

CONTADOR DE EVENTOS DE 4 DIGITOS MULTIPROPOSITO

Eje temático #6: Aplicaciones académicas
Treser, Lucas Martín - Argentina
lmtreser@gmail.com

Se pretende lograr que el alumno asimile los conceptos básicos sobre la programación de microcontroladores en lenguaje Basic que lo guiarán hacia la construcción de un contador digital de 4 dígitos con múltiples aplicaciones en el ámbito industrial.

Las características más resonantes del prototipo son: sencilla construcción y programación, diseño con microcontrolador PIC de la firma Microchip, rango de conteo desde 0 a 9999, reinicio manual mediante pulsador, indicación de final de cuenta alcanzado, entrada de conteo por interrupción (flanco ascendente), tamaño compacto, entrada para gran cantidad de tipo de sensores para leer los datos.

Palabras claves: microcontrolador, programación, lenguaje Basic, display, prototipo, sensor hall, conceptos básicos, reemplazo de tecnologías.

Introducción.

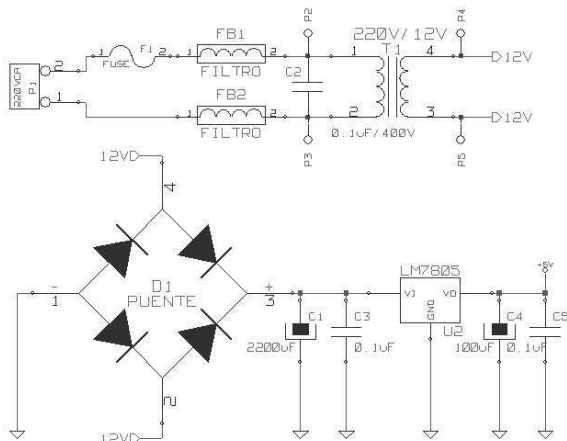
La idea de poder realizar un prototipo funcional, con amplias posibilidades de su inclusión en una aplicación comercial y sin perder el foco principal que es que el alumno aprenda lo que es un microcontrolador, como funciona, como es su programación y la variedad de formas para crear el código, resulta sumamente tentador a la hora de desarrollar este didáctico proyecto.

El hardware.

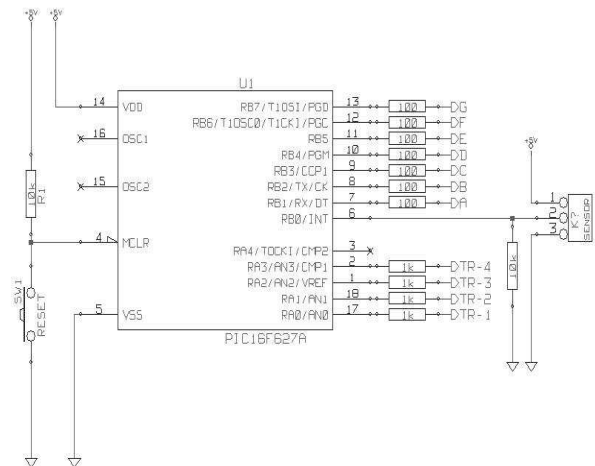
El circuito diseñado es muy sencillo, incluso puede armarse en un protoboard o plaqueta de pruebas para su puesta en marcha. Básicamente consta de tres bloques principales:

1. Fuente de alimentación.
2. Microcontrolador y sensor.
3. Visualización.

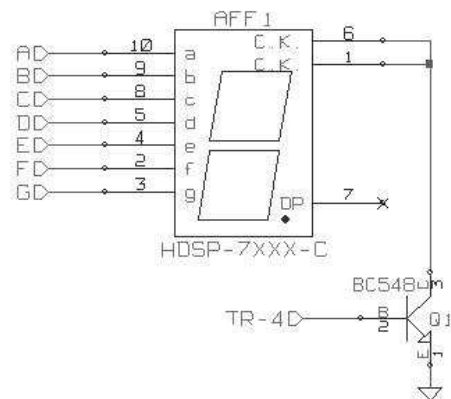
1. Fuente de alimentación. Es el bloque encargado de convertir la tensión de línea a la tensión de trabajo del microcontrolador. Para lograrlo dispone de un transformador que reduce los 220VCA a 12VCA, luego dicha tensión es rectificada por un puente de diodos, filtrada por capacitores y regulada con un circuito integrado LM7805 [1] que es capaz de entregar 5VCC con un máximo de corriente de 1A, más que suficiente.



2. Circuito de control. Basado en un microcontrolador PIC16F627A [2] es el encargado de recibir la señal desde el sensor adecuado (que puede ser mecánico, barrera infrarroja, de tipo hall, etc.), procesarla y luego enviar los datos al bloque de visualización. También deberá realizar el multiplexado de los displays de 7 segmentos. Para que este bloque funcione adecuadamente es necesario cargar en la memoria del microcontrolador las instrucciones operativas.

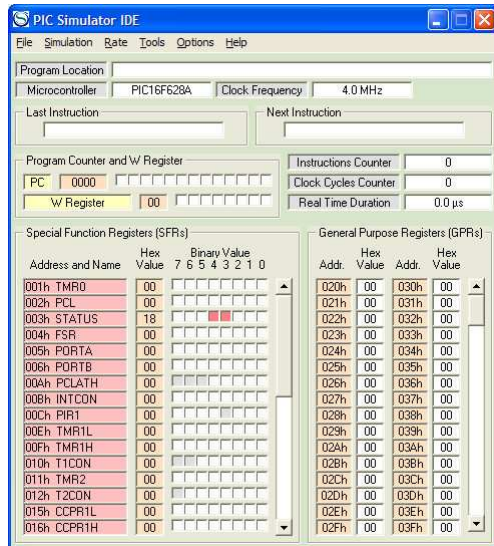


3. Circuito de visualización. Conformado por 4 displays de 7 segmentos y 4 transistores NPN que serán los encargados de activar uno a uno los dígitos de manera muy rápida que resultará imperceptible al ojo humano. Esta técnica de multiplexado es sumamente necesaria ya que no contamos con la cantidad suficiente de pines de salida en el microcontrolador para poder manejar todos los displays.



