COMENTARIO TECNICO

Microchip Tips & Tricks...

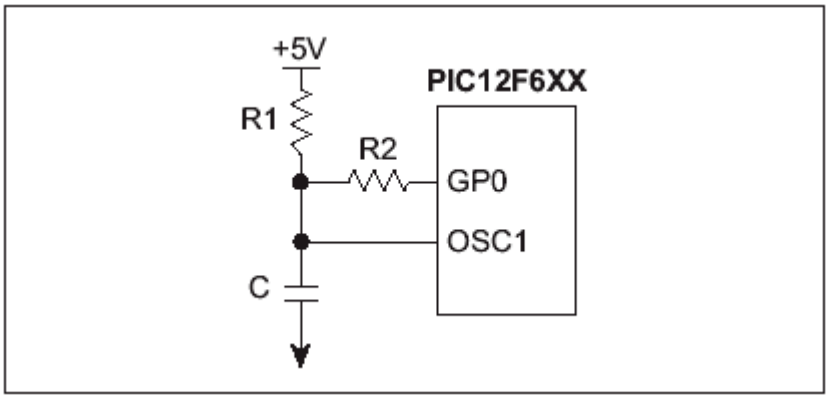
Por el Departamento de Ingeniería de Electrónica Elemon S.A.

N. de R: La presente serie de artículos técnicos tiene por objetivo proponer consejos y trucos que esperamos le sean de utilidad al diseñador de sistemas y aplicaciones con MCUs Microchip. En estas primeras entregas nos dedicaremos a los dispositivos más pequeños de 8 pines como la familia 12Fxxx.

TIP Nº1.

Oscilador RC de doble frecuencia.

En el presente "Tip", nos ocuparemos de una situación muy útil cuando queremos trabajar con nuestro PIC en distintas condiciones de consumo. El truco aquí será lograr que nuestro MCU pueda seleccionar por software entre 2 frecuencias de trabajo de nuestro oscilador RC en forma práctica y sencilla. La frecuencia menor nos dará un menor consumo, manteniendo la operatividad de nuestro sistema, mientras que la frecuencia mayor nos dará una rápida respuesta ante situaciones de control o cálculo.



Pasos a seguir:

1. Luego de un reset, el pin I/O está en alta (Z)

2. Forzarlo como Output ‘1’ en el pin I/O elegido (GP0)

3. R1, R2 y C determinan la frecuencia de OSC (oscilador)

4. También funciona con capacitores adicionales

La frecuencia del PIC® MCU en **"RC oscillator mode"** depende de la resistencia y capacitancia en el **pin OSC1**. La resistencia equivalente cambia con la tensión de salida en GP0. GP0 en ‘1’ pone R2 en paralelo con R1 reduciendo la resistencia en OSC 1 y aumentando la frecuencia.GP0 como entrada aumenta la resistencia en OSC1 minimizando la corriente en R2, disminuyendo la frecuencia y el consumo de potencia.

Resumen:

GP0 = Input: Baja velocidad → corrientes pequeñas
GP0 = Output : Alta velocidad → procesamiento rápido

TIP Nº2

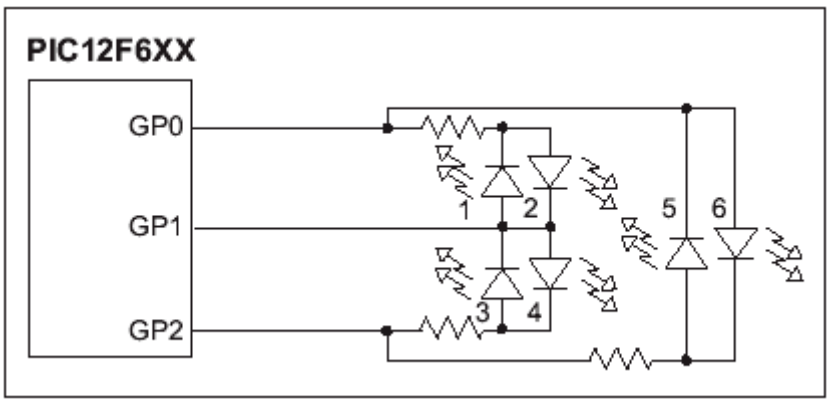
Multiplexación de pines I/O como Outputs.

