| | |
|---|---|
| | **Manual de Prácticas Microprocesadores** |
| **División: Ingeniería Eléctrica** | **Departamento: Ingeniería Electrónica** |

# Puertos de entrada/salida

## N.° de práctica: 07

| Nombre completo del alumno | | Firma |
|---|---|---|
| nombre | | |
| N.° de brigada: | Fecha de elaboración | Grupo: |

# Manual de Prácticas Microprocesadores

| División: Ingeniería Eléctrica | Departamento: Ingeniería Electrónica |
|---|---|

## 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.**

a. Presente un diagrama de conexión de una Resistencia de Pull-up y Pull-down y explique su funcionamiento.

b. Describa la función de los siguientes registros de los puertos de E/S y su configuración tras un Reset. Indique la dirección de cada registro en el mapa de memoria.

| REGISTRO | FUNCIÓN | ESTADO EN RESET |
|---|---|---|
| **GPIODATA Dir:_____** | | |
| **GPIODIR Dir:_____** | | |
| **GPIOAFSEL Dir:_____** | | |
| **GPIOPUR Dir:_____** | | |
| **GPIOPDR Dir:_____** | | |
| **GPIODEN Dir:_____** | | |
| **GPIOLOCK Dir:_____** | | |
| **GPIOPCTL Dir:_____** | | |

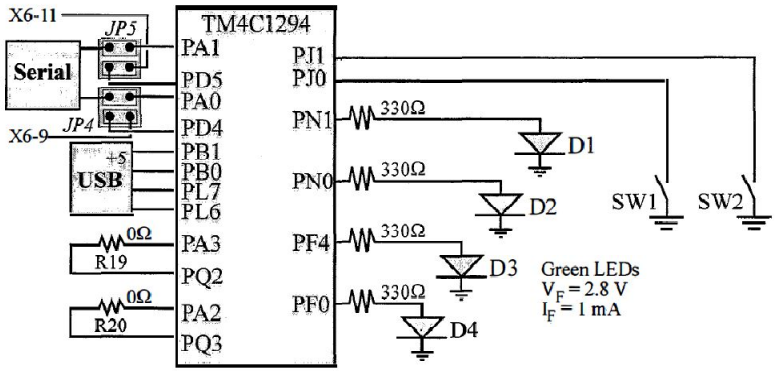
c. Describa la secuencia de pasos secuenciales para programar un puerto paralelo como digital (entrada y salida).

d. Describa el modo de funcionamiento de bits direccionables (o direccionamiento de bits específico) en un puerto GPIO.

e. Para hacer lectura y escritura de todos los bits de un puerto, ¿cuál es el registro al cual se hace acceso de lectura/escritura?

f. ¿Cómo se hace referencia a una dirección de un registro en lenguaje C?

g. Revise (y en medida de lo posible ejecute en la tarjeta de desarrollo) los programas ***blink.asm***, ***simpleIO.asm***, ***main.c*** y ***main2.c*** considerando el hardware de la tarjeta Tiva TM4C1293.




Explique detalladamente lo que hace cada programa.

NOTA: para los programas ***blink.asm***, ***simpleIO.asm*** incluya los archivos *"macros.s"* y *"gpio_regs.s"* en la carpeta del proyecto.

5. **Desarrollo.**

**En lenguaje ensamblador:** Incluya los archivos *macro.s* y *gpio_regs.s* en su programa. El archivo *macros.s* contiene la macro (o pseudo instrucción) para cargar un valor de 32 bits en un registro. El archivo *gpio_regs.s* contiene los nombre y las direcciones asociadas de los registros de puertos GPIO.

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:

| | |
|---|---|
| SW1 (PK5) | LED1 (PH0) |
| SW2 (PK7) | LED2 (PH1) |

Determine si es necesario configurar el registro de resistencias de Pull-up o Pull-down.

**En lenguaje C:** Realice el programa con la misma funcionalidad. Puede incluir el archivo *tm4c1294ncpdt.h*

**6. Cuestionario.**

a. ¿Cómo puede acceder a las terminales empleadas en el programa de forma específica?

b. 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 ^= 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").

c. Especifique la duración de la siguiente función de retardo. Para esto, investigue la duración en ciclos de reloj que tarda cada instrucción en la subrutina y calcule las veces que se ejecutan en cada llamada.

```
delay:
                LDR32 R8, 0x2FFFF   ; CONTADOR: 196,607 * T_Clock =xxxx[s]
dl_loop         SUBS  R8, #1        ;
                BNE   dl_loop       ;
                BX    LR
```

d. Determine el valor de la carga inicial para que esta rutina tenga una duración de 1 segundo.

## 7. Conclusiones.

## 8. Bibliografía