Creación del código.

Para la creación del código que le dirá al microcontrolador las tareas que deberá realizar utilizaremos el software "PIC Simulator IDE" [3]. El mismo consta de un potente simulador junto a un compilador BASIC.



Gracias a él podremos escribir el código en lenguaje BASIC, simular su comportamiento para depurar errores, en caso que los hubiera, y luego compilarlo para crear el archivo que cargaremos en la memoria de programa del PIC.

El código no nos debería llevar demasiada tarea. A grandes rasgos tendremos que monitorizar un pin de entrada por presencia de la señal de conteo, incrementar y guardar la cuenta en una variable, convertir dicha variable y exhibirla en los displays mediante la técnica de multiplexado.

Para desarrollar esas tareas nos valemos de varias funciones importantes que facilitarán la operación [4].

Uso de interrupciones. Una interrupción permite llamar la atención del microcontrolador en cualquier momento sin importar las instrucciones que este ejecutando. En nuestro caso fue

necesario utilizar la "Interrupción RB0" para monitorizar la presencia de señal en el pin de entrada. Este recurso permite ejecutar un segmento de código específico cuando existe un cambio de estado en el pin RB0 del microcontrolador. Existen varios parámetros a configurar, los más importantes son los siguientes:

`INTCON.INTE = 1`

Habilita la interrupción en el pin RB0

`INTCON.GIE = 1`

Habilita todas las interrupciones del micro

On Interrupt

Sentencia con los comandos que deberán ser ejecutados cuando este evento ocurra.

`INTCON.INTF = 0`

Al finalizar se deberá habilitar nuevamente la interrupción.

Uso de tablas. Para llevar el dato binario a los pines de salida correspondientes se utilizan tablas que convierten el valor de la variable en un número binario.

Nos valemos de la instrucción "LookUp" que buscará en una tabla los valores ya definidos (del 0 al 9) de acuerdo al valor de un índice, en nuestro caso son cuatro: unidad, decena, centena y unidad de mil. El valor recuperado se almacenará en otra variable llamada "mascara", la encargada de cargar el puerto de salida con los datos correctos. Un ejemplo de utilización sería:

`mascara = LookUp (valor1, valor2, valor3, valor4, valor5, valor6, valor7, valor8, valor9), indice`

Técnica de multiplexado. Podremos manejar los 4 displays conectándolos en paralelo y activándolos de a uno por vez mediante sendos transistores conectados en el cátodo común de cada uno. Si

hacemos esto muy rápido el ojo humano no percibirá el cambio y parecerá que se encuentran todos encendidos al mismo tiempo. Como ya dijimos ahorraremos algunos componentes y reduciremos el consumo ya que estarán encendidos por un breve periodo de tiempo.

Es importante experimentar con los tiempos de activación para lograr disminuir el parpadeo y que el efecto sea el deseado. Veamos un ejemplo:

PORTB = mascara

Llevamos el dato obtenido de la tabla al puerto de salida, en nuestro caso es el B.

RA.0 = 1

Activamos el primer transistor.

WaitMs 5

Esperamos un tiempo, este es el valor con el cual se deberá experimentar.

RA.0 = 0

Apagamos el primer transistor.

Esta rutina la deberemos realizar cuatro veces, una por cada display a encender.

Funcionamiento del prototipo.

El equipo cuenta con una entrada para anexarle cualquier tipo de sensor, previo acondicionamiento de la señal en caso de ser indispensable. Para que se produzca un incremento en la cuenta es necesario un cambio de nivel por flanco ascendente (de 0V a 5V) en la entrada.

En estado de espera, ausencia de señal de conteo, se exhibirá en los displays la última cifra.

Para reiniciar el funcionamiento, es decir setear la cuenta en 0000 es necesario pulsar el botón "Reset" manualmente.

Si se llegará a lograr superar el límite de cuenta, 9999 eventos, automáticamente se dejará de monitorizar el pin de entrada y se exhibirá en los displays el símbolo "----", quedando a la espera de un reinicio manual por parte del usuario.

Conclusiones.

Estudiando no muchas instrucciones Basic podremos empezar a experimentar con un microcontrolador de rango medio provisto de varios periféricos logrando armar prototipos de mediana complejidad.

No es muy difícil añadirle prestaciones al proyecto aquí presentado. Espero dejar la puerta abierta a inquietudes y ganas de modificarlo para aprender y superar objetivos.

Referencias.

[1] LM78XX/78XXA 3-Terminal 1A Positive Voltage Regulator
www.fairchildsemi.com/ds/lm/LM7805.pdf

[2] PIC16F627A/628A/648A FLASH-Based 8-Bit CMOS Data Sheet
<http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>

[3] PIC SIMULATOR IDE
<http://www.oshonsoft.com/pic.html>

[4] Ariel Palazzesi - Aprendiendo a programar PICs en BASIC
[http://www.ucontrol.com.ar/wiki/index.php/PIC_BASIC_\(PSI\)](http://www.ucontrol.com.ar/wiki/index.php/PIC_BASIC_(PSI))