Z      ;Compara CONT == 0
GOTO EDO1           ;Si no es cero volver a EDO1

    MOVLW 0X07
    MOVWF CONT       ;CONT = 07
    MOVLW 0X80
    MOVWF PORTB      ;PORTB = 80
    CALL RETARDO     ;Llamar retardo

EDO2:               ;Rotar a la derecha
    RRF PORTB        ;Rotar a la derecha el valor del puerto
    CALL RETARDO     ;Llamar al retardo
    DECF CONT        ;Decrementar el contador
    CLRW
    XORWF CONT,W
    BTFSZ STATUS,Z   ;Comparar CONT == 0
    GOTO EDO2        ;Si no es cero volver a EDO2

    MOVLW 0X07
    MOVWF CONT       ;CONT = 07
    MOVLW 0X01
    MOVWF PORTB      ;PORTB = 01
    CALL RETARDO     ;Llamar al retardo

EDO3:               ;Rotar a la izquierda
    RLF PORTB        ;Rotar a la izquierda el valor del puerto
    CALL RETARDO     ;Llamar al retardo
    DECF CONT        ;Decrementar el contador
    CLRW
    XORWF CONT,W
    BTFSZ STATUS,Z   ;Comparar CONT == 0
    GOTO EDO3        ;Si no es cero volver a EDO3

    MOVLW 0X0A
    MOVWF CONT       ;CONT = A

EDO4:               ;Prender uno sí, uno no y alternar

    MOVLW 0XAA
    MOVWF PORTB      ;PORTB = AA
    CALL RETARDO     ;Llamar retardo
    MOVLW 0X55
    MOVWF PORTB      ;PORTB = 55
    CALL RETARDO     ;Llamar retardo
    DECF CONT        ;Decrementar el contador
    CLRW
    XORWF CONT,W
    BTFSZ STATUS,Z   ;Comparar CONT == 0
    GOTO EDO4        ;Si no es cero volver a EDO4
    GOTO LOOP        ;Volver a iniciar el LOOP

RETARDO             ;Subrutina retardo
    MOVLW cte1
    MOVWF valor1     ;valor1 = cte1
tres
    MOVLW cte2
    MOVWF valor2     ;valor2 = cte2
dos
    MOVLW cte3
    MOVWF valor3     ;valor3 = cte3
uno
    DECFSZ valor3    ;decrementar valor 3
    GOTO uno         ;ir a uno
    DECFSZ valor2    ;decrementar valor 2
    GOTO dos         ;ir a dos
    DECFSZ valor1    ;decrementar valor 1
    GOTO tres        ;ir a tres
    RETURN          ;fin subrutina

END

```

Conclusiones y/o comentarios

Zepeda Baeza Jessica:

Para este proyecto se pudo construir el sistema mínimo requerido para el microcontrolador PIC16F877 con todos los elementos necesarios como capacitores, resistencias, push-botton y un cristal de cuarzo de 20MHz. Lo primero que se realizó fue cargar el bootloader al microcontrolador. Esta operación es esencial ya que permite que se le puedan cargar programas al microcontrolador. Después, se realizó todo el alambrado a partir de los diagramas que se proporcionaron. Se tuvo que revisar cada uno de los nodos que se alambraron y los pines a los que se conectaron en el microcontrolador porque si no se realizaba de manera correcta el sistema mínimo no funciona. Luego se realizó el alambrado de lo que sería la salida a partir del puerto paralelo B. Se identificaron los pines del microcontrolador y se conectaron los diodos, en este caso LEDs, con su configuración con una resistencia y conexión a tierra. Por último, se conectó el módulo MAX232 para poder tener comunicación serial con el microcontrolador para cargarle programas. Con este último paso se logró crear el sistema mínimo por lo que la única tarea faltante fue crear el código de prueba. El código consistió en diversas operaciones que se han utilizado en las prácticas como el corrimiento de bits, la comparación de dos números y el uso de etiquetas para controlar el flujo del programa. El programa genera cuatro efectos en las salidas que se realizan varias veces con un retardo y se repiten. Cargando el programa en el microcontrolador se pudo comprobar que el sistema mínimo fue construido de manera correcta y que se logró cumplir el objetivo principal del proyecto.

Barreiro Valdez Alejandro:

En este proyecto se pudo alambra un sistema mínimo para el microcontrolador PIC16F877. Para esto se necesitó entender cómo funcionan los pines en el microcontrolador y entender el diagrama con todos los componentes que se necesitan para el sistema mínimo. Se alambra el sistema con todos los componentes y se utilizó una interfaz serial a partir de un circuito MAX232. Además, se realizó el alambrado para la salida del sistema mínimo para el cual se utilizó el puerto paralelo B. Para estas salidas se utilizaron LEDs que formaron la palabra "ALI". Después de programar el bootloader en el microcontrolador se pudieron cargar programas para este sistema. El programa que se realizó para este proyecto hacía diferentes recorridos en las salidas. Se programó una salida donde se tienen cuatro estados. El primer estado prende y apaga todos los LEDs diez veces; el segundo estado hace un recorrido hacia la izquierda; el tercero hace un recorrido hacia la derecha; y el último prende una hilera sí y otra no y empieza a alternar estas hileras. A partir del

programa y la realización del alambrado se pudo realizar el sistema mínimo utilizando el microcontrolador PIC16F877 y se pudo entender cómo funciona este y cómo se construye.