

Práctica 2: Ensamblador, Interrupciones Temporizadores

Alberto Acosta López, Francisco Javier López Franco
Luis Fernando Peña Flores

Resumen—En esta práctica del laboratorio de Principios de Mecatrónica se utilizó el lenguaje ensamblador y el lenguaje C para poder programar operaciones básicas en un microcontrolador. Asimismo, se contrastó y se comparó el código en lenguaje ensamblador con el código compilado de un lenguaje de alto nivel, en este caso, C. También se implementó programas básicos que utilizaron interrupciones y contadores.

1. INTRODUCCIÓN

Se sabe que en un microcontrolador se puede programar en un lenguaje de alto nivel, pero lo que no es muy conocido es qué hay detrás de esas letras que hacen correr un programa. El lenguaje ensamblador es lo más puro que un humano puede entender en términos de programación, y por eso el conocerlo nos ayuda a realizar software de mejor calidad.

2. MARCO TEÓRICO

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM/ROM), puertos de entrada, salida y periféricos. Todas estas partes están interconectadas dentro del microcontrolador y, en conjunto, forman lo que se le conoce como microcomputadora, como se puede ver en la figura 1. [1]

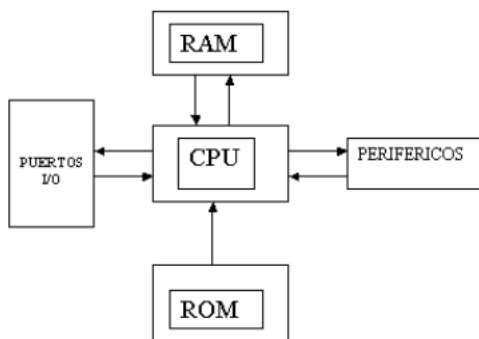


Figura 1. Esquema de un microcontrolador

Toda microcomputadora requiere de un programa para que realice una función específica, la cual se almacena normalmente en la memoria ROM.

El propósito fundamental de los microcontroladores es el de leer y ejecutar los programas que el usuario escribe, por lo que se utiliza el lenguaje ensamblador. Dicho lenguaje es una lista con un limitado número de instrucciones a las cuales puede responder un microcontrolador. Las instrucciones en lenguaje ensamblador son fáciles de entender y permiten operar directamente con los registros de memoria, así como con las instrucciones intrínsecas (alojadas en la memoria ROM) del microcontrolador. Este lenguaje permite un uso eficiente de la memoria y minimiza el tiempo de ejecución de un programa. [2]

Dado que las instrucciones fueron escritas en un lenguaje de alto nivel, éstas deben ser transformadas en código máquina, por lo cual existe el compilador, que traduce y transforma el programa, tal como se observa en la figura 2.



Figura 2. Esquema de un compilador

[3]

Un timer es aquel que únicamente cuenta ciclos de reloj. El reloj del timer puede ser igual al reloj del sistema o puede ser alentado por un preescalador. Cuando se utiliza un preescalador, se pueden obtener mayores valores por tiempo, mientras que la precisión baja. Un timer de 8 bits presenta la siguiente estructura:

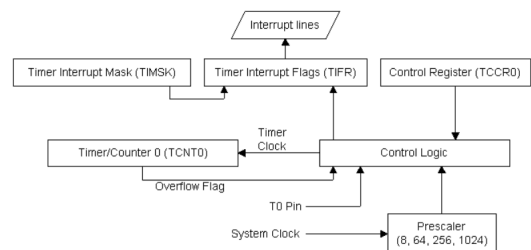


Figura 3. Esquema de un Timer de 8 bits

[4]

3. DESARROLLO

La presente práctica se dividió y se desarrolló en tres secciones. La primera consistió en utilizar el lenguaje ensamblador y el lenguaje C para implementar algoritmo matemáticos sencillos, a saber, el cálculo del determinante de una ecuación de segundo grado y el cálculo del promedio de N números.

Posteriormente se utilizó ambos lenguajes para trabajar con medios físicos, al prender un led utilizando el microcontrolador como herramienta de apoyo. Finalmente, se comparó el código en lenguaje ensamblador con el código compilado en lenguaje C y se hicieron las comparaciones pertinentes.

En seguida, y para la segunda sección de la práctica, se procedió a trabajar con interrupciones. Para dicho fin se hicieron parpadear leds a una frecuencia de 2Hz utilizando interrupciones entre cada ciclo de prendido y de apagado de los leds. Para finalizar la sección se agregaron fotodiodos que determinaban, dependiendo del estado del fotodiodo receptor, si un contador debía o no incrementar su valor.

Para terminar la práctica se configuraron temporizadores en dos modos, a saber, el normal y el CTC. En ambos casos, y a modo de prueba, se hicieron parpadear leds a 2Hz y 4Hz, respectivamente. Y para terminar, se utilizaron temporizadores en su modo normal para construir un circuito para que emulara el funcionamiento común de un semáforo. El cuál se comportaba de tal que:

- El color rojo durara 10 segundos.
- El color verde durara 15 segundos.
- En los últimos tres segundos de transición entre el color verde y el color rojo se encendiera el color amarillo.
- Al apagarse el color rojo se enciende el color verde.

4. RESULTADOS

Para la primera sección de la practica se tuvieron muchas consideraciones adicionales que las estudiadas en la clase teórica. En el caso del determinante las operaciones de multiplicación son, en la mayoría de los casos, tratados en dos registros en lugar de uno. Esto genera el problema de hacer demás operaciones tomando en cuenta un acarreo intermedio. Para ello se utilizó la operación ADC, que realiza la suma tomando en cuenta el valor de la bandera de acarreo.

En el caso del promedio de N números se utilizó un número fijo de variables de entrada. Es decir, no se puede dar un arreglo con N elementos para el cálculo de su promedio. Dicho problema fue causado por la dificultad de obtener las direcciones de memoria del arreglo una por una. Descartando lo antes mencionado, el resto de la sección se desarrolló sin mayores inconvenientes.

Durante la segunda sección se implementaron las interrupciones de forma satisfactoria. Asimismo, se identificó que al utilizar el botón como un activador del

contador no es fácil controlar el ritmo de aumento en la cuenta. También es importante mencionar que al utilizar los fotodiodos fue difícil verificar si estos funcionaban correctamente, debido a la invisibilidad de luz infrarroja al ojo humano.

Finalmente, en la tercera sección de la práctica se alambrió y se programó el circuito del semáforo sin mayores inconvenientes. El único detalle a mencionar fue el cálculo matemático para hacer que el temporizador durara un segundo. Una vez hecho esto, la implementación del controlador fue relativamente sencilla.

5. CONCLUSIONES

5.1. Alberto Acosta

Al finalizar esta práctica aprendí el uso del ensamblador, así como su funcionamiento con las operaciones que se realizaron, tales como el promedio, multiplicación y división. Es un modo más detallado de escribir lo que se quiere que se haga y, no se puede obtener el resultado deseado con la misma sencillez que en un lenguaje de programación. Definitivamente prefiero utilizar lenguajes de programación y sus compiladores, ya que me permiten hacer más cosas con una menor atención a los detalles, además de que es un modo más sencillo.

5.2. Francisco López

Realizando la práctica uno se da cuenta de las ventajas que se tienen usando el lenguaje ensamblador. Se tiene un control total de los componentes de la computadora, lo que permite realizar cosas que en un lenguaje común difícilmente se podría. Aún así, trabajar en él puede llegar a ser muy tedioso, por lo que no podemos despreciar la gran ventaja que nos dan los lenguajes de alto nivel, ya que sin ellos no se podrían realizar programas tan complejos como los que hoy en día usamos. Entender el código ensamblador que hay detrás del código al que estamos acostumbrados es la clave para realizar programas de alto desempeño que no desperdicien recursos algunos.

5.3. Fernando Peña

Una vez terminada la práctica aprendí lo que es, de forma física, un microcontrolador y como programar funciones básicas dentro del mismo. Asimismo, tuve la oportunidad de tener un primer acercamiento al lenguaje ensamblador y poder comparar y contrastar su funcionamiento operacional con un lenguaje de alto nivel. Finalmente, pude poner en práctica los conocimientos de la clase de teoría respecto a instrucciones básicas del lenguaje ensamblador, así como configurar de forma básica tanto las interrupciones como los temporizadores.

6. ROL O PAPEL

1. **Alberto:** Revisar puntos a realizar de la práctica, armar el circuito y checar errores presentados.
2. **Francisco:** Programar el código en Arduino y checar errores presentados.
3. **Fernando:** Armar el circuito y checar errores presentados.

7. FUENTES CONSULTADAS

REFERENCIAS

- [1] EcuRed. Microcontrolador. [Online]. Available: <https://www.ecured.cu/Microcontrolador>
- [2] U. de Oviedo. Ensamblador. [Online]. Available: <https://www.unioviedo.es/ate/alberto/TEMA3-Ensamblador.pdf>
- [3] E. Estudio. ¿qué es un microcontrolador? [Online]. Available: <https://www.electronicaestudio.com/que-es-un-microcontrolador/>
- [4] AvrBeginners. Timers. [Online]. Available: <http://www.avrbeginners.net/architecture/timers/timers.html><https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>