```
'--------Title--------

' File......switch_limit1.pbp
' Started....2/4/08
' Microcontroller used:  Microchip Technology 16F88
'                        microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
'                    melabs.com

'--------Program Desciption--------

' Program runs a 2 rpm motor forward until it
' triggers the forward limit switch, at which point
' the motor pauses 1 second before reversing
' direction until it triggers the back limit switch,
' causing it to return to the forward direction.
' The 2 rpm gearhead motor is the GH35GM series
' from Jameco.

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

' Be certain that the forward limit switch is
' activated when the motor is in the forward mode,
' not the backward mode.

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

' 3/17/10 Cleaned up code and comments

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

    pwm_motor1      VAR PORTB.2 ' Labels PORTB.2 as pwm_motor1
    dx_motor1       VAR PORTB.3 ' Labels PORTB.3 as pwm_motor1
    forward_switch  VAR PORTB.4 ' Labels PORTB.4 as forward_switch
    backward_switch VAR PORTB.5 ' Labels PORTB.5 as backward_switch

'---------Initialization--------

    TRISB = %11110000       ' Sets up pins RB0-RB3 as an outputs
                            ' and pins RB4-RB7 of PORTB as inputs

    ANSEL = 0               ' Configure all pins to digital
                            ' operation since not using ADC
                            ' (Analog to Digital Converter)

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

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

main:                   ' Main loop

    GOSUB forward       ' Jump to forward subroutine
    GOSUB backward      ' Jump to backward subroutine
```

```
    GOTO main             ' Jump to main loop

    END

forward:                  ' forward subroutine

    pwm_motor1 = 1        ' PWM = 100% duty cycle
    dx_motor1 = 1         ' Motor direction is forward

    IF forward_switch = 1 THEN
                          ' If the forward limit switch is
                          ' pressed, RB4 goes HIGH and the program
                          ' makes the PWM = 0% (turning the motor
                          ' off). The program pauses for 1 second.
        pwm_motor1 = 0    ' PWM = 0% duty cycle, turning
                          ' off the motor
        PAUSE 1000        ' Pause 1 second to allow the motor to
                          ' coast to a stop before changing direction.
                          ' Be careful that the coast is not so
                          ' long that the motor mechanism
                          ' damages the switch lever.

    ELSE

       GOTO forward       ' If the forward limit switch is not
                          ' pressed, the motor continues to go
                          ' forward.

    ENDIF
    RETURN                ' Return to statement after
                          ' GOSUB forward statement.

backward:                 ' backward subroutine

    pwm_motor1 = 1        ' PWM = 100% duty cycle
    dx_motor1 = 0         ' Motor direction is reverse

    IF backward_switch = 1 THEN
                          ' If the backward limit switch is
                          ' pressed, RB5 goes HIGH and the program
                          ' makes the PWM = 0% (turning the motor
                          ' off). The program pauses for 1 second.
        pwm_motor1 = 0    ' PWM = 0% duty cycle, turning
                          ' off the motor
        PAUSE 1000        ' Pause 1 second to allow the motor to
                          ' coast to a stop before changing direction.


    ELSE

        GOTO backward     ' If the backward limit switch is not
                          ' pressed, the motor continues to go
                          ' backward.

    ENDIF
```

```
RETURN                  ' Return to statement after
                        ' GOSUB backward statement.
```