LABORATORIO DE MICROCONTROLADORES

COMPILACIÓN, SIMULACIÓN Y CREACIÓN DE APLICACIONES BÁSICAS CON MICROCOCNTROLADORES

Barrera. Yesica, Mora. Jawer
Yesica.barrera@uptc.edu.co, Jawer.mora@uptc.edu.co
Universidad pedagógica y tecnológica de Colombia
Seccional Sogamoso – Ingeniería Electrónica

RESUMEN: En esta práctica se desarrolló una aplicación utilizando el microcontrolador PIC16F887 capaz de realizar conversión de números binarios a BCD y su posterior visualización mediante leds conectados a sus puertos de entrada salida. Esto con el fin de utilizar algunas instrucciones claves para aplicaciones complejas en assembler al igual que el uso de subrutinas.

Palabras clave: M*icrocontrolador, Conversión binaria a BCD, Set de instrucciones, Retardos.*

1. INTRODUCCIÓN

Para el desarrollo de aplicaciones con microcontroladores son de vital importancia las herramientas que permiten llevar a cabo esta labor, el MPLAB y el PROTEUS, son solo una pequeña muestra de las aplicaciones desarrolladas con este fin. Con el uso de las mismas se puede realizar las operaciones de creación, depuración y terminado final de sistemas de control completo realizadas alrededor del uso de estos versátiles dispositivos.

Para entender y utilizar de manera clara el set de instrucciones, las rutinas y la buena elaboración de los diagramas de flujo para los diferentes aplicaciones, en el presente informe se realizan diferentes programas básicos donde se han puesto en práctica los aspectos anteriores, de igual forma el manejo del software de simulación PROTEUS para la parte de revisar los resultados.

2. MARCO CONCEPTUAL

MPLAB® X IDE: Es un software que es usado para desarrollar aplicaciones para microcontroladores de microchip y control de señales digitales, estas herramientas de desarrollado comúnmente se les conoce como IDE o entorno de desarrollo integrado[1].

DIAGRAMAS DE FLUJO: La importancia de un diagrama de flujo es que hace al software más entendible, esto ayudara a visualizar de distintas formas la complejidad del problema y unas posibles soluciones, sin embargo al ofrecer este tipo de ayuda no hace comprensible un software para muchos, esto depende del alto nivel de complejidad del código como lo cita el autor[2].

RUTINAS: Son uno de los recursos muy valiosos cuando se trabaja en programación; ellas permiten que los programas sean más simples, debido a que el programa principal se disminuye cuando algunas tareas se escriben en forma de programas pequeños e independientes.

3. CONFIGURACION DE HARDWARE

3.1 PROGRAMA PRINCIPAL

En esta aplicación el microcontrolador empieza realizando el conteo desde 0 hasta 255 cada segundo cuanto en el pin RE0 del puerto E hay un 0 lógico. El valor del conteo se almacena en un registro de propósito general llamado **NUMERO_AUX** y se convierte de código **BINARIO** a **BCD** mediante una subrutina en el programa llamada **BINARIO_BCD,** esta rutina obedece al algoritmo mostrado en el diagrama de flujo representado en el anexo 02 el cual consiste en tomar el valor del conteo y decrementarlo de 10 en 10 para conocer la cantidad de unidades, decenas y centenas que lo contienen**.**

Una vez conociendo las unidades, decenas y centenas del conteo, se muestran mediante 12 leds ubicados de la siguiente manera:

CENTENAS: 4 leds en la parte baja del PUERTO C
DECENAS: 4 leds en la parte alta del PUERTO B
UNIDADES: 4 leds en la parte baja del PUERTO B

Para lograr ubicar los valores de unidades y decenas en el puerto b, se utiliza la instrucción **SWAPF** para que se intercambien los valores de la parte alta y baja del registro donde se almacenan las decenas y mediante la operación **XOR** entre decenas y unidades queden los dos valores en una sola variable. Al final esta variable se mueve al puerto B.

La aplicación también debe tener la capacidad de que una vez el contador llegue a 255, muestre una secuencia en los leds con cambio entre estado de un segundo.

Para lograr la secuencia en el microcontrolador se almacenan los valores correspondientes a esta en tablas guardadas en posiciones diferentes en la memoria de programa y cada valor se va llamando mediante la instrucción **RETLW** y el registro de trabajo. Este valor se mueve a los diferentes puertos utilizados en la aplicación.

Si en el pin RE0 está en 1 lógico, el contador se detiene y toma el número binario que detecte el puerto A del microcontrolador para conversión y se visualiza de igual manera que lo anteriormente descrito.

El dato que el microcontrolador recibe por el puerto A, se configura mediante dip swich y resistencias de pull-down.

3.2 Configuración de Hardware

3.2.1. Configuración de puertos

Para la aplicación se configura el puerto A como entrada digital usando los registros **TRISA** configurando todos sus bits con 1 lógico y **ANSEL** en cero lógico. El puerto E de igual manera se configura como entrada digital mediante el **TRISE** en 1 lógico.

