

```
'-----Title-----
'
' File.....switch2.pbp
' Started....6/4/05
' Microcontroller used:  Microchip Technology 16F88
'                          microchip.com
' PicBasic Pro Code:  micro-Engineering Labs, Inc.
'                          melabs.com

'-----Program Description-----

' Switch drives LED and servo

'-----Related Lesson-----

' switch2.pbp is used in the lesson ACTIVE HIGH ACTIVE LOW at:
' http://www.cornerstonerobotics.
org/curriculum/lessons_year2/erii19_active_high_active_low.pdf

'-----Comments-----

' WITH THE PIC16F88, MAKE SURE TO HAVE TWO SEPARATE +5VDC
' POWER SUPPLIES FOR THE PIC AND THE SERVO.  ALSO
' HAVE A COMMON GROUND BETWEEN THE POWER SUPPLIES.
' We use one 9V battery and two 78L05 voltage regulators.
' See discussion about voltage regulators at:
' http://cornerstonerobotics.
org/curriculum/lessons_year2/erii3_diodes_power_supplies_voltage_reg.pdf

' Also, initialize the state of PORTB as LOW
' since that will set the correct polarity of the
' PULSOUT statement.  See PULSOUT in PicBasic Pro
' Compiler manual by microEngineering Labs, Inc.
' Look around page 121 in the PicBasic Pro Compiler Manual

'-----Connections-----

'      16F88 Pin          Wiring
'      -----          -
'      RA0                Momentary Switch
'      RB2                Servo Control Wire (Futaba uses white)
'      RB0                LED

'-----Revision History-----

' 3/2/06  Added comments
' 1/14/08 Change MCU from 16F84A to 16F88
' 1/14/08 Add 16F88 oscillator initialization

'-----Variables-----

c0      VAR      BYTE      ' Byte for counter
switch  VAR      PORTA.0    ' Pin RA0 is assigned the name switch
' using the VAR command.
```

## '-----Initialization-----

```

TRISA = %11111111      ' All PORTA pins are setup as inputs.
                       ' This can also be written TRISA = 255
TRISB = %11111010      ' Pins RB0 and RB2 are set as outputs,
                       ' all other PORTB pins are set as inputs

PORTB = %00000000      ' Sets all PORTB pins to LOW(0 volts)
                       ' Make certain to include this
                       ' initialization for the servo output
                       ' pin as it sets the proper polarity
                       ' of pulses in the PULSOUT command.

OSCCON = $60           ' Sets the internal oscillator in the
                       ' 16F88 to 4 MHz

ANSEL = %00000000      ' Changes analog bits to digital.
                       ' See table below.

```

Analog Bit	Analog or Digital	PIC16F88 Pin
AN0	Digital	RA0
AN1	Digital	RA1
AN2	Digital	RA2
AN3	Digital	RA3
AN4	Digital	RA4
AN5	Digital	RB5
AN6	Digital	RB6

## '-----Main Code-----

```

start:                ' start label

IF switch = 0 THEN  ' If the momentary switch is pressed, pin
                       ' RA0 goes from HIGH(+5 vdc) to LOW(0 vdc).
                       ' The comparison is true and the
                       ' THEN statement is executed.
                       ' If the switch is not pressed, the
                       ' comparison is false and the ELSE
                       ' statement is executed.

FOR c0 = 1 TO 10    ' Send left servo signal 10 times

HIGH 0              ' LED connected to RB0 is turned on

PULSOUT 2,100      ' Pulse Width:
                       ' Sends a pulse out on pin RB2 for 1.0 msec.
                       ' The period,(100), is multiplied by the
                       ' increment for a 4 MHz oscillator(10 usec)
                       ' to get a pulse out time of 1.0 msec.
                       ' To get the full range of your
                       ' servo, you may have to decrease the period
                       ' to less than 100, being careful not to go too
                       ' low thereby forcing the servo to stop

```

```

' mechanically against the internal
' mechanical stops.

PAUSE 19      ' Pulse Interval:
              ' Wait 19 msec before sending next pulse

NEXT c0      ' Goes to next value of c0

ELSE        ' If the momentary switch connected to RA0
            ' is not pressed and RA0 remains HIGH(+5 vdc).
            ' The comparison is false and
            ' the ELSE statement is executed.

FOR c0 = 1 TO 10  ' Send right servo signal 10 times

LOW 0        ' LED connected to RB0 is turned off

PULSOUT 2,200  ' Pulse Width:
              ' Sends a pulse out on pin B2 for 2.0 msec.
              ' The period,(200), is multiplied by the
              ' increment for a 4 MHz oscillator(10 usec)
              ' to get a pulse out time of 2.0 msec.
              ' To get the full range of your
              ' servo, you may have to increase the period
              ' to more than 200, being careful not to go too
              ' high thereby forcing the servo to stop
              ' mechanically against the internal
              ' mechanical stops.

PAUSE 18      ' Pulse Interval:
              ' Wait 18 msec before sending next pulse

NEXT c0      ' Goes to next value of c0

ENDIF       ' End of IF..THEN statement

GOTO start  ' Loop to start label

END
```