En pequeños MCUs, es una necesidad común enfrentarse a la falta de puertos I/O en ciertas condiciones del diseño. El empleo de diodos, es una muy buena solución al tema. Un diodo o una combinación de diodos puede ser habilitada alternando los I/Os entre high y low, o estableciéndolos como "inputs" (Z). El número de diodos que se pueden controlar depende del numero de I/Os (GP) utilizados.

Se cumple que: D = GP x (GP - 1).

Ejemplo : 6 LEDs en 3 Pines I/O

GPx			LEDs					
0	1	2	1	2	3	4	5	6
0	0	0	0	0	0	0	0	0
0	1	Z	1	0	0	0	0	0
1	0	Z	0	1	0	0	0	0
Z	0	1	0	0	1	0	0	0
Z	1	0	0	0	0	1	0	0
0	Z	1	0	0	0	0	1	0
1	Z	0	0	0	0	0	0	1
0	0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0	0
0	1	1	1	0	0	0	1	0
1	0	0	0	1	0	0	0	1
1	0	1	0	1	1	0	0	0
1	1	0	0	0	0	1	0	1
1	1	1	0	0	0	0	0	0



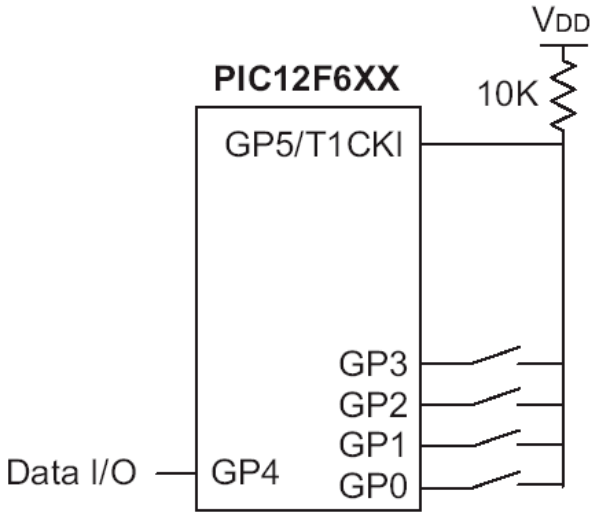
TIP Nº 3

Leyendo un Dip Switch.

El input del timer puede ser usado para testear que switch(s) está cerrado. El input del Timer1 contiene una resistencia de pull-up.
Secuencialmente, cada switch I/O es seteado como I y se chequea si hubo un incremento en el Timer1 que indica que el switch está cerrado.

Cada bit del registro del DP representa la posición del switch correspondiente. Estableciendo el Timer1 en FFFFh y habilitando su interrupción, un incremento causará un "rollover "y generará una interrupción. Esto simplifica el software ya que se elimina el testeo del bit en el registro TMR1L . Secuencialmente establezca cada GPIO en "Input" y verifique si hubo un incremento en el TMR1 (o 0 si se usa el pin I/O estándar)

```
        movlw   b'11111111'
        movwf   TRISIO
        movwf   DIP
        movlw   b'00000111'
        movwf   T1CON
        movlw   b'11111110'
        movwf   Mask
        clrf    GPIO
LOOP
        clrf    TMR1L
        movf    Mask,W
        movwf   TRISIO
        btfsc   TMR1L,0
        andwf   DIP,F
        bsf     STATUS,C
        rlf     Mask,F
        btfsc   Mask,4
        goto    Loop
        retlw   0
```



TIP Nº 4

Barrido de muchas teclas con una sola entrada.

Muchas técnicas pueden emplearse para “ahorrar pines” cuando tenemos que “barrer” un teclado, desde armar una matriz de barrido por filas y columnas, utilizar un conversor A/D y resistores para sensar diferencias de tensión y otras técnicas igualmente ingeniosas. La que describiremos en este “tip” hace uso del Timer del PIC y de la diferencia de los tiempos de carga de múltiples RC habilitados al pulsar cada una de las teclas de nuestro teclado.

