Autor: Pedro I. López
Contacto: dreilopz@gmail.com | www.dreilopz.me

Fecha: Febrero 2012.

Práctica 11

Contadores por hardware

Objetivo

Que el estudiante comprenda la importancia de los contadores por hardware, conozca algunas de sus aplicaciones y logre implementar sus propios dispositivos.

Introducción

Para esta práctica se configurará un microcontrolador para que trabaje como un contador. Aunque la práctica se llama contadores por hardware, el contador implementado aquí es por software. Aún así el experimento se lleva a cabo.

Desarrollo

Implementación de un contador de pulsos digitales

A lo largo de esta sección el estudiante conocerá el funcionamiento de un contador por hardware y desarrollara un contador simple de alta velocidad, esta practica tiene una gran importancia debido a que se realiza e implementa un contador a partir de componentes fáciles de conseguir.

Material a utilizar:

- Modulo PIC 16F84A.
- NI ELVIS.
- PIC 16F84.
- DAQ 6.

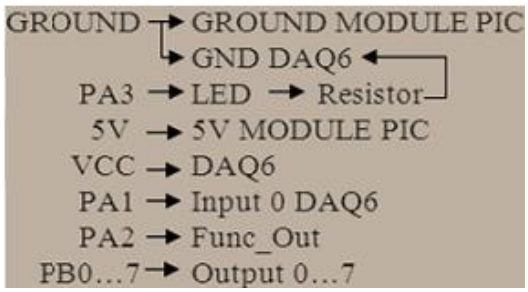
La DAQ 6 es una tarjeta para lectura y escritura de puertos digitales, será empleada solamente para controlar la GATE del contador y para leer la OUT del mismo. Se analizó el programa del documento de la práctica para poder comprenderlo a detalle, y posteriormente realizar una recodificación del mismo con compilador ***Great Cow Basic***.

El código resultante es el siguiente:

```
E '=============================================================
'===== CHIP SETUP
'=============================================================
#chip 16F84A, 4
#config _CP_OFF & _PWRTE_ON & _WDT_OFF & _HS_OSC

#define StartUpDelay 10 ms

'=============================================================
'===== variables
```

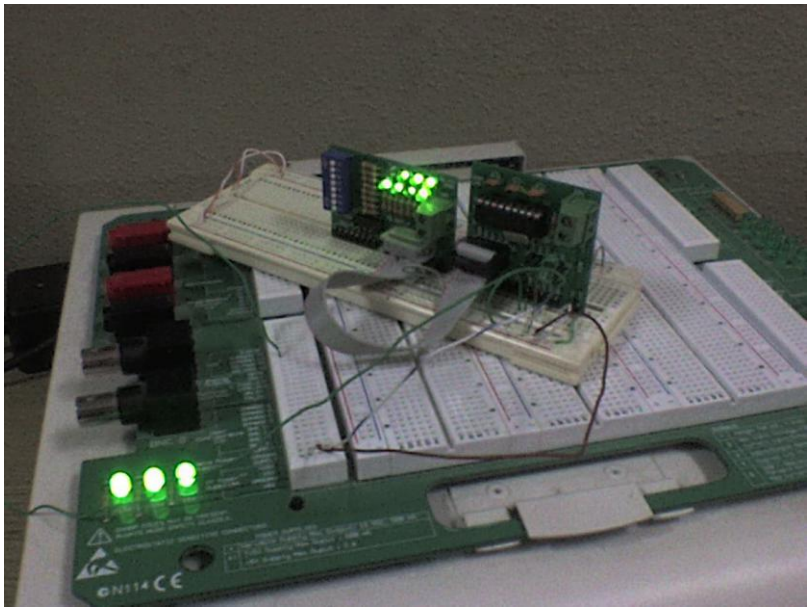
```
'================================================================
dim COUNTREG as byte
dim SAMPLE as byte
dim LATCH as byte

'================================================================
'===== definiciones
'================================================================

#define GATE PORTA.0
#define SOURCE PORTA.1
#define out0 PORTA.2
#define led PORTA.3

'================================================================
'===== SET DIRECTION OF THE PORTS
'================================================================

dir GATE in
dir SOURCE in
dir out0 out
dir led out
dir PORTB out

'================================================================
'===== MAIN PROGRAM LOOP
'================================================================

COUNTREG=0
out0=0
SAMPLE=0
LATCH=0

set led on
wait 1 s
set led off
wait 1 s


Main:

   PORTB=COUNTREG

   if COUNTREG=255 then
          out0=1
          wait 1 ms
          COUNTREG=0
          out0=0
   end if

   if GATE=1 then
          SAMPLE=SOURCE

          if LATCH=SAMPLE then
                 goto main
          end if

          '
          if LATCH=0 then
                 LATCH=1
                 COUNTREG=COUNTREG+1
                 goto main
          end if

          if LATCH=1 then
                 LATCH=0
                 goto main
          end if
          '

   end if
Goto Main
```

Se compila y ensambla el código, y luego se programa el microcontrolador. Estamos listos ahora para realizar las conexiones de acuerdo a *f11-1*.

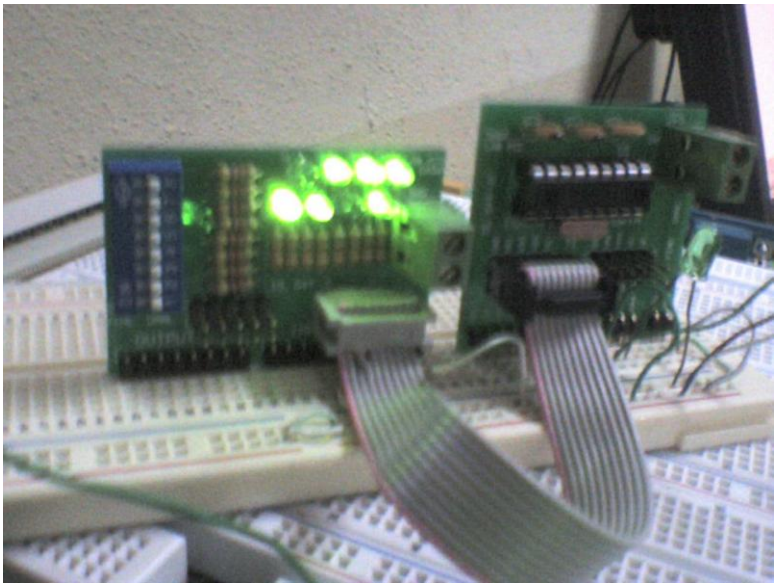


f11- 1. Coneciones para implementación

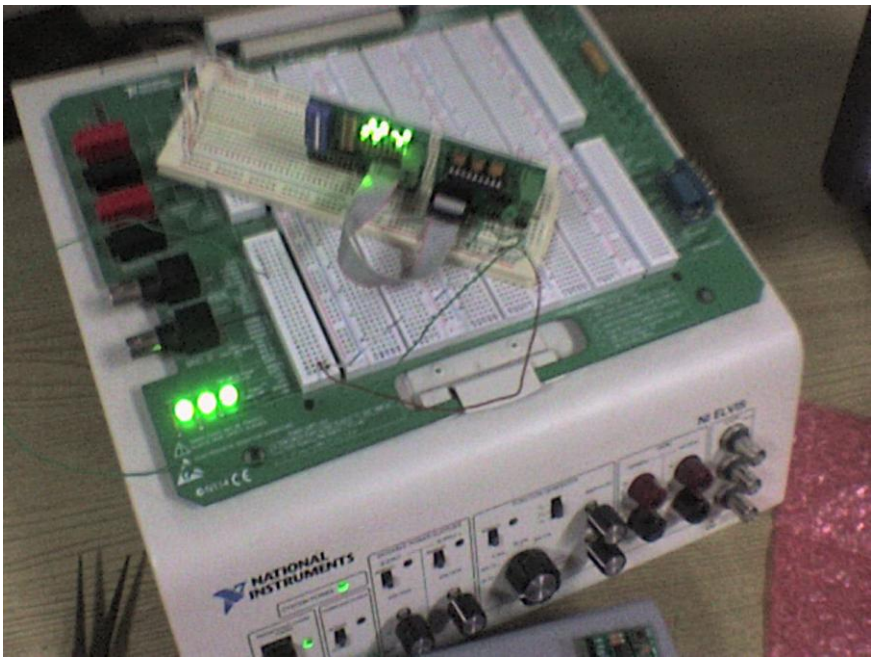
Incluimos imágenes de la implementación del sistema.



f11- 2. Fotografía de implementación



f11- 3. Fotografía de implementación



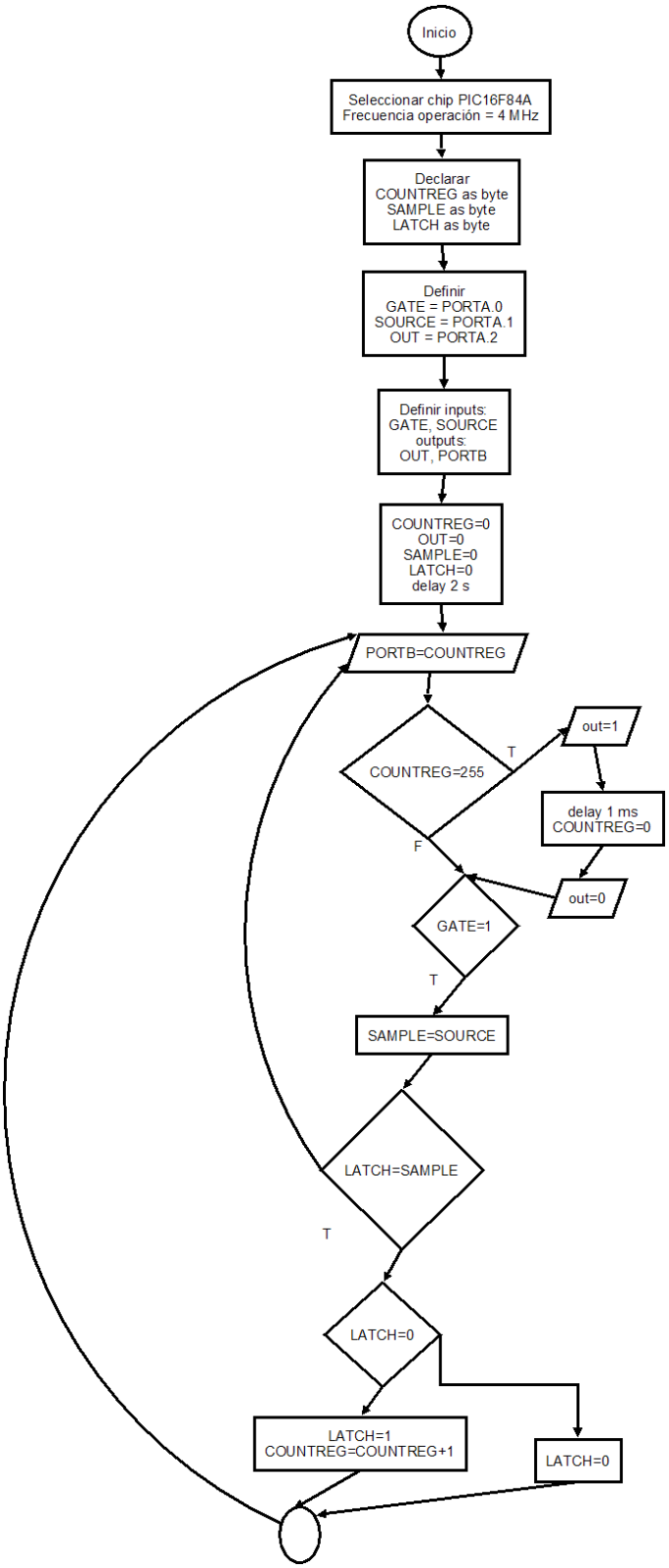
f11- 4. Fotografía de implementación

Reporte

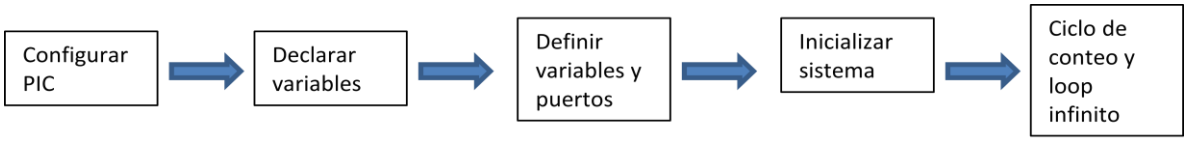
Identificar los parámetros del contador por hardware desarrollado (GATE, SOURCE, OUT, COUNT REGISTER), si falta alguno mencionarlo.

