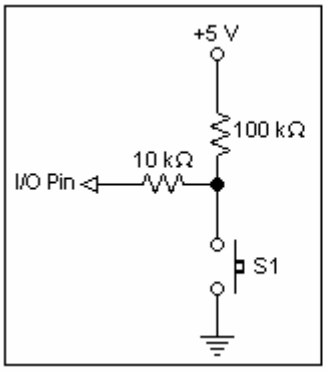
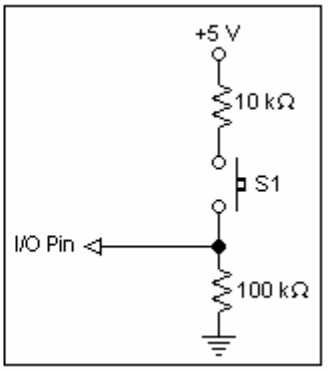
Basic Express
Nota de aplicación

Conexión de botones e interruptores

Introducción

Esta nota de aplicación describe una serie de métodos hardware y software para la conexión de botones e interruptores con un sistema BasicX.





La necesidad de recibir información o una entrada de usuario desde el exterior es una de las aplicaciones más importantes de los microcontroladores. Por ejemplo, se podría utilizar un juego de pulsadores para poner en funcionamiento y controlar una fresadora conectada a un BasicX. Al mismo tiempo, los interruptores limitadores podrían permitir al sistema BasicX detectar la apertura de una tapa protectora, y que el procesador apague automáticamente la fresadora por razones de seguridad.

Figura 1 **Figura 2**

Interfaz de Hardware

Hay múltiples maneras de leer el estado de un pulsador o interruptor. Esta nota de aplicación se centra en los dos métodos más frecuentes. El primer método (aparece en la figura 1) se trata de la conexión del pin I/O pin a través de una resistencia de alto valor. Un interruptor baja el pin I/O a través de una segunda resistencia más pequeña. Este método funcionará igualmente bien con interruptores normalmente abiertos o cerrados.

El segundo método (aparece en la figura 2) se trata en la conexión del pin I/O a tierra a través de una resistencia de alto valor. En el diagrama de ejemplo, se utiliza una resistencia de 100 kΩ. El botón entonces se utiliza para subir el pin I/O a través de una resistencia pequeña. Al mismo tiempo, este método funcionará igualmente bien con interruptores normalmente abiertos o cerrados.

Programas de ejemplo

Este primer programa de ejemplo supone que está utilizando el pin 16 I/O de un sistema BasicX y, según el diagrama de la figura 1, que tiene un interruptor cableado. El programa lee el estado de cualquier interruptor conectado al pin y escribe ese valor al segundo pin (pin I/O 17, en este caso). El estado lógico del segundo pin I/O puede después verificarse con un osciloscopio o con una sonda lógica. También es posible utilizar un indicador LED (ver figura 3).

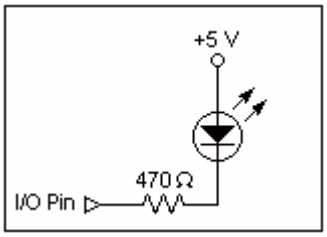


Figure 3

Programa de ejemplo 1:

```
Sub Main()

    Dim State as Byte

    Do
        ' Leer pin I/O 16.
        State = GetPin(16)

        ' Copiar el estado al pin 17.
        Call PutPin(17, State)
    Loop

End Sub
```

Este programa funciona bien con interruptores, pero si ya ha puesto a prueba este programa utilizando un botón temporal, habrá podido comprobar que el pin 17 sólo permanece en su estado temporal mientras esté pulsando el botón. Este programa no funcionaría correctamente si quisiera conmutar el estado de un pin I/O y que permaneciera en ese estado sin tener que mantener pulsado el botón.

Para poder utilizar los botones o interruptores temporales se necesita un enfoque diferente. Este ejemplo supone que está utilizando el pin 16 para la entrada y el pin 17 para la salida y que dispone de un botón pulsador cableado de acuerdo con el diagrama de la figura 1. El programa de ejemplo 2 utilizará el modo de operación de números binarios de bit por bit Xor (OR exclusivo) para conmutar el estado del pin 17 cada vez que se presione el botón temporal.

Programa de ejemplo 2:

```
Sub Main()

' Este programa lee el estado del interruptor y conmuta el estado del pin
de salida cuando se presione el interruptor.'

    Const InputPin As Byte = 16
    Const OutputPin As Byte = 17

    ' Configuración de los pines.
    Call PutPin(InputPin, bxInputTristate)
    Call PutPin(OutputPin, bxOutputHigh)

    Dim State as Byte

    State = 0

    Do
        ' Conmutación del estado si se presiona el interruptor.
        If GetPin(InputPin) = 0 Then

            State = State Xor 1

            ' Pausa de un cuarto de segundo para cambiar de
entrada/salida (tiempo de debounce).
            Call Delay(0.25)
        End If

        ' Escribir el estado al pin de salida.
        Call PutPin(OutputPin, State)
    Loop

End Sub
```

Un programa similar se encuentra en un fichero independiente denominado **Buttons.bas**

2.00.A

Traducción Española: Alicia Bernal, revision: Pablo Pompa www.superrobotica.com