El tiempo requerido para cargar un capacitor depende de la resistencia entre VDD y el capacitor. Al apretar un pulsador, VDD se conecta a un punto diferente del conjunto de resistencias haciendo que la resistencia entre y el capacitor disminuya, lo cual reduce el tiempo de carga del capacitor. Un timer es usado con un comparador o con una entrada digital para medir el tiempo de carga del capacitor. Este tiempo es utilizado para determinar qué pulsador fue apretado.

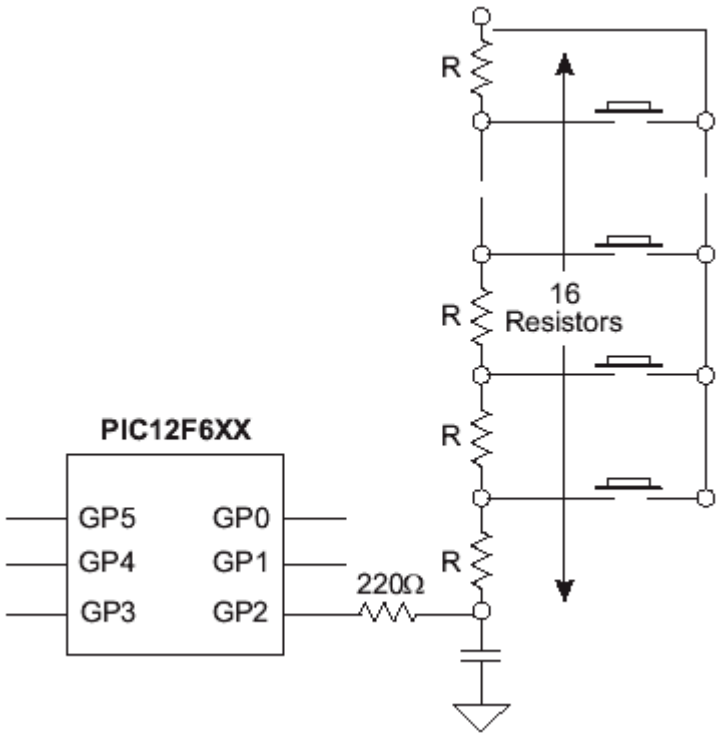
Secuencia del software:

1. Configurar GP2 para que entregue una baja tensión y así descargar el capacitor mediante la R de I/O.

2. Configurar GP2 como un comparador de entrada y CVREF.

3. Utilizar un timer para medir el tiempo. Si este tiempo medido es mayor que el máximo permitido entonces se debe repetir; sino se puede determinar cual de los pulsadores fue presionado.

Cuando se aprieta un pulsador, el divisor resistivo provoca un cambio en el valor de RC.



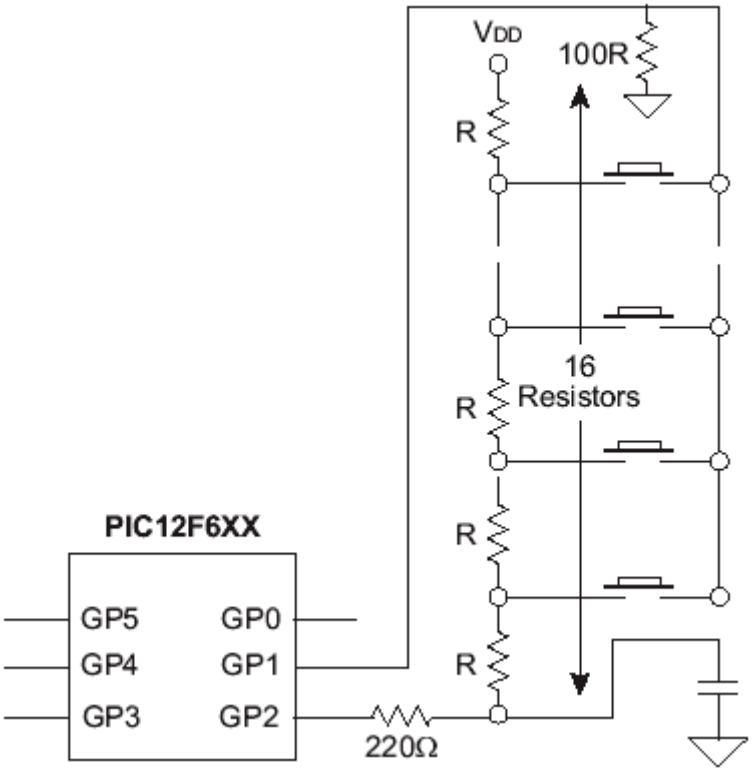
TIP Nº 5

Barrido de muchas teclas con una sola entrada y Wake - Up desde modo Sleep.

