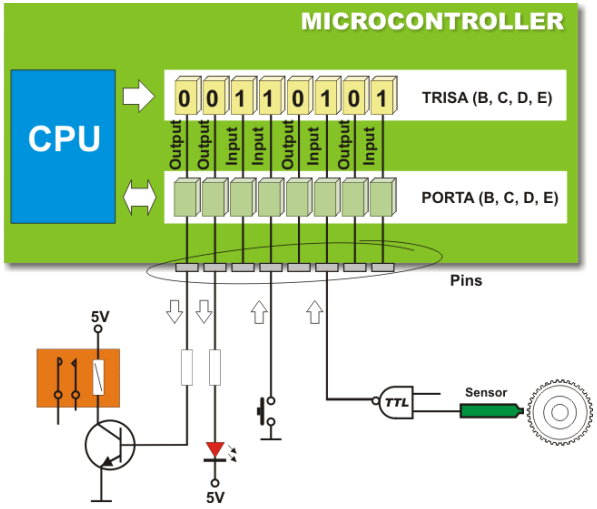
Laboratorio de Microcomputadoras

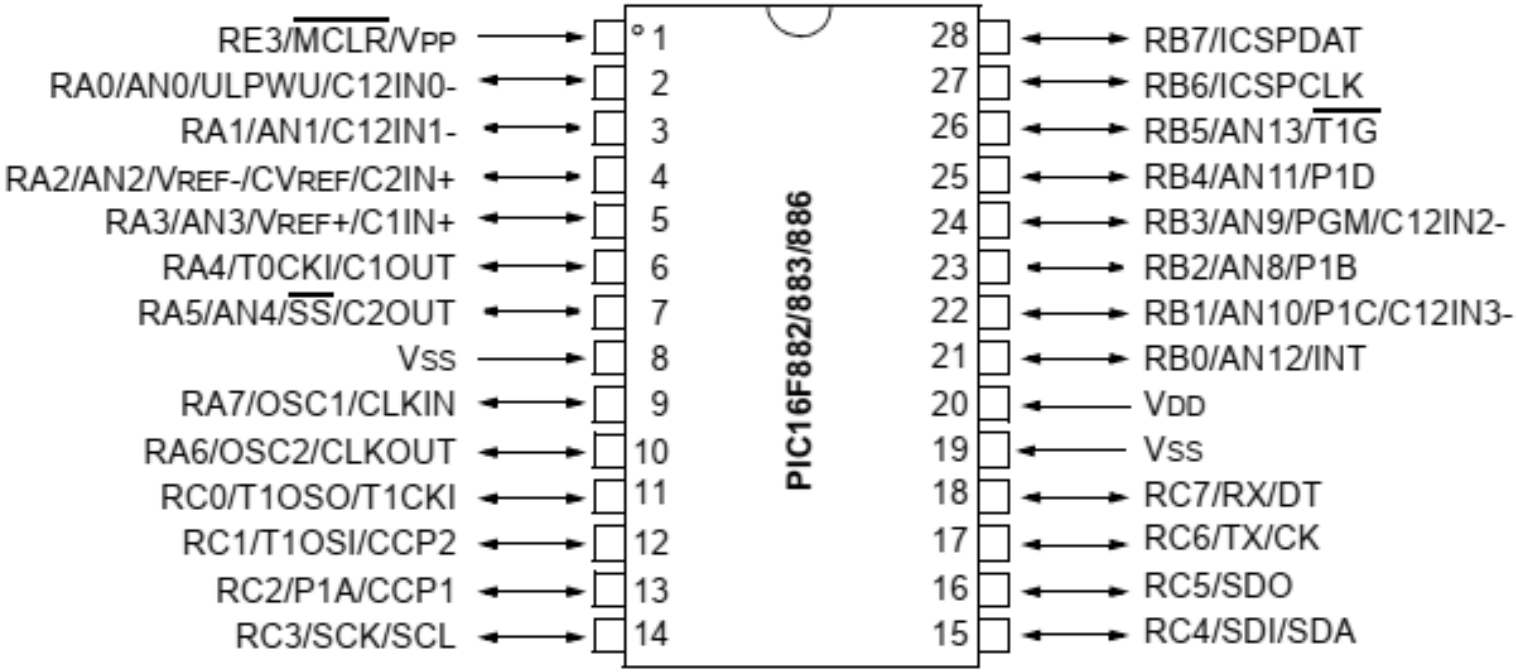
Puertos Paralelos
E/S digitales



Diana A. Cruz Hernández
Amaranto de J. Dávila Jáuregui

Puertos Paralelos PIC16f886

- 24 líneas de entrada salida E/S
  - Puerto A: 8 bits
  - Puerto B: 8 bits
  - Puerto C: 8 bits