Para que el microcontrolador muestre los datos deseados, se configuran los puertos B y C como salidas digitales clareando el **TRISB Y TRISC**, al igual que el **ANSELH.**

3.2.3 Configuración del oscilador

Para la configuración del oscilador del microcontrolador se usa el registro **OSCCON,** se elige el oscilador interno del microcontrolador con frecuencia de 8 MHz.

3.3 Instrucciones Utilizadas

SWAPF: Lo que realiza esta instrucción es que los bits superiores e inferiores del registro F se intercambian. Si el destino es 0, el resultado se pone en el registro de trabajo. Si el destino es 1, el resultado es puesto en el registro F.

SUBWF: que significa substracción entre dos registros.

RETLW: Retorna W con un literal

ADDWF: Suma entre dos registros.

3.5. Resultados prácticos:

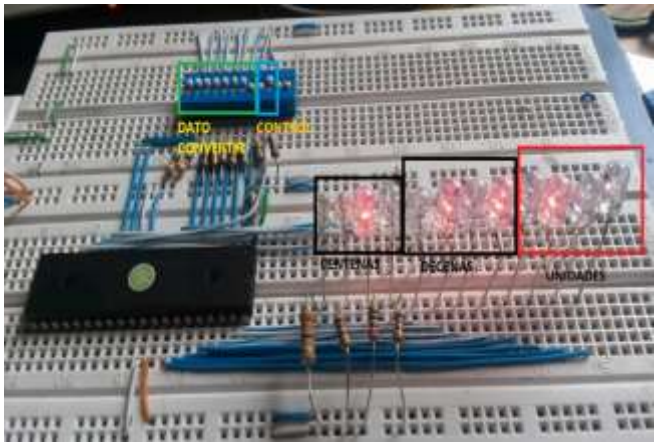


Figura 1. Fotografía funcionamiento conversor BINARIO A BCD

En la anterior imagen se detalla el funcionamiento de la aplicación, en cada recuadro se muestran los leds que indican las unidades, decenas y centenas del numero convertido, para este caso es el 254 en binario (parte superior) y BCD a la derecha del montaje.

3.6. Resultados en simulación:

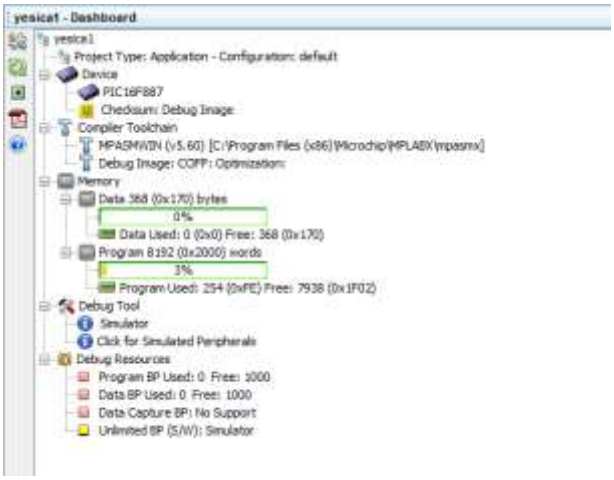


Figura 2. Calculo de memoria de programa usada por la aplicación según el software MPLAB de MICROSHIP.

En la figura anterior se visualiza la cantidad de memoria de programa consumida por la aplicación, como se puede observar es realmente baja con respecto a la cantidad con la que cuenta el dispositivo.

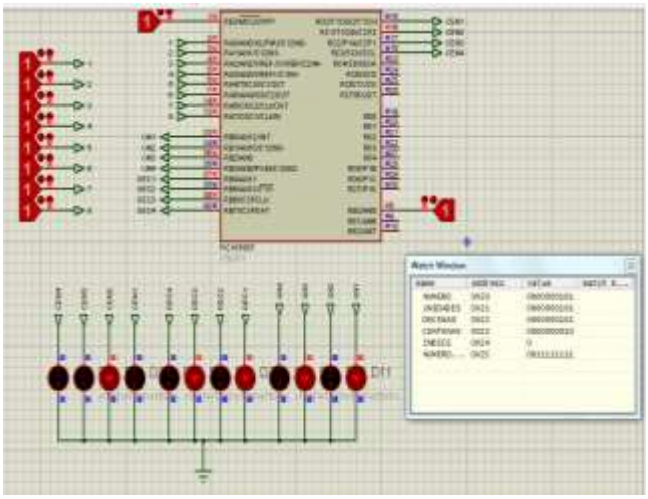


Figura 2. Simulación obtenida del Simulador Isis de Proteus para la conversión

En la figura 03, se ve el funcionamiento del conversor binario-bcd, para este ejemplo se usó el número 0xFF y se muestra la salida BCD mediante los leds.

CONCLUSIONES

- El uso de rutinas facilita la programación y el orden en el código, sin embargo en ocasiones es necesario paginar algunas de ellas para evitar el desborde en la ejecución de las tareas.

- En la programación y uso de SFR se debe tener muy en cuenta los bancos de la memoria RAM y su direccionamiento mediante el RP0 y RP1 del registro STATUS.

- Según el cálculo del software sobre la memoria de programa consumida por la aplicación, se puede inferir que este dispositivo se puede usar en aplicaciones mucho más complejas y versátiles donde se usen todos sus recursos.

- En el desarrollo de la aplicación se incluye la librería del microcontrolador usado en una extensión .inc debido a que hay que definir la posición de memoria en la que se ubica cada registro sfr, también los registros definidos por el hardware mismo del microcontrolador y los cuales no son modificables por software.

REFERENCIAS

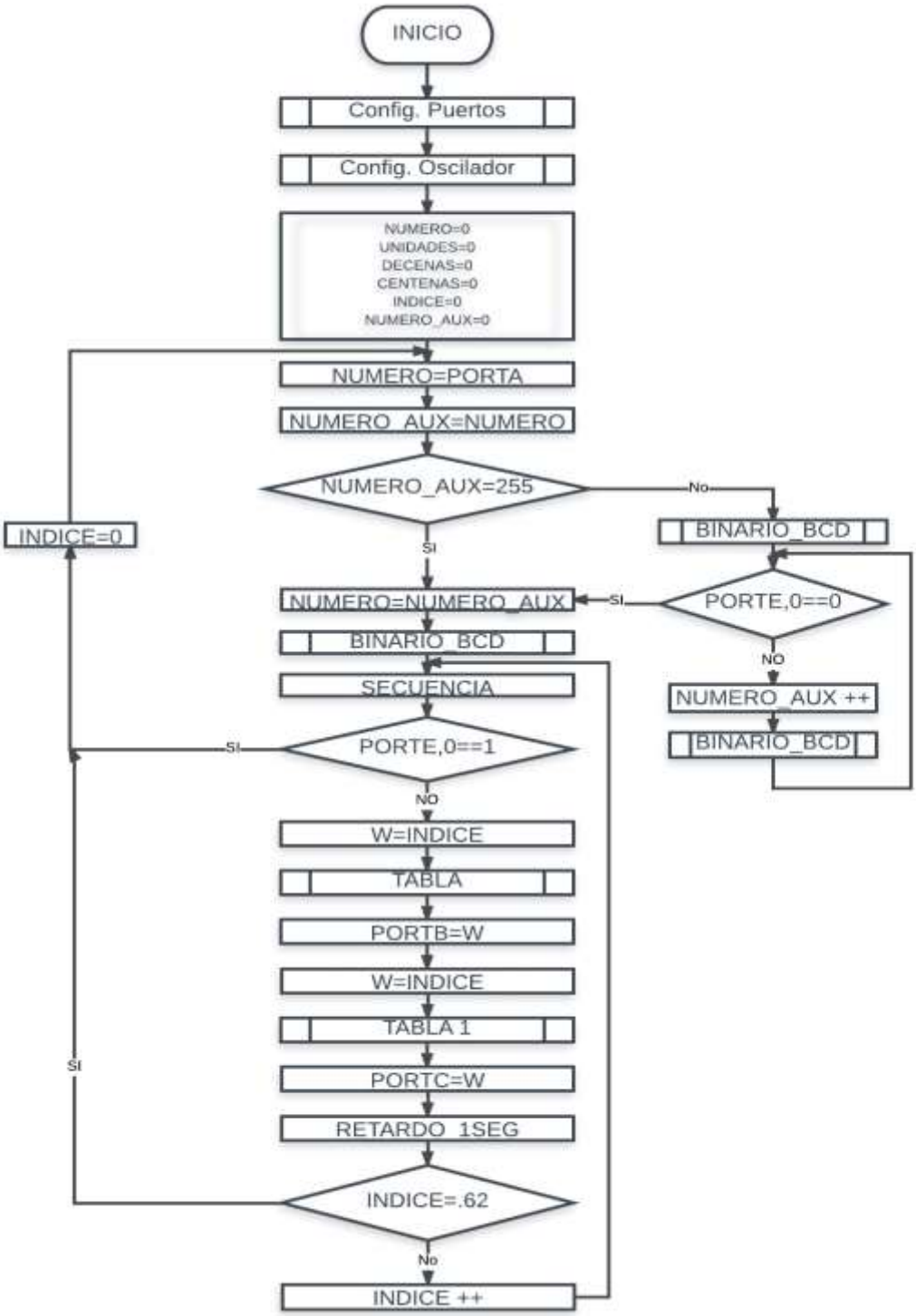
[1] "MPLINK ™ Object Linker , MPLIB ™ Object Librarian User ' s Guide," 2005.

[2] D. a. Kosower and J. J. Lopez-Villarejo, "Flowgen: Flowchart-based documentation for C<mml:math altimg='si1.gif' display='inline' overflow='scroll' xmlns:xocs='http://www.elsevier.com/xml/xocs/dtd' xmlns:xs='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' x," *Comput. Phys. Commun.*, 2015.

[3] E. T. Range and O. Start-up, "Pic16f882/883/884/886/887," 2012.

ANEXOS

ANEXO 1. DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL DE LA APLICACIÓN



ANEXO 2. DIAGRAMA DE FLUJO CONVERSIÓN BINARIA A BCD