Se puede agregar un I/O adicional para despertar el sistema cuando se presiona un pulsador. Antes de entrar en modo "sleep", configurar GP1 como entrada habilitando el **"interrupt on change"** y GP2 como salida "high". La resistencia de pull down mantiene GP1 "low" hasta que se aprieta un pulsador, el valor de dicha resistencia de pull down debe ser 100 veces superior a las de las "R" de sensado. Así, GP1 es llevado a high vía GP2 y VDD generando la interrupción. Luego, GP2 se configura como salida "low" para descargar el capacitor por medio de la R=220 ohms. GP1 se configura como salida "high" y GP2 como entrada para medir el tiempo de carga del capacitor.

- GP1 pin conectado al común de los pulsadores.
- Habilitar el **"wake-up on port change".**
- Establecer GP1 → entrada y GP2 "high" antes del "Sleep"

- Al presionar un pulsador el PIC® MCU despierta, GP2 debe llevarse a low para descargar el capacitor.
- Establecer GP1 high con el **"wake-up"** para detectar pulsador



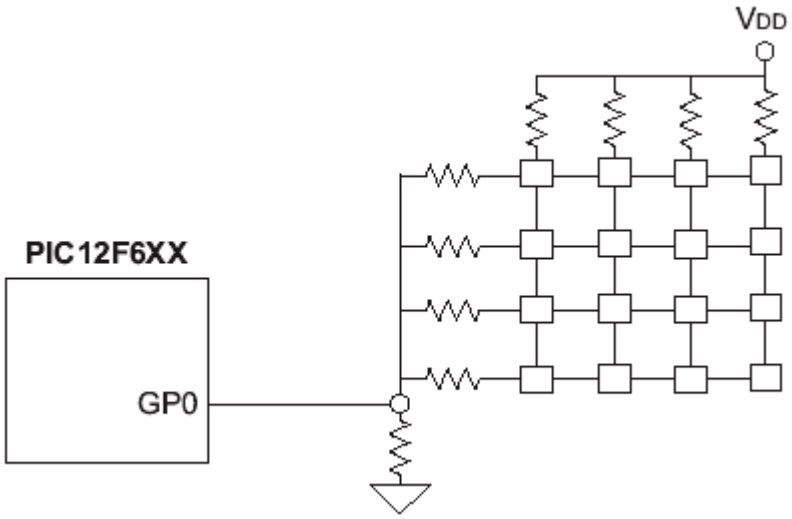
TIP Nº 6

Teclado 4 x 4 con una sola entrada.

Tal como se había comentado en los tips anteriores, aquí se combina la técnica de teclado “matricial” con la del uso del conversor A/D del PIC.

Seleccionando cuidadosamente el valor de las resistencias, cada pulsador genera un único voltaje.

Este voltaje puede ser medido con el A/D para determinar qué pulsador fue. Resistencias de mayor precisión (1% lo recomendado) deben ser utilizadas para maximizar la unicidad. El A/D mide alrededor de 0 cuando no se presiona ningún pulsador.



