NetMedia

Basic Express BX-35
Application Note

Counting Pulses With Hardware Interrupts

Introduction

This application note illustrates how to use hardware interrupts to count pulses.

System call WaitForInterrupt

WaitForInterrupt allows BasicX programs to respond quickly to hardware interrupts on an I/O pin. In the following example program, task PulseCountTask runs in a continuous loop calling WaitForInterrupt. The task blocks until a rising edge appears on the INT1 pin, which is pin 17 on a BX-35.

When the interrupt occurs, counter PulseCount is incremented, and an LED is pulsed in order to provide visual feedback.

Another task in the main program converts the pulse count to a string and transmits it through the Com1 serial port. The pulse count is transmitted once per second. Example code:

```
Private Const StackSize As Integer = 25
Private PulseCountStack(1 To StackSize) As Byte
Private PulseCount As Integer

Public Sub Main()

    PulseCount = 0

    Debug.Print
    Debug.Print "Pulse counting demonstration"
    Debug.Print

    CallTask "PulseCountTask", PulseCountStack

    Do
        Debug.Print CStr(PulseCount)
        Call Delay(1.0)
    Loop

End Sub
```

```
Private Sub PulseCountTask()

    Const LEDpin As Byte = 20
    Const LEDon  As Byte = 1
    Const LEDoff As Byte = 0

    Do
        Call WaitForInterrupt(bxPinRisingEdge)
        PulseCount = PulseCount + 1

        Call PutPin(LEDpin, LEDon)

        ' Debounce.
        Call Sleep(0.1)

        Call PutPin(LEDpin, LEDoff)
    Loop

End Sub
```

Although the pulse count is transmitted at a slow rate as a sort of foreground task, the PulseCountTask, runnning in the background, is able to respond much more quickly to a rising edge on the interrupt pin.

Note that the WaitForInterrupt parameter can specify 3 different types of interrupt triggers -- falling edge, rising edge or logic low. The operating system supplies the predefined enumerations bxPinFallingEdge, bxPinRisingEdge and bxPinLow for these parameters.

Caution – you should insure that pin INT0 is held high. Otherwise unwanted interrupts can interfere with the operation of the program. On a BX-35, INT0 is pin 16.

This source code file the accompanies this application note is called CountingPulses.bas.

2.00.A