

Práctica 2. Lenguaje Ensamblador, Interrupciones y Temporizadores

1. Objetivos

- a. Analizar los casos en que una aplicación amerite la implementación de código en lenguaje ensamblador, atendiendo a los conceptos de arquitectura de computadoras y sistemas en tiempo real.
- b. Hacer una comparativa entre los códigos de lenguaje ensamblador y los lenguajes de alto nivel para aplicaciones análogas.
- c. Conocer la configuración básica de temporizadores y del sistema de interrupciones propio de los microcontroladores.

2. Antecedentes

- a. El lenguaje ensamblador, también conocido como lenguaje máquina, es un sistema de codificación de instrucciones de bajo nivel que trabaja directamente a nivel de los registros y localidades de memoria de un microprocesador o microcontrolador. Este nivel de lenguaje es capaz de garantizar tiempos de ejecución de muy alta exactitud pues cada instrucción mantiene una cantidad determinada de ciclos de reloj para ejecutarse. Los sistemas en tiempo real son aquellos en los que se garantiza la ocurrencia de eventos en momentos determinados, por lo que es crucial tener una buena planificación de tareas.
- b. El servicio de interrupciones es una utilidad de los sistemas en tiempo real en la que se permite interrumpir momentáneamente el flujo principal del programa para atender a tareas ya sea de prioridad más alta o de ejecución periódica. Los dos tipos principales de interrupciones son aquellos basados en temporizadores y las controladas por estímulos externos.
- c. Los temporizadores son registros internos del microcontrolador o microprocesador que llevan un conteo estable de los ciclos de reloj que han transcurrido a partir de un evento determinado. Una vez que la cuenta ha sido completada o el registro se ha desbordado se lanza una interrupción, que previa configuración, puede ser utilizada para iniciar procesos alternos de periodicidad constante.

3. Aspectos técnicos

- a. Periféricos del microcontrolador AT MEGA 2560
 - 2 temporizadores/contadores de 8 bits con preescalador y comparador.
 - 4 temporizadores/contadores de 16 bits con preescalador, comparador y captura.
 - Contador de tiempo real con oscilador dedicado.
 - 4 canales PWM de 8 bits.
 - 12 canales PWM con resolución programmable de 2 a 16 bits.
 - 16 canales de conversión analógico-digital (ADC) de 10 bits.
 - 4 puertos seriales USART programables.
 - Interfaz serial SPI Maestro/Esclavo.
 - Temporizador tipo Watchdog programmable con oscilador dedicado.
 - Interrupciones y Wake-up activados por cambio en pin.

4. Procedimientos

ENSAMBLADOR



- Analizar el conjunto reducido de instrucciones del microcontrolador ATMEGA y reportar brevemente el tipo (categoría) de instrucciones que se pueden identificar.
- Realizar un programa en Arduino que haga parpadear 4 diodos LED a diferentes frecuencias en 0.25, 0.5, 1 y 2 Hz respectivamente.
- Escribir un código en lenguaje ensamblador para la misma tarea del punto anterior y reportar una comparativa entre ambos códigos. El código ensamblador debe estar debidamente incrustado dentro de la plataforma de Arduino para facilitar su escritura en el dispositivo.
- Generar una rutina de tiempo de 1 segundo mediante instrucciones de ensamblador aplicadas directamente a registros y localidades de memoria.
- Mediante dos juegos de diodos LED rojos, amarillos y verdes, simular en la protoboard los semáforos de un cruce peatonal en el que cada luz permanece encendida durante 6, 1 y 5 segundos respectivamente. El código debe ser implementado en lenguaje ensamblador, haciendo llamados a una única rutina de tiempo. Reporte sus conclusiones.

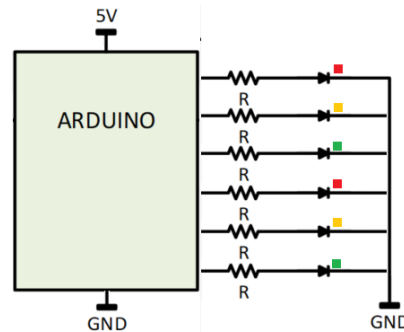


Figura 1. Semáforo digital.

INTERRUPCIONES



- Crear un programa en Arduino que haga parpadear un LED a una frecuencia de 1 Hz.
- Implementar una rutina que incremente la variable de un contador cada vez que se hace una pulsación en un push-button externo. Esta tarea no debe interferir con los tiempos de oscilación del primer LED.
- Sustituir la función del push-button con un interruptor óptico.
- Habilitar y configurar una interrupción externa para ejecutar el código correspondiente a la pulsación del push-button.

A continuación se muestra el código de ejemplo para activar la interrupción 1:

```
void setup()
{
  cli();
  DDRD &= ~(1 << DDR1);
  PORTD |= (1 << PORTD1);
  EICRA |= (1 << ISC10);
  EIMSK |= (1 << INT1);
  sei();
}
ISR(INT1_vect)
{
  ... // Código a ejecutar al activarse la interrupción }
```

TEMPORIZADOR



- j. Habilitar y configurar un temporizador para tener un periodo de 1 segundo.
- k. Deshabilitar interrupciones
- l. Configurar registros de control
- m. Establecer el preescalador
- n. Establecer el valor inicial del contador mediante la fórmula:

$$2^{16} - \frac{16MHz}{pre \cdot f(Hz)}$$

- o. Activar la máscara de interrupción correspondiente
- p. Habilitar interrupciones
- q. A continuación se muestra el código para activar el temporizador 1 en modo normal:

```
void setup()
{
  cli();
  TCCR1B = 0; TCCR1A = 0;
  TCCR1B |= (1 << CS12);
  TCNT1 = 3036;
  TIMSK1 |= (1 << TOIE1);
  sei();
}
ISR(TIMER1_OVF_vect)
{
  ... // Código a ejecutar al activarse la interrupción }
```

- r. Realizar la simulación del semáforo atendiendo a las interrupciones del temporizador en lugar de llamar a las rutinas de tiempo.
- s. Añadir todos los códigos generados al repositorio en la cuenta GIT del equipo.

5. Estructura del reporte

El reporte de la práctica de laboratorio debe incluir bajo el formato a dos columnas de conferencia IEEE (https://www.ieee.org/conferences_events/conferences/publishing/templates.html) lo siguiente:

Resumen (abstract), Introducción, Marco Teórico, Desarrollo, Resultados, Conclusiones y Referencias

Incluyendo los puntos resaltados durante los procedimientos de la práctica.

6. Referencias

- Arduino reference
<https://www.arduino.cc/reference/en/>
- ATMEGA 2560
http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- ARDUINO INTERRUPTS
<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>