TIP Nº 7

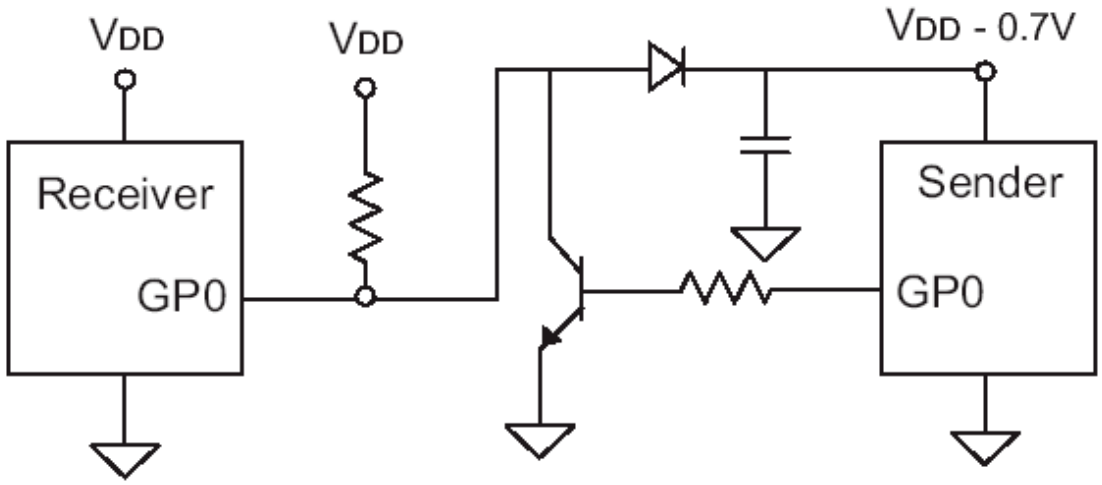
Datos y Alimentación por un solo hilo al estilo “1 – Wire Dallas”.

Muchas veces nos vemos en la necesidad de implementar algún tipo de comunicación “unidireccional” a un solo hilo entre un dispositivo y otro, y además tener que alimentar a alguno de ellos durante la comunicación. Esta situación se da cuando deseamos implementar una comunicación a un solo hilo del tipo “1 – Wire” de Dallas o similares donde el dispositivo “emisor” (Sender) “vive” cuando se conecta al dispositivo “receptor” (Receiver) ya que el emisor no posee energía propia y toma la del receptor.

En esta aplicación, un único I/O puede ser usado para datos unidireccionales y como una fuente para otro microcontrolador. El I/O se matiene en "high" por medio de la R de "pull-up" conectada a VDD. El emisor utiliza un transistor "pull-down" para llevar la línea de datos a "low" o inhabilita al transistor que permita que el "pull-up" pase a "high" para enviar datos al receptor. El VDD se suministra al emisor por medio de la línea de datos.

El capacitor estabiliza el VDD del emisor y el diodo previene la descarga del capacitor mediante la línea de I/O cuando esta en low.

El VDD del emisor es una caída de diodo menor que el del receptor.



TIP Nº 8.

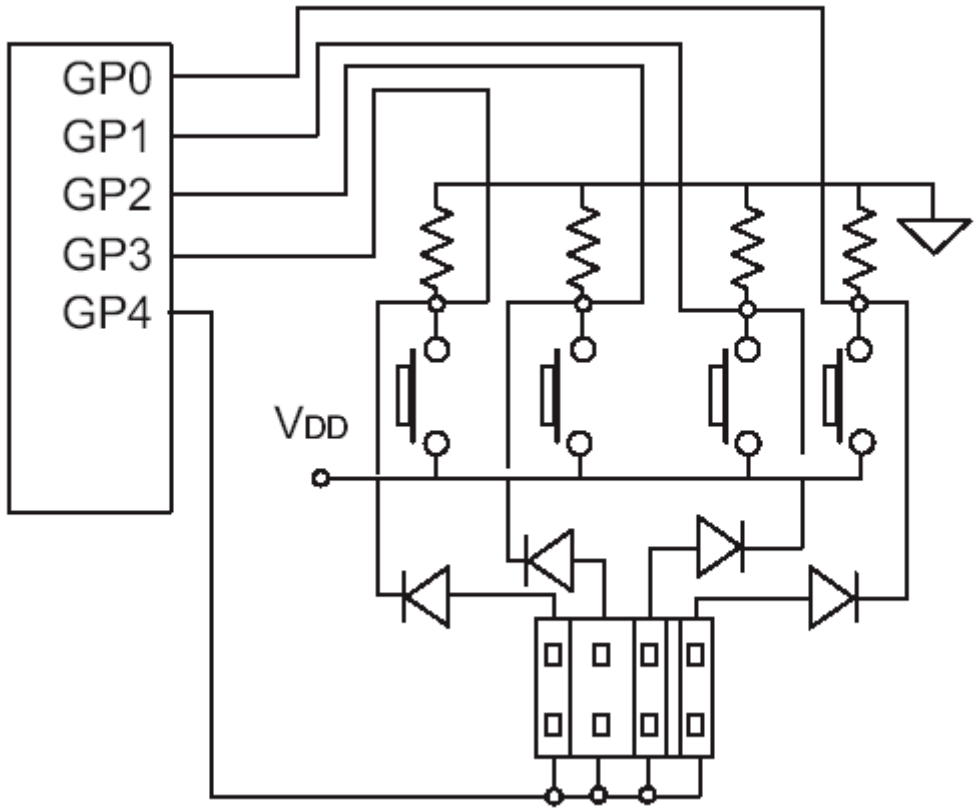
Decodificación de Pulsadores y de Jumpers compartiendo I/Os.

Cuando los puertos I/Os no abundan, todo recurso es válido....

Pulsadores y jumpers pueden compartir I/Os usando otro puerto I/O para seleccionar cual va a ser leído. Ambos comparten una resistencia de "pull-down". Por esta razón, se leerá un ‘0’ hasta se apriete un pulsador o un jumper sea conectado.

Cada input (GP3/2/1/0) comparte un jumper y un pulsador. Para leer la configuración del jumper hay que establecer GP4 como "high" y cada jumper conectado será leído como un ‘1’ en su correspondiente puerto I/O o ‘0’ si no está conectado.

Con GP4 en Low, un pulsador apretado será leído como ‘1’, o se leerá un ‘0’ de lo contrario.



- Con GP4 = 1 y ningún pulsador apretado, lee configuración ID.
- Con GP4 = 0, lee pulsadores.

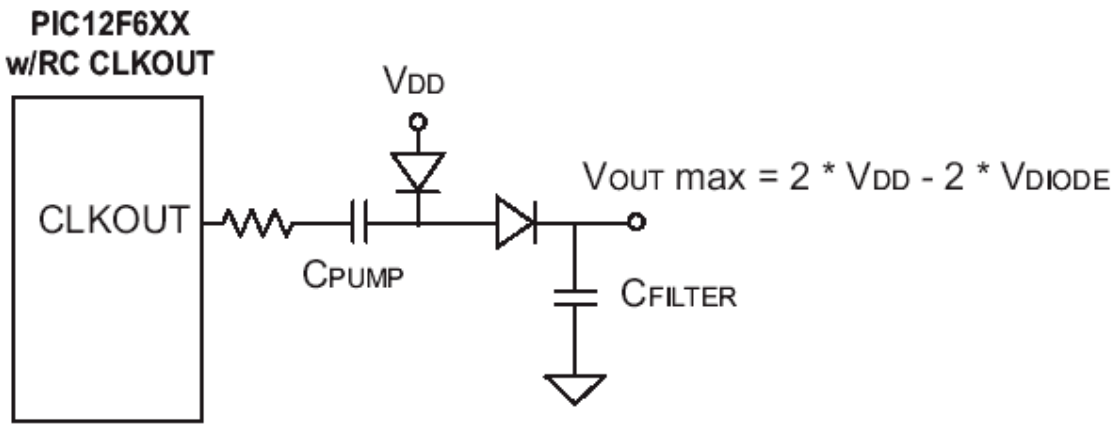
TIP Nº 9.

Generando tensiones elevadas con un puerto I/O del MCU.

Tensiones mayores que VDD pueden ser generadas alternando los niveles lógicos de un puerto I/O. El pin **CLKOUT/OSC2** del PIC® MCU entrega una oscilación de señal a un cuarto de la frecuencia de **OSC1** cuando se utiliza el modo **"external RC oscillator".**

Cuando **OSC2** esta en Low, el diodo de VDD está en directa y conduce corriente cargando **Cpump.**

Después que **OSC2** esté en "high", el otro diodo comienza a conducir y carga el **Cfilter.** Esto resulta en una carga igual al doble de VDD menos dos caídas de diodo. Esto puede ser utilizado con PWM, alternando un puerto I/O o alternando otro pin.



Solicite mayor información en:

Electrónica Elemon S.A.
www.elemon.com.ar
e-mail: ventas@elemon.com.ar