Registros que controlan los puertos

- Registro de configuración
  - TRISx
    - 1 entrada
    - 0 salida
- Registro de datos
  - PORTx

PUERTO	PORTX	TRISX
A	05h	85h
B	06h	86h
C	07h	87h

Para el puerto A y B además se debe escribir 0x00 en el registro ANSEL (188h) y ANSELH(189h), para configurar el puerto A y B como E/S digitales respectivamente.

Ejemplo

Al poner en cero todo el registro ANSELH, se configura todo el puerto B como digital

	7	6	5	4	3	2	1	0
ANSELH	0	0	0	0	0	0	0	0

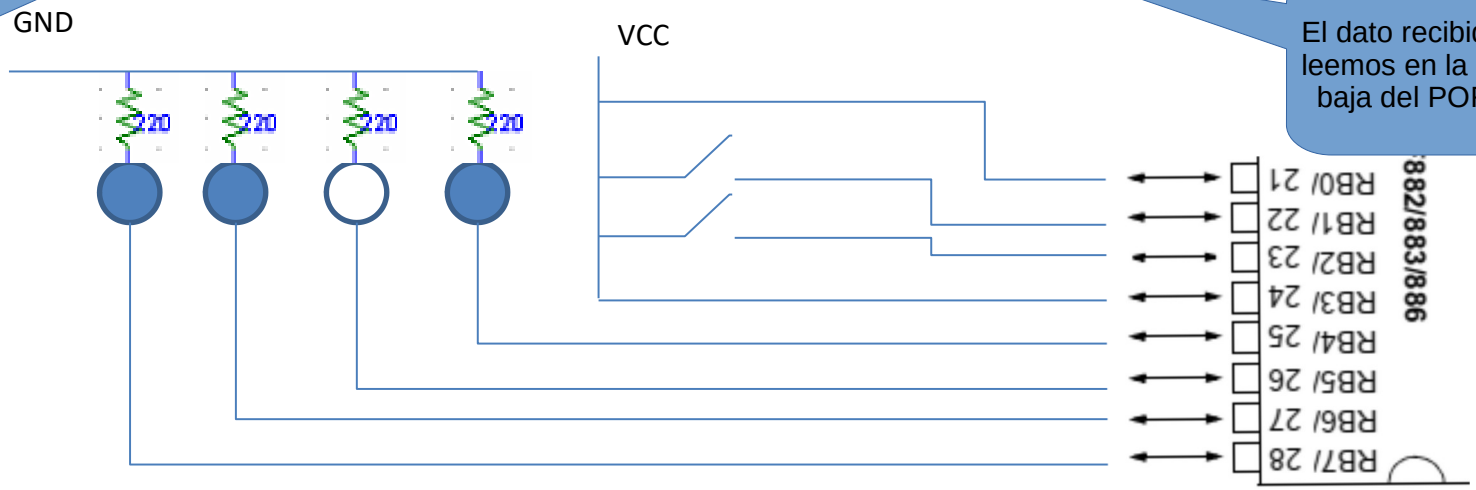
En este caso la parte alta del puerto se configuró como salida y la parte baja como entrada

	7	6	5	4	3	2	1	0
TRISB	0	0	0	0	1	1	1	1

	7	6	5	4	3	2	1	0
PORTB	1	1	0	1	1	0	0	1

En este caso el valor de la parte alta del PORTB indica el dato que sale

El dato recibido lo leemos en la parte baja del PORTB



Enmascaramiento

- El enmascaramiento es el proceso mediante el cual se cambian algunos bits de un dato pero el resto no se alteran
- Se puede hacer mediante una operación de
  - AND con el dato y una constante (mascara)
    - Cuando queremos apagar algunos bits sin modificar el resto
  - OR con el dato y una constante
    - Cuando queremos encender algunos bits sin modificar el resto

Enmascaramiento

- Si un bit está configurado como entrada y no hay nada conectado a la entrada no se puede asumir un valor por default.
- Por lo que es conveniente el enmascaramiento
  - AND con una constante que tenga ceros en los bits que queremos eliminar y unos en los que queremos conservar
  - Por ejemplo supongamos que configuramos todo el puerto B como entrada pero solo conectamos señal de entrada en la parte baja

		7	6	5	4	3	2	1	0
	PORTB	*	*	*	*	1	1	0	1
AND	MASCARA	0	0	0	0	1	1	1	1
	ENTRADA	0	0	0	0	1	1	0	1

Retardos

- Una retardo es una rutina cuyo objetivo es que el procesador consuma ciclos de instrucción sin realizar alguna tarea específica, generalmente se utiliza para poder sincronizarse con algún mecanismo o dar tiempo de leer una entrada externa, etc.

- Los retardos se pueden implementar mediante ciclos anidados vacíos.

Cálculo del tiempo del retardo

- Considerando que el microcontrolador tiene un oscilador de 4MHz cada instrucción se ejecuta en 1µs, excepto los saltos que tardan dos.

  El ciclo de instrucción se calcula mediante:

  Tcy = 1 / (Fosc/4) = 1 / (4 Mhz/4) = 1 µs

- De tal forma que para calcular el tiempo del retardo podemos aplicar la siguiente fórmula:
  - Tiempo del retardo: [nifc + nr (nidc)]*1µs
  - Donde:
    - nifc:número de instrucciones fuera del ciclo
    - nr: número de repeticiones
    - nidc: número de instrucciones dentro del ciclo

- Si tenemos varios ciclos anidados esta fórmula se anida también

Ejemplo

- Las instrucciones de salto se cuentan como dos instrucciones dado que tardan dos ciclos de instrucción en ejecutarse,
- En el caso de la instrucción decfsz sólo se contabiliza como salto cuando la condición se cumple

```
RETARDO
        movlw   N           ; 1
        movwf   CONT        ; 1
CICLO   decfsz  CONT,f      ; 1*(N-1)+2
        goto    CICLO       ; 2*(N-1)
```

- De tal forma que para este código la fórmula queda como:

  Tiempo del retardo: [4 + (N-1) (3)]*1µs

Ejemplo 2

- En este ejemplo se están anidando tres ciclos para lograr un retardo mayor

Cálculo del retardo:

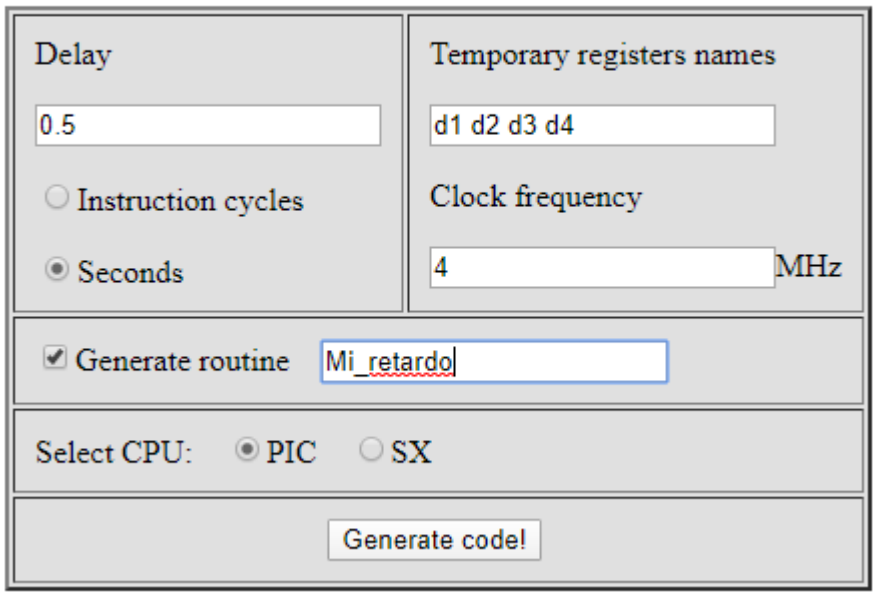
[4 + (P-1)
([4 + (M-1)
([4 + (N-1) (3)])
])
]*1µs

```
RETARDO
        movlw   P           ; 1
        movwf   CONT3       ; 1
CICLO3  movlw   M           ; 1
        movwf   CONT2       ; 1
CICLO2  movlw   N           ; 1
        movwf   CONT        ; 1
CICLO   decfsz  CONT,f      ; 1*(N-1)+2
        goto    CICLO       ; 2*(N-1)
        decfsz  CONT2,f     ; 1*(M-1)+2
        goto    CICLO2      ; 2*(M-1)
        decfsz  CONT3,f     ; 1*(P-1)+2
        goto    CICLO3      ; 2*(P-1)
```

Herramientas

Para facilitar el cálculo de las rutinas de retardo y hay herramientas, una que es la que se encuentra en la siguiente liga.

http://www.golovchenko.org/cgi-bin/delay




```
; Delay = 0.5 seconds
; Clock frequency = 4 MHz

; Actual delay = 0.5 seconds = 500000 cycles
; Error = 0 %

        cblock
        d1
        d2
        d3
        endc

Mi_retardo
                        ;499994 cycles
        movlw   0x03
        movwf   d1
        movlw   0x18
        movwf   d2
        movlw   0x02
        movwf   d3
Mi_retardo_0
        decfsz  d1, f
        goto    $+2
        decfsz  d2, f
        goto    $+2
        decfsz  d3, f
        goto    Mi_retardo_0

                        ;2 cycles
        goto    $+1

                        ;4 cycles (including call)
        return
```

Retardo

- Genera una rutina de retardo de 0.5 segundos llamada Mi_retardo, esta se debe llamar mediante la instrucción:

  CALL Mi_retardo

- La rutina usa tres registros de memoria temporales d1, d2, d3, éstas deben estar declaradas al inicio del código mediante la directiva cblock y se debe especificar la dirección de memoria a partir de la cual van a quedar almacenadas, por ejemplo cblock 0x20

```
; Delay = 0.5 seconds
; Clock frequency = 4 MHz

; Actual delay = 0.5 seconds = 500000 cycles
; Error = 0 %

        cblock
        d1
        d2
        d3
        endc

Mi_retardo
                        ;499994 cycles
        movlw   0x03
        movwf   d1
        movlw   0x18
        movwf   d2
        movlw   0x02
        movwf   d3
Mi_retardo_0
        decfsz  d1, f
        goto    $+2
        decfsz  d2, f
        goto    $+2
        decfsz  d3, f
        goto    Mi_retardo_0

                        ;2 cycles
        goto    $+1

                        ;4 cycles (including call)
        return
```

Ejemplo 1. Diseñar un programa que encienda y apague un LED en el pin RB0 cada 0.5 [s]

```
processor pic16f886
#include p16f886.inc

cblock 0x20
    d1
    d2
    d3
    endc

ORG 0X00
GOTO 0X05
ORG 0X05

BSF STATUS,RP0    ;BANCO1
BCF STATUS,RP1
BCF TRISB,0       ;RB0 COMO SALIDA
BSF STATUS,RP1    ;BANCO3
CLRF ANSELH  ;PORTB DIGITAL
BCF STATUS,RP0
BCF STATUS,RP1    ;BANCO0

LOOP BSF PORTB,RB0
    ; CALL Mi_retardo
     BCF PORTB,RB0
    ; CALL Mi_retardo
     GOTO LOOP
```

```
Mi_retardo
            ;499994 cycles
    movlw   0x03
    movwf   d1
    movlw   0x18
    movwf   d2
    movlw   0x02
    movwf   d3
Mi_retardo_0
    decfsz  d1, f
    goto    $+2
    decfsz  d2, f
    goto    $+2
    decfsz  d3, f
    goto    Mi_retardo_0

            ;2 cycles
    goto    $+1

            ;4 cycles (including call)
    return

END
```

- *archivo enciende.asm

Práctica

Realizar desplazamientos de un bit sobre el PUERTO B, la dirección del desplazamiento estará determinada por una entrada digital, de tal forma que si la entrada es:

- “Uno” el desplazamiento es hacia la derecha y si es
- “Cero” el desplazamiento es hacia la izquierda,

Dicha entada llega por el bit cero del PUERTO A

El tiempo entre desplazamientos del bit es de 0.5 [s]

El circuito se encuentra en el archivo desplaza.pdsprj