- **Gate**: pin 0 de PORTA, habilita el conteo (tiene que estar en 1, es entrada).
- **Source**: pin 1 de PORTA, recibe los pulsos para ser contador por el programa del microcontrolador
- **Out**: es salida, un indicador lógico temporal que avisa cuando el contador se ha desbordado y comienza un conteo desde cero, en este caso es el pin 2 de PORTA.
- **Count Register**: registro en RAM destinado a almacenar el conteo constante. En nuestro caso utilizamos dos registros para Count Register, un registro general de RAM para almacenarlo y luego ser cargado al registro de control de PORTB para mostrarlo como salida con la tarjeta de entradas/salidas digitales.
- **Source Edge**: no está implementado.
- **Event Source/Timebase**: selecciona si se quieren contar evento o tiempo, no está implementado en el programa.
- **Counter size**: el registro de conteo tiene una resolución de 8 bits (contará del 0 a 255 en decimal).
- **Start/Restart, Stop**: activan o desactivan el conteo, no fue implementado propiamente en el sistema, aunque el pin MCLR o apagar la fuente pueden ser considerados que realicen tal función.

Realizar el diagrama de bloques y de flujo para la programación utilizada



f11- 5. Diagrama de flujo de programa PIC



f11- 6. Diagrama de bloques de programa PIC, ver diagrama de flujo para detalles.

[texto borrado por Copyright]

Se anexa un código para microcontrolador PIC18F4550 que se utilizó para hacer un tacómetro digital, contará pulsos con uno de sus pines que habilitan interrupciones, y utiliza el enfoque anterior explicado.

```
'TACOMETRO||||||||||||||||||||||||||||||||||||
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'-----------------------------------------------

'CONFIGURATION bitS|||||||||||||||||||||||||
Define CONFIG1L = %00000100
Define CONFIG1H = %00001100
Define CONFIG2L = 0x3f
Define CONFIG2H = 0x1e
Define CONFIG3H = 0x81
Define CONFIG4L = 0x81
Define CONFIG5L = 0x0f
Define CONFIG5H = 0xc0
Define CONFIG6L = 0x0f
Define CONFIG6H = 0xa0
Define CONFIG7L = 0x0f
'||||||||||||||||||||||||||||||||||||||||||||

Define SIMULATION_WAITMS_VALUE = 1  'tiempos de simulacion
Define CLOCK_FREQUENCY = 20

AllDigital
Config PORTD = Output
Config PORTB.0 = Input
Config PORTB.1 = Output
Define LCD_BITS = 4
Define LCD_DREG = PORTD
Define LCD_DBIT = 4
Define LCD_RSREG = PORTD
Define LCD_RSBIT = 3
Define LCD_EREG = PORTD
Define LCD_EBIT = 1
Define LCD_RWREG = PORTD
Define LCD_RWBIT = 2
Define LCD_READ_BUSY_FLAG = 1

Dim muestreo As Byte
Dim velocidad As Word
Dim pos As Byte

PORTD.0 = True
Enable High
RCON.IPEN = True
INTCON.GIEH = True
INTCON.GIEL = False
INTCON2.TMR0IP = True
INTCON.TMR0IF = False
INTCON.TMR0IE = True
PIE1.TMR1IE = False

T0CON.T08BIT = False
T0CON.T0CS = False
T0CON.T0SE = True
T0CON.PSA = False
T0CON.T0PS2 = False
T0CON.T0PS1 = True
T0CON.T0PS0 = False
T0CON.TMR0ON = False

T1CON.RD16 = True
T1CON.T1RUN = False
```

```
T1CON.T1CKPS1 = False
T1CON.T1CKPS0 = False
T1CON.T1OSCEN = False
T1CON.T1SYNC = False
T1CON.TMR1CS = True
T1CON.TMR1ON = False

'Lcdinit 2
'Lcddefchar 0, %00000, %00000, %00000, %00000, %00000, %00000, %00000, %00000
'Lcddefchar 1, %11111, %11111, %11111, %11111, %11111, %11111, %11111, %11111
'Lcdcmdout LcdClear
''Lcdcmdout LcdHome
'Lcdout "Bienvenido ---->"

For pos = 1 To 16 Step 1
   'Lcdcmdout LcdLine2Pos(1)
   'Lcdout "                "
   'Lcdcmdout LcdLine2Pos(pos)
   'Lcdout 1
   'WaitMs 50
Next pos

For pos = 16 To 1 Step -1
   'Lcdcmdout LcdLine2Pos(1)
   'Lcdout "                "
   'Lcdcmdout LcdLine2Pos(pos)
   'Lcdout 1
   'WaitMs 50
Next pos

'Lcdcmdout LcdLine2Pos(1)
'Lcdout 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
WaitMs 333
'Lcdcmdout LcdLine2Pos(1)
'Lcdout 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
WaitMs 333
'Lcdcmdout LcdLine2Pos(1)
'Lcdout 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
WaitMs 1000

'Lcdcmdout LcdClear
'Lcdcmdout LcdHome
'Lcdout "   Tacometro    "
'Lcdcmdout LcdLine2Pos(1)
'Lcdout "    con LCD     "
WaitMs 1000
'Lcdcmdout LcdClear
'Lcdcmdout LcdHome
'Lcdout "Comenzar a medir   "
'Lcdcmdout LcdLine2Pos(1)
'Lcdout "velocidad angular   "
WaitMs 1000
'Lcdcmdout LcdShiftLeft
WaitMs 1000

TMR1H = 0x00
TMR1L = 0x00
T1CON.TMR1ON = True

TMR0H = 0x00
TMR0L = 0x00
T0CON.TMR0ON = True

muestreo = 0
'Lcdcmdout LcdClear
'Lcdcmdout LcdHome
'Lcdout "Velocidad"

mainloop:
   velocidad = muestreo * 120
          If PORTD.0 Then
                 Goto mainloop
          Else
                 'Lcdcmdout LcdLine2Pos(1)
                 'Lcdout #velocidad, " RPM         "
                 WaitMs 250
                 Goto mainloop
          Endif
End

interrupcion:
```

```
    muestreo = TMR1L
    TMR1H = 0x00
    TMR1L = 0x00
    TMR0H = 0x00
    TMR0L = 0x00
Return

On High Interrupt
    Save System
    T1CON.TMR1ON = False
    Gosub interrupcion
    T1CON.TMR1ON = True
    INTCON.TMR0IF = False
Resume
```

¿Cuáles son las condiciones necesarias para que el PIC entregue información equivocada en este experimento?
Algo que haría que el experimento entregue información incorrecta, es una condición de fallo de hecho para cualquier otro contador. Esto pasará cuando la frecuencia de pulsos de entrada que llegan por Source sea más rápida que la velocidad de la máquina al repetir su programa principal, preguntándose si Latch = Sample (revisar comentarios en programa ejemplo de documento de laboratorio, página 151). Esto está directamente relacionado con la frecuencia de operación del PIC (la cual es 4 MHz en nuestro caso) y evidentemente el error puede evitarse en mayor medida utilizando un reloj más veloz, o cambiando todo el sistema para que sea un contador por hardware.

Conclusión

Un contador por hardware es una característica muy útil en sistemas de adquisición de datos. Frecuentemente es una de las características que uno busca evaluar a la hora de seleccionar algún dispositivo para implementar en proyectos (microcontroladores), ver cuantos hay, que resolución tienen, si se puede escribir sobre ellos y leer directamente los registros de conteo, y más importante, si son capaces de generar interrupciones.

El hecho de realizar conteos por hardware es muy útil porque entonces la computadora no necesita dedicarle mucho tiempo a tal proceso, ya que se encuentra operando con elementos de hardware, y cuando suceda algo importante, la CPU puede atender el proceso al ser llamada con una interrupción. Así, los contadores nos permiten diseñar medidores de frecuencias, de periodo, de eventos, tacómetros, contadores individuales, PWMs, etcétera.

Durante el desarrollo de esta práctica se revisaron los conceptos teóricos de un contador por hardware. Se analizaron las condiciones de operación, medidas y límites. Para la sección de desarrollo, se utilizó el módulo PIC y la tarjeta de entradas y salidas digitales para implementar un contador por software, y realizar analogías en comparación con la teoría. Cabe menciones que el microcontrolador utilizado en esta práctica cuenta de hecho con 2 timers, el clásico WDT para monitorear la operación del PIC, y TMR0 para propósitos generales (TMR0 es un timer/counter configurable de 8 bits, que puede generar interrupciones).

Bibliografía

- **Documento de práctica de laboratorio**
  Práctica 11 – Contadores por hardware
  Laboratorio de Adquisición de Datos
  FIME – UANL

- **PIC16F84A – Data Sheet**
  Microchip
  2001

- **PIC18F4550 – Data Sheet**
  Microchip
  2004

- **Sensores y acondicionadores de señal**
  Ramón Pallás Areny
  4ª edición
  Alfaomega-Marcombo