INTRODUCTION

This application note will describe how to drive a bipolar stepping motor with the PIC16F684. The PIC16F684 has an ideal set of peripherals for driving a stepper motor. These peripherals include two on-chip comparators and an Enhanced Capture Compare PWM (ECCP) module. The ECCP module is used to microstep the motor while the on-chip comparators limit the current in the windings of the motor.

Note: Please refer to AN907: Stepping Motor Fundamentals for information on the types of stepper motors, microstepping and current limiting techniques.

FIGURE 1: HIGH TORQUE MICROSTEPPING

A brief description of what is happening is that one winding is powered while the current in the other winding is gradually dropped to zero, reversed, and then ramped up again. This sequence is then repeated for the other winding. Note that the transition between a winding being energized in one direction and then energized in the other direction has a sinusoidal shape (refer to Figure 1). This shape gives the smoothest transition between the motor’s rated step increments (i.e., 7.5 degrees). The way this shape is achieved using a microcontroller is through the use of pulse-width modulation. Modulating the input to the drive circuitry for a particular winding will result in a current that is proportional to the duty cycle of the modulated waveform.

For instance, if a 5V stepping motor is rated at 1 amp, then modulating a 5V supply across the winding at 50% will result in a current of 1/2 amp. Equation 1 shows this relationship:

\[ I = D \times I_{\text{MAX}} \]

where \( I_{\text{MAX}} \) is the rated current of the motor and \( D \) is the duty cycle.

MICROSTEPPING

Single stepping, or turning a stepping motor at its rated step size, results in jerky movement. Microstepping is a technique used to smooth the motor’s movement between full steps and to improve the step resolution of the motor. Microstepping also improves the efficiency of the system because the current in the windings of the motor is manipulated in a controlled manner rather than being turned on and off abruptly.

A microstepping technique known as high torque microstepping alternately varies the current in the two windings of a stepping motor. Figure 1 shows a graph of the current in the windings vs. angular position using this technique.
In order to achieve the sinusoidal transition from a positive to negative charge in a winding, numerous microsteps are needed. The number of microsteps typically ranges from 4 to 32 microsteps per rated step size. Rather than calculating the duty cycle for a particular microstep on the fly, a duty cycle look-up table is implemented in firmware. The number of table values is equal to the number of steps desired for a particular microstepping sequence. Equation 2 is used to obtain the duty cycle values for the top half of the table. The second half of the table is simply the top half in reverse order.

**Equation 2:**

\[ D(\text{step number}) = \cos\left(\frac{\text{step number} \times \Pi}{(\text{number of steps})+1}\right) \times (2^{\text{bits resolution}}-1) \]

Using Equation 2 the following duty cycle values were calculated for a 16 microsteps per full step sequence using an 8-bit resolution PWM waveform:

**TABLE 1: DUTY CYCLE VALUES FOR MICROSTEPPING**

<table>
<thead>
<tr>
<th>Step Number</th>
<th>D</th>
<th>Step Number</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>251</td>
<td>9</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>240</td>
<td>10</td>
<td>87</td>
</tr>
<tr>
<td>3</td>
<td>221</td>
<td>11</td>
<td>128</td>
</tr>
<tr>
<td>4</td>
<td>195</td>
<td>12</td>
<td>164</td>
</tr>
<tr>
<td>5</td>
<td>164</td>
<td>13</td>
<td>195</td>
</tr>
<tr>
<td>6</td>
<td>128</td>
<td>14</td>
<td>221</td>
</tr>
<tr>
<td>7</td>
<td>87</td>
<td>15</td>
<td>240</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>16</td>
<td>251</td>
</tr>
</tbody>
</table>

**PWM Generation Using the ECCP Module**

The ECCP module on the PIC16F684 is ideal for generating the PWM signal required for microstepping. The module is capable of generating a 10-bit resolution PWM waveform at frequencies ranging up to 7.81 kHz when using the microcontroller’s 8 MHz internal oscillator. Higher frequencies are possible, but at the cost of PWM resolution.

The ECCP module has four modes of operation:
1) Single output
2) Half-bridge output
3) Full-bridge forward output
4) Full-bridge reverse output

In half-bridge mode the module modulates two pins simultaneously, pins P1A and P1B. For this application, these two outputs are used to drive the two windings of a stepping motor. The following circuit diagram shows how these pins are connected to a bipolar drive circuit.
FIGURE 2: BIPOLAR DRIVE CIRCUIT
Note the pull-up resistor on pins P1A and P1B. These resistors clamp the respective line high when the pin is tristated. It is important that the non-modulated line be clamped high so that the NAND gates on either end of the winding can drive current through the winding when the respective control line is enabled.

The ECCP module is set up so that the waveforms on pins P1A and P1B are identical. This is done by configuring the CCP1CON register so that P1A is active high and P1B is active low. With no dead band delay these pins will behave identically. Configuring the module in this way enables each winding control block to use the same duty cycle look-up table values for its transition sequence. The following table shows all eight winding states.

### TABLE 2: WINDING STATES

<table>
<thead>
<tr>
<th>STATE</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winding 1</td>
<td>+ to 0</td>
<td>0 to -</td>
<td>-</td>
<td>-</td>
<td>- to 0</td>
<td>0 to +</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Polarity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winding 2</td>
<td>+</td>
<td>+</td>
<td>+ to 0</td>
<td>0 to -</td>
<td>-</td>
<td>-</td>
<td>- to 0</td>
<td>0 to +</td>
</tr>
<tr>
<td>Polarity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1A Duty Cycle</td>
<td>100% to 0</td>
<td>0 to 100%</td>
<td>100%*</td>
<td>100%*</td>
<td>100% to 0</td>
<td>0 to 100%</td>
<td>100%*</td>
<td>100%*</td>
</tr>
<tr>
<td>P1B Duty Cycle</td>
<td>100%*</td>
<td>100%*</td>
<td>100% to 0</td>
<td>0 to 100%</td>
<td>100%*</td>
<td>100%*</td>
<td>100% to 0</td>
<td>0 to 100%</td>
</tr>
<tr>
<td>TRISC, P1A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TRISC, P1B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CTRL1A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CTRL1B</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CTRL2A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>CTRL2B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

* Pin is tristated and the pull-up resistor is clamping the line high.

In States 0, 2, 4 and 6 the first half of the duty cycle sine look-up table (decreasing values) is referenced. In States 1, 3, 5 and 7 the second half of the duty cycle sine look-up table (increasing values) is referenced.

### CURRENT LIMITING

Current limiting is used when a stepping motor is driven at a voltage that is higher than the motor’s rated voltage. There are several advantages to driving a motor at high voltage, namely, the torque and speed characteristics of the motor are improved. These parameters are improved because the current in the motor windings is more responsive to changes made by the controller.

When running a motor at a voltage that is higher than its rated voltage, current limiting considerations must be made. The current in the windings of the motor can not exceed the maximum current rating of the motor or motor life will be severely affected. One of the most effective ways to limit current is through the use of a chopper circuit. The chopper circuit for one winding is shown in Figure 3. Winding 2 uses the second on-chip comparator in its chopper circuit.
FIGURE 3: BIPOLAR STEPPING MOTOR DRIVE CIRCUIT WITH CHOPPER CONTROL
An over current condition is detected by comparing the voltage measured across R\text{ SENSE} (V_{\text{SENSE}}) to a control voltage (V_{\text{CONTROL}}). The maximum voltage V_{\text{SENSE}} should be allowed to reach is determined by:

**EQUATION 3:**

\[ V_{\text{CONTROL}} = R_{\text{SENSE}} \times I_{\text{MAX}} \]

V_{\text{CONTROL}} is set so that it is equal to V_{\text{SENSE}} when I_{\text{MAX}} is reached. A simple voltage divider is used to create V_{\text{CONTROL}} where the values of R1 and R2 are chosen to satisfy the following equation:

**EQUATION 4:**

\[ V_{\text{CONTROL}} = \frac{R_1}{(R_1+R_2)}V_{\text{CC}} \]

R_{\text{SENSE}} is chosen arbitrarily, however, the smaller R_{\text{SENSE}} is the more efficient the circuit. If R_{\text{SENSE}} is too small then V_{\text{SENSE}} will be very small and the comparator output will be more sensitive to noise. The power dissipation of R_{\text{SENSE}} was determined by:

**EQUATION 5:**

\[ P = I_{\text{MAX}}^2 \times R_{\text{SENSE}} \]

When the comparator indicates that I_{\text{MAX}} has been reached, the control lines are forced low for the respective winding. This is done in firmware.

It is only necessary to detect over-current in the winding that is not being modulated. For example, this would be states 2, 3, 6 and 7 for winding 1 (refer to Table 2). Current in the winding that is being modulated is controlled by limiting the maximum duty cycle of the PWM waveform. This requires a modification to the duty cycle sine look-up table talked about in the previous section. The table is modified according to the following relation:

**EQUATION 6:**

\[ D(\text{step number}) = \frac{D_\text{x}(\text{step number}) \times V_{\text{MAX}}}{V_{\text{SUPPLY}}} \]

where V_{\text{MAX}} is the rated voltage of the motor and D_{\text{x}} is the array of duty cycle values calculated using Equation 2 from the previous section.

**EXAMPLE APPLICATION**

This example application demonstrates how to drive a 1.8 degree-per-step stepping motor. The motor used is a bipolar stepping motor rated to draw 1/2 amp at 5V. For this example, the motor is driven at 12V or 2.4 times the motor’s rated voltage.

**Hardware**

Appendix A shows a schematic for the example application included with this application note. The drive circuit is composed of four Fairchild Semiconductor half-bridge MOSFET ICs (part number FDC6420C). These ICs are tied to ground via a 1 ohm, 1/2 W resistor. The value of R_{\text{SENSE}} was chosen using Equation 1 so that V_{\text{SENSE}} will be equal to 0.5V when I_{\text{MAX}} is reached. Using Equation 3, the power dissipated by R_{\text{SENSE}} is 1/4 W, therefore, a 1/2 W resistor is sufficient.

Two Microchip logic-input CMOS quad drivers are used to drive the MOSFET ICs and to provide the logic necessary for the implementation described in this application note. The TC4467 has four on-chip NAND gates and the TC4468 has four on-chip AND gates. The inputs to each of the AND gates on the TC4468 are tied together because this IC is used as a non-inverting quad MOSFET driver for this implementation. Resistors R17 and R13 were chosen so that V_{\text{CONTROL}} equals approximately 0.5 volts. R17 is 10 k\text{Ohm} and R13 is 90.0 k\text{Ohm}.

**Firmware**

A flowchart illustrating the firmware implementation of this example is in Appendix B. The source code for this application note is written in the Microchip assembly language and is included with this application note on Microchip’s web site, www.microchip.com.

**Operation**

The speed of the motor is adjusted using a potentiometer (R18). The motor is started and stopped via Switch 1. There are four modes of operation in the example that are sequenced through with single button presses. The modes of operation are off, forward, off, and reverse.
CONCLUSION

The PIC16F684 has an ideal set of peripherals for low-cost stepping motor control. The ECCP module can be configured for microstepping applications while the two on-chip comparators can be configured to provide current feedback for current limiting applications. Extra I/O pins and CPU resources are left over for implementing serial communication, an LCD interface, or any number of I/O configurations.

REFERENCES

AN907: Stepper Motor Fundamentals
APPENDIX A: (CONTINUED)

```
J7      VSUPPLY
DIN5P_RECEPTICAL

U10
LM78L05ACM
IN       OUT
GND GND GND GND

100 uF
C11
1 uF
C12
0.1 uF
C13
0.1 uF
C14

+5V

P5  P4  P3
1  1  1
2  2  RA5
3  3  RA4
4  4  VPP
5  5  RC5
6  6  RC4
7  7  RC3
8  8  ICSPDAT
9  9  ICSPCLK
10 10  RA2
11 11  RC0
12 12  RC1
13 13  RC2
14 14  +5V

HDR1X14  HDR1X14  HDR1X14

RA5  RA4  +5V
VPP
P1A
P1B
RC5
RC4
RC3
ICSPDAT
ICSPCLK
RA2
RC0
RC1
RC2
+5V
```

© 2004 Microchip Technology Inc.
APPENDIX B:

Initialize

Internal oscillator frequency = 8 MHz

Assign I/O pins

Setup ADC pin: AN2

Setup ECCP Module: Half Bridge Mode, 31.25 kHz waveform

Set TMR0 parameters

Turn on Comparators

Turn on TMR2

goto Main
APPENDIX B: (CONTINUED)

Main

Read Switch

Yes

Button Press?

No

Debounce Routine

Increment Mode

Yes

Mode?

0

Clear control lines: CTRLA1, CTRLA2, CTRLB1, CTRLB2

goto Main

1

Set Direction Flag

goto MotorState

2

Clear control lines: CTRLA1, CTRLA2, CTRLB1, CTRLB2

goto Main

3

Clear Direction Flag

goto MotorState

goto Main
APPENDIX B: (CONTINUED)

MotorState

TMR2 Interrupt Flag Set?
Yes

Clear TMR2 Interrupt Flag
Read Potentiometer
Move high 4-bits of ADC value into TMR2 Prescaler
Initiate next ADC read

R3

Motor Direction?
Forward
Reverse

Duty Cycle Lookup Table

Read previous Duty Cycle value

End/Middle of Lookup Table?
Yes

Increment State

No

State?

Duty Cycle Lookup Table

Increment State

goto State0
goto State2
goto State4
goto State6
goto State1
goto State3
goto State5
goto State7
APPENDIX B: (CONTINUED)

State0

Enable P1A
Disable P1B
CTRLA1 = 0
CTRLA2 = 0
CTRLB1 = 1
CTRLB2 = 1

Imax Reached?

No

CTRLBx = 0

goto Main

Yes

CTRLBx = 0

goto Main

State1

Enable P1A
Disable P1B
CTRLA1 = 0
CTRLA2 = 1
CTRLB1 = 1
CTRLB2 = 0

Imax Reached?

No

CTRLBx = 0

goto Main

Yes

CTRLBx = 0

goto Main

State2

Disable P1A
Enable P1B
CTRLA1 = 0
CTRLA2 = 0
CTRLB1 = 1
CTRLB2 = 1

Imax Reached?

No

CTRLAx = 0

goto Main

Yes

CTRLAx = 0

goto Main

State3

Disable P1A
Enable P1B
CTRLA1 = 1
CTRLA2 = 0
CTRLB1 = 0
CTRLB2 = 1

Imax Reached?

No

CTRLAx = 0

goto Main

Yes

CTRLAx = 0

goto Main

State4

Enable P1A
Disable P1B
CTRLA1 = 0
CTRLA2 = 1
CTRLB1 = 0
CTRLB2 = 1

Imax Reached?

No

CTRLBx = 0

goto Main

Yes

CTRLBx = 0

goto Main

State5

Enable P1A
Disable P1B
CTRLA1 = 1
CTