

'-----Title-----'

' File.....4331\_encoder3.pbp  
' Started....10/28/09

' Microcontroller Used: Microchip Technology 18F4331  
' Available at:  
' <http://www.microchipdirect.com/ProductDetails.aspx?Category=PIC18F4331>  
' or <http://www.digikey.com/>  
' Motor Controller Used: Xavien 2 Motor Driver "XDDCMD-1  
' Available at: [http://encodergeek.com/Xavien\\_Amplifier.html](http://encodergeek.com/Xavien_Amplifier.html)  
' Motor and Encoder Used: Small Motor with Quadrature Incremental Encoder  
' Available at: [http://encodergeek.com/DCMtr\\_SMALL.html](http://encodergeek.com/DCMtr_SMALL.html)  
'  
' PicBasic Pro Code: micro-Engineering Labs, Inc.  
' [melabs.com](http://melabs.com)

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

' Program slows motor as it approaches target position  
' (diff = 0). Motor power is no longer controlled by  
' potentiometer wired to AN0, but by the difference (diff)  
' between the position and target values. If the motor  
' overshoots the target, the motor will change the direction  
' of rotation and return to the target position.

'---Review PicBasic Pro Command---

' The PicBasic Pro Compiler Manual is on line at:  
' <http://www.microengineeringlabs.com/resources/index.htm#Manuals>  
'  
' HPWM Channel,Dutycycle,Frequency  
'  
' Outputs a PWM signal using the PICs hardware which  
' is available on some PICs including the PIC18G4331.  
' Channel specifies which PWM channel to use.  
' Dutycycle ranges from 0 (0%) to 255 (100%).  
' Frequency - lowest frequency depends upon oscillator speed,  
' highest frequency at any oscillator speed is 32,767 Hz.  
' Look around page 75 in the PicBasic Pro Compiler Manual  
' for detailed discussion of the HPWM command.

'-----PIC Connections-----'

| 18F4331 Pin | Wiring   |
|-------------|--|
| RA0(AN0)    | Potentiometer, controls motor power                              |
| RA3         | Signal 1 from Encoder  |
| RA4         | Signal 2 from Encoder  |
| RB5         | In Circuit Serial Programming (ICSP) PGM<br>100K Resistor to GND |
| RB6         | ICSP PGC (Clock)   |
| RB7         | ICSP PGD (Data)  |
| RC0         | Brake Motor 1 on Xavien XDDCMD-1 (Pin 1)                         |
| RC1         | PWM Motor 1 on Xavien XDDCMD-1 (Pin 2)                           |

```
'
    RC3                Direction Motor 1 on Xavien XDDCMD-1 (Pin 3)
    RD4                LCD Data Bit 4
    RD5                LCD Data Bit 5
    RD6                LCD Data Bit 6
    RD7                LCD Data Bit 7
    RE0                LCD Register Select
    RE1                LCD Enable
    MCLR               4.7K Resistor to +5V & ICSP Vpp
    VDD                +5V
    VSS                GND
    OSC1 & OSC2       4 MHz Crystal w/ 2-22 pF Cap. to GND
```

'----Xavien XDDCMD-1 Connections----

| Xavien 2x5 Header Pin   | Wiring | Pin Layout | 2x5 Header |
|-------------------------|--------|------------|------------|
| -----                   | -----  | -----      | -----      |
|                         |        | 2 4 6 8 10 |            |
| Pin 1 Motor 1 Brake     | RC0    | o o o o o  |            |
| Pin 2 Motor 1 PWM       | RC1    | o o o o o  |            |
| Pin 3 Motor 1 Direction | RC3    | 1 3 5 7 9  |            |

' See schematic at:  
'[http://cornerstonerobotics.org/schematics/18f4331\\_hpwm\\_motor\\_encoder.pdf](http://cornerstonerobotics.org/schematics/18f4331_hpwm_motor_encoder.pdf)

'--Sample POSCNTH, POSCNTL Values and Corresponding Position Counter--

' position = 256 \* POSCNTH + POSCNTL

| POSCNTH | POSCNTL | Position Counter |
|---------|---------|------------------|
| -----   | -----   | -----            |
| 0       | 0       | 0                |
| 0       | 1       | 1                |
| 1       | 0       | 255              |
| 0       | 128     | 128              |
| 128     | 0       | 32768            |
| 0       | 255     | 255              |
| 255     | 0       | 65280            |
| 255     | 255     | 65535            |

'-----Defines-----

```
DEFINE LCD_DREG PORTD      ' Set LCD Data port
DEFINE LCD_DBIT 4          ' Set starting Data bit to 4
DEFINE LCD_BITS 4         ' Set LCD bus size to 4
DEFINE LCD_RSREG PORTE    ' Set LCD Register Select port to E
DEFINE LCD_RSBIT 0        ' Set LCD Register Select bit to 0
DEFINE LCD_EREG PORTE     ' Set LCD Enable port to E
DEFINE LCD_EBIT 1         ' Set LCD Enable bit to 1
DEFINE LCD_LINES 2        ' Set number of lines on LCD to 2
DEFINE LCD_COMMANDUS 2000 ' Set command delay time to 2000 us
DEFINE LCD_DATAUS 50      ' Set data delay time to 50 us
DEFINE ADC_BITS 8         ' Set number of bits in result to 8
DEFINE ADC_CLOCK 3        ' Set clock source (rc = 3)
DEFINE ADC_SAMPLEUS 50    ' Set sampling time in us
DEFINE CCP2_REG PORTC     ' Set HPWM Channel 2 port to C
```

```

DEFINE CCP2_BIT 1          ' Set HPWM Channel 2 bit to 1

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

mot_pwr  VAR    BYTE      ' Declare mot_pwr variable, reserve byte
position VAR    WORD      ' Declare position, reserve word
target   VAR    WORD      ' Declare target, reserve word
diff     VAR    WORD      ' Declare diff, reserve word

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

CCP1CON = %00111111        ' Set Capture/Compare/PWM Module Control
                               ' Register CCP1CON in PWM mode (bits 0-3),
                               ' bits 4,5 set LSBs of 10-bit duty cycle,
                               ' see 18F4331 datasheet page 151 +/-.
ANSEL0  = %00000001        ' Set AN0 to analog, AN1-AN7 to digital,
                               ' see datasheet page 249 +/-.
ANSEL1  = %00000000        ' Set AN8 to digital, see datasheet
                               ' page 249 +/-.
TRISA   = %00011111        ' Set TRISA register, RA7-RA5 as outputs,
                               ' RA4-RA0 as inputs, see datasheet
                               ' page 107 +/-.
LATA    = %00000000        ' Set all LATA register bits to 0.
TRISB   = %00000000        ' Set RB7-RB0 pins in PORTB as outputs.
TRISC   = %00000000        ' Set RC7-RC0 pins in PORTC as outputs.
QEICON  = %10001000        ' Set Quadrature Encoder Interface Control
                               ' Register. See page 171 +/- for
                               ' encoder set up.
PORTC.0 = 1                ' Turn on brake.
PORTC.1 = 0                ' Set PWM bit for Channel 2 of HPWM to LOW.

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

PAUSE 500                  ' Start up LCD
target = 33400              ' Set target position

' Set counter starting position:

POSCNTH = 127              ' Set counter for encoder, H bit
POSCNTL = 0                ' Set counter for encoder, L bit
                               ' With POSCNTH = 127 and POSCNTL = 0,
                               ' position counter will start at 32512.
                               ' See table above for more sample values.

loop:

    position = 256 * POSCNTH + POSCNTL    ' Read position

' Set motor direction:

IF target < position THEN ' Routine to set correct motor direction.

    PORTC.3 = 1              ' Set motor direction, you may have to flip
                               ' motor directions for position to converge
                               ' on target, that is, PORTC.3 = 0 here.

ELSE
```

```

PORTC.3 = 0          ' Set motor direction, you may have to flip
                    ' motor directions for position to converge
                    ' on target, that is, PORTC.3 = 1 here.

ENDIF

' Calculate difference:

IF target >= position THEN ' Use IF..THEN to get positive value of
diff
diff = target - position
ELSE
diff = position - target
ENDIF

' Control motor speed and brake:

SELECT CASE diff          ' Use SELECT CASE statement to compare the
range                    ' variable diff with the value (0), the
                        ' value > 201, and range value <= 201.
CASE IS = 0              ' If diff = 0, arrived at target,
                        ' activate brake.
        PORTC.0 = 1      ' Turn on brake
        GOSUB lcd        ' Go to lcd subroutine
CASE IS > 201            ' If diff > 201, full motor power.
        PORTC.0 = 0      ' Turn off brake
        mot_pwr = 255    ' Full motor power = 255
        GOSUB lcd        ' Go to lcd subroutine
CASE IS <= 200          ' If diff <= 200, slow motor as position
                        ' approaches target.
        PORTC.0 = 0      ' Turn off brake
        mot_pwr = diff * 9/10 + 75
                        ' Motor power (mot_pwr) is reduced as diff
                        ' becomes smaller. The number 75 is about
                        ' the lowest HPWM Duty cycle before the
                        ' motor stalls. If diff = 200, mot_pwr =
255,
                        ' if diff = 1, mot_pwr = 76.
        GOSUB lcd        ' Go to lcd subroutine
END SELECT

GOTO loop                ' Return to loop
END

' Subroutine:

lcd:

HPWM 2, mot_pwr, 20000   ' Send PWM signal from RC1 to Pin 2 on
                        ' the Xavien XDDCMD-1 DC motor driver.
LCDOUT $FE, $80, "Pwr=",DEC3 mot_pwr, " Df=",DEC5 diff
                        ' On the first line, display mot_pwr value
                        ' in 3 decimal digits and diff value in 5
                        ' decimal digits.
LCDOUT $FE, $C0, "T=",DEC5 target, " P=", DEC5 position

```

**RETURN**

' On the second line, display target in 5  
' decimal digits and position in 5  
' decimal digits.  
' Go back to main routine that called us.