



1. Seguridad en la Ejecución.

	Peligro o fuente de energía	Riesgo asociado
1	Manejo de Corriente Alterna	Electrochoque
2	Manejo de corriente Continua	Daño al equipo

2. Objetivos de aprendizaje.

- El alumno programará los Puertos de Entrada/Salida del procesador ARM M4 tanto en lenguaje ensamblador como en lenguaje C, para emplearlos como puertos digitales de Propósito General, e implementará la conexión de hardware externo al microcontrolador tomando en cuenta los parámetros eléctricos nominales de las terminales.

3. Material y equipo.

- Hoja de datos del microcontrolador TM4C1294NCPDT.
- Guía mínima, Worksheet Impresa.
- Sistema de desarrollo.
- Cable BNC y Osciloscopio.
- Leds, Switches N.O. y resistencias de 330Ohms a 1kOhm, alambre AWG22.

4. Actividad previa.

- ¿Para qué sirve la directiva `#define`?
- ¿Qué es una variable global y una local?
- ¿Cómo se hace referencia a una dirección de un registro en lenguaje C empleando la directiva `#define`?
- Describa el modo de funcionamiento de bits direccionables (o direccionamiento de bits específico) en un puerto GPIO. En los programas, `main.c` y `main2.c` complete los valores de configuración faltantes (marcados como (a), (b), etc. . .) considerando para el procesador empleado TM4C1294. Tome en cuenta los operadores lógicos en C `:=`, `|`, `=`, `&` `=`.
- Escriba el código para implementar un retardo de aproximadamente 1 segundo y agregarlo entre cada cambio de estado del LED(2).

5. Desarrollo

Implemente una rutina de inicialización para las terminales PH0 y PH1 como de salida digital y las terminales PK5 y PK7 como de entrada digital. Identifique las terminales. Conecte en las terminales de salida un Led en cada una empleando una resistencia limitadora de corriente. Su activación se hará con lógica positiva. Conecte en las terminales de entrada un Switch en cada una alambrados con lógica positiva.

Implemente un ciclo continuo para leer el estado de los switches y reflejarlo en los Leds de acuerdo a la siguiente tabla:

Determine si es necesario configurar el registro de resistencias de Pull-up o Pull-down. Declare en el archivo fuente, los nombres de los registros empleados y sus direcciones empleando la directiva `#define`. Consulte el archivo *tm4c1294ncpdt.h*

6. Cuestionario

1. Sea el registro $GPIO_PORTA_DATA_R$ un registro definido en lenguaje C conteniendo el dato 0x92. ¿qué operación realiza las siguientes sentencias en lenguaje C? Especifique el resultado en formato binario y hexadecimal.

- $GPIO_PORTA_DATA_R \bar{} 0x01$
- $GPIO_PORTA_DATA_R | = 0x01$
- $GPIO_PORTA_DATA_R \& = 0x10$
- $GPIO_PORTA_DATA_R \& = (0x10)$

Explique qué función realiza cada sentencia según la *máscara* (valor del lado derecho del signo *igual*).

7. Conclusiones.

Referencias

- [1] Como citar: http://www.cva.itesm.mx/biblioteca/pagina_con_formato_version_oct/apa.htm
- [2] Autor, (Fecha de publicacion), Titulo, paginas, Fecha de recuperacion, Sitio web: <http://www.google.com>
- [3] Repositorio del proyecto <https://github.com/penserbjorne>