**LABORATORIO DE MICROCONTROLADORES N°1**

**COMPILACION, SIMULACION Y CREACION DE APLICACIONES BASICAS CON MICROCONTROLADORES**

Alvaro Iván Parra Albarracín 200811766

Luis Fernando Granados Rivera 201110045

Ingeniería Electrónica

Universidad Pedagógica y Tecnológica de Colombia

Alvaro.parra@uptc.edu.co

Luisfernando.granados@uptc.edu.co

***Resumen:*** *En esta práctica se procede a realizar una aplicación, con el fin de conocer las instrucciones del microcontrolador*

PIC16F887, *además de la compilación y simulación de dicha aplicación.*

1. **INTRODUCCIÓN**

En la creación de las aplicaciones, se requiere tener un cierto conocimiento de las instrucciones básicas del microcontrolador, para esto se necesita tener de apoyo el datasheet del microcontrolador PIC16F887, el cual se va a utilizar para la implementación de la aplicación a realizar. Además se necesita conocer la distribución de los pines para así tener un buen funcionamiento del ejercicio.

**Palabras claves**: Instrucciones, PIC16F887, secuencia.

1. **MATERIALES Y EQUIPOS**

* PIC 16F887, PROTEUS, MPLAB X , LED.

1. **PROCEDIMIENTO Y RESULTADO**

Contador/conversor (bin a BCD). Se debe realizar un contador que cuente hasta 255. Cada cambio debe durar mínimo 1 seg. Ésta cuenta se debe visualizar en 12 led. Unidades y decenas serán parte baja y parte alta de PORTB y centenas serán la parte baja de PORTC. Además deberá contar con una opción de carga de datos, de esta manera se podrá cargar el valor de inicio del respectivo conteo por un dip switch a PORTA, este valor será cargado en base hexa. Cada que el conteo finalice se deberá visualizar sobre los 12 led una determinada secuencia de intermitencia diseñada por el estudiante que duré no menos de 1 min. Debe contar con una opción de reinicio.

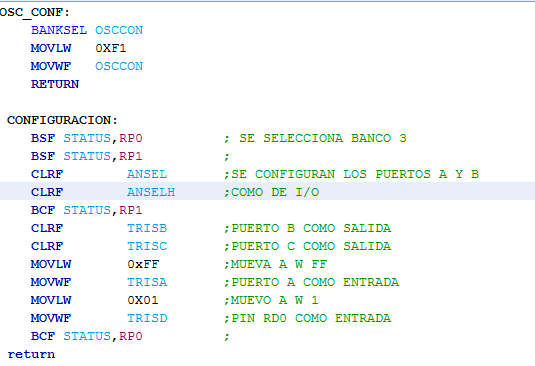


Fig.1 Configuración de los registros que se van a utilizar en el desarrollo de la aplicación.

En la figura 1 se hace la configuración del registro OSCCON, se carga 0xF1 para configurar el micro a 8MHz, como también se configuran los puertos A, B, C como entradas o salidas con sus pines correspondientes, el puerto A se configura como entradas digitales en estos se lee el dato de entrada a convertir, los puertos B y C se configuran como salidas donde se visualizara en 12 leds el dato en código BCD. El pin RD0 se configura como entrada digital, este para leer el estado del mismo y reiniciar el contador o no.

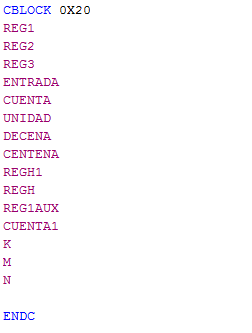
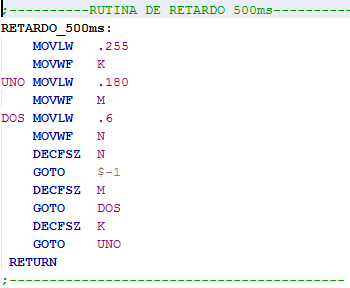


Fig.2 Definición de los registros de propósito general.

En la figura dos se definen las variables que vamos a utilizar durante el desarrollo del trabajo y le asignamos una ubicación a cada una de ellas, partiendo desde la ubicación 0X20



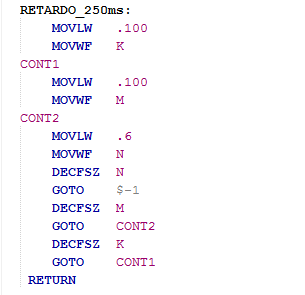


Fig.3 Retardos creados por software de 250 y 500 milisegundos respectivamente.

En la figura 3 creamos las rutinas de retardos de 250 milisegundos para eliminar el efecto de rebote ocasionado por el pulsador del reset y 500 milisegundos, que consiste en ir decrementando un valor que este almacenado dentro de una variable y de esta manera darle el tiempo de espera requerido.

Para calcular este retardo se trabajó la siguiente ecuación:

Fosc=8MHz.

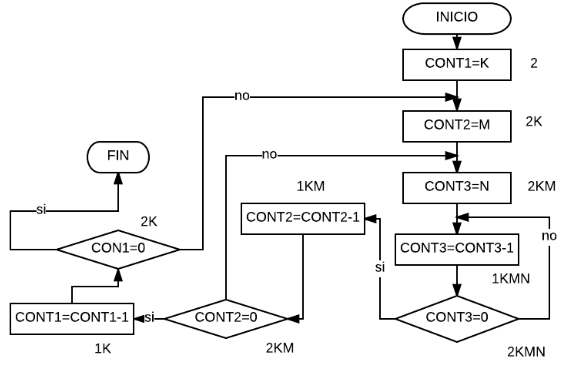


Fig.4 Diagrama de flujo de la rutina de retardo.

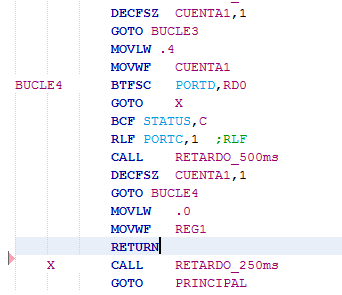
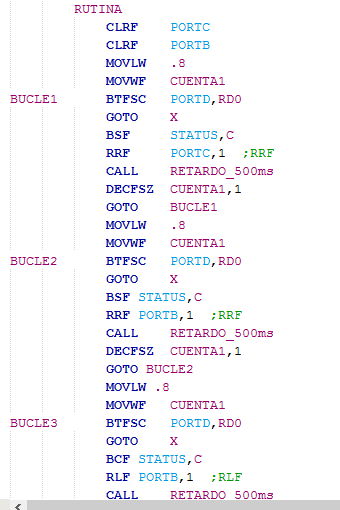
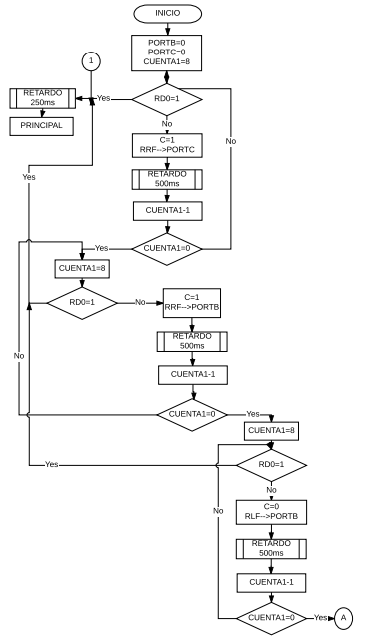


Fig.5 Rutina de la secuencia de luces que será visualizada en los leds.



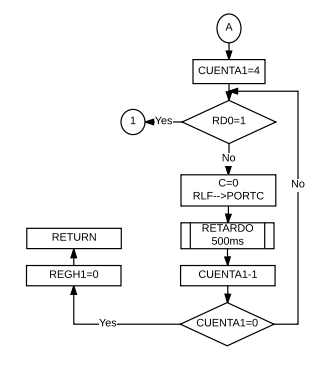


Fig.6 Diagrama de bloques para la secuencia de luces que será visualizada en los leds.

En la figura 5 se crea el código que representa la secuencia de luces que será visualizada en los leds; la cual consiste en un desplazamiento del carry de derecha a izquierda para encender uno a uno los bombillos leds y posteriormente se hace el mismo procedimiento de izquierda a derecha para des energizarlos.

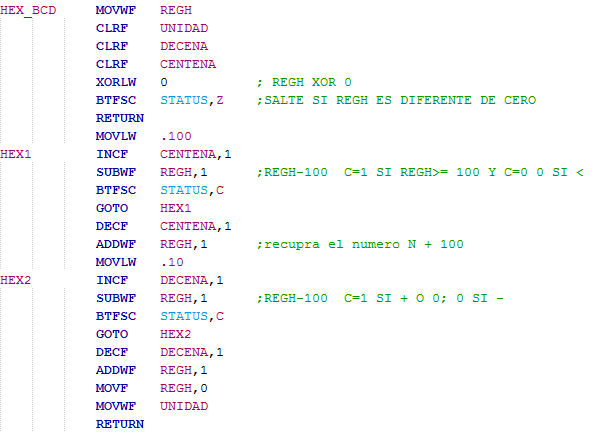


Fig.7 Rutina para pasar de hexadecimal a formato a BCD.

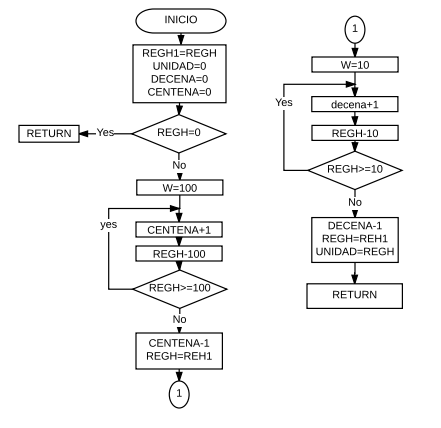


Fig.8Diagrama de bloques para de hexadecimal a formato a BCD.

En la figura 7 realizamos la rutina para pasar un numero de formato hexadecimal a formato BCD y consiste en descomponer el numero en unidades decenas y centenas para luego aplicar el método llamado división por restas sucesivas. En pocas palabras se toma el dato que el usuario ingrese por el puerto A, ese dato lo almacenamos en una variable ,teniendo el dato almacenado en la variable, le aplico una resta con el valor de cien con el fin de realizarlas las veces que sea necesario antes de que el número de la variable sea menor que cien, cuando se cumple esta condición ya tengo el valor de las centenas, enseguida tomo el valor de la variable que me iba cambiando a medida que realizaba una resta, ahora realizo de nuevo la resta pero esta vez por 10 y voy contando las veces que realizo la resta hasta que el valor de la variable sea menor que 10, y ahí tengo las decenas especificadas del número, además ya el ultimo valor almacenado en la variable número es el correspondiente a las unidades y este procedimiento se vuelve a repetir en un ciclo infinito dentro del micro controlador.

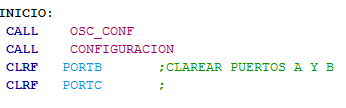


Fig.9 Llamado de las ruinas de configuración.

En la figura siete se hacen el llamado de las configuraciones previamente hechas de la configuración del reloj y de los registros. Adicionalmente se usa la instrucción clrf para asegurarse que los puertos Ay B estén en cero.

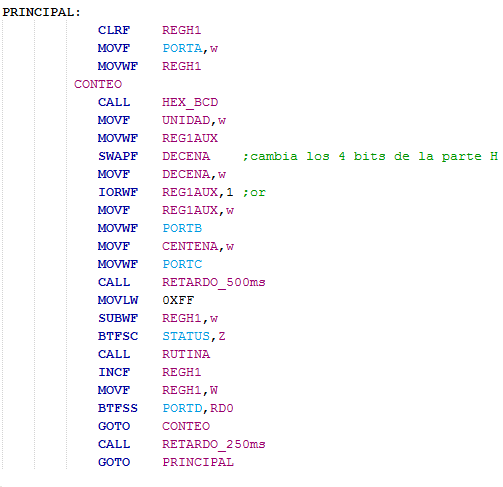


Fig.10 programa principal del desarrollo de la aplicación.

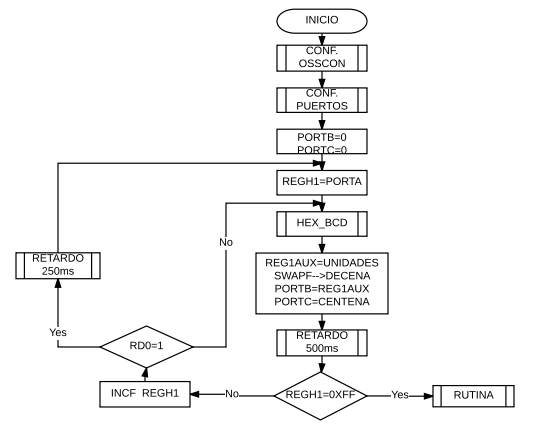


Fig.11Diagrama de bloques del programa principal.

Para el desarrollo de la práctica se lee el dato de la entrada en el puerto A y se guarda en un registro auxiliar, ahora se hace la conversión del dato retornando el valor de unidades en un registro, decenas y centenas en otro registro similar llamado centenas, a este registro le hacemos un cambio de la parte alta a la parte baja y la parte baja a la parte alta, ahora se procede a mostrar los datos en los puertos de salida, se hace un retardo para poder visualizar el dato en la salida, ahora se pregunta el dato que se mostro es igual a 255, si lo es se procede a mostrar la rutina diseñada, si no se incrementa el dato mostrado y preguntamos si el pin RD0 está activo si está activo reiniciamos el sistema al dato de entrada, si no se activa comenzamos a realizar la conversión.

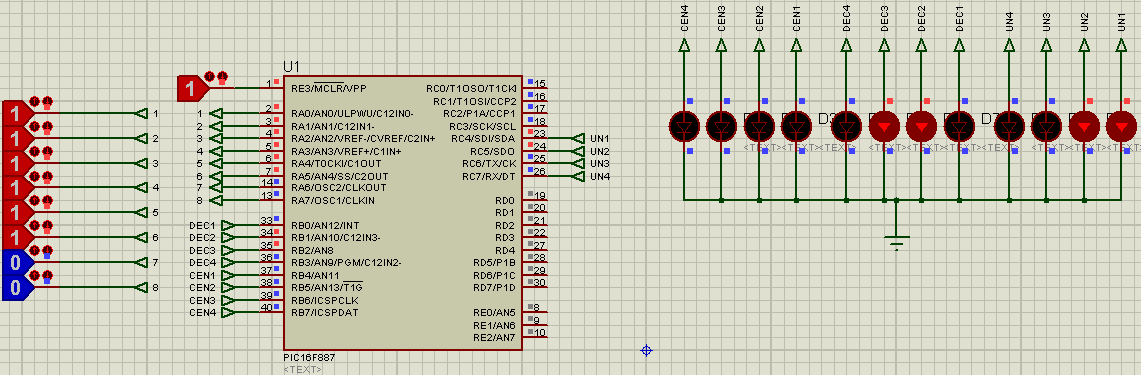


Fig.12 Circuito esquemático para la simulación de la aplicación.

En la figura 12 observamos el circuito esquemático de la aplicación montada en proteus donde se puede observar la conversión y la secuencia .Además observamos su correcto funcionamiento en la etapa de simulación.

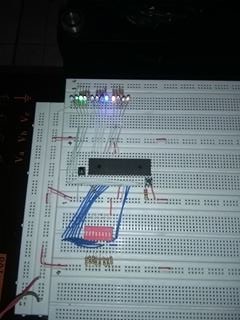


Fig.13 Implementación de la aplicación sobre la protoboard.

En la Figura 13 finalmente se observa la implementación en físico de la aplicación que anteriormente fue diseñada y simulada. En esta etapa del proceso ya se puede corroborar el funcionamiento tanto de la conversión de los números a formato BCD; como de la secuencia

diseñada para que se visualice en los bombillos led.

1. **CONCLUSIONES**

* La implementación de un código requiere un análisis previo y detallado de la lógica a implementar, ya que esta difiera de la dificultad en la implementación del código
* El conocimiento detallado del set de instrucciones del micro controlador hace que la implementación del código resulte aún más eficientey rapida.

.

**BIBLIOGRAFIA**

B.P. Lathi; Modern digital and analog communications systems 3rd ed

F.G. Stremler; Introduccion a los sistemas de comunicacion.

Data sheet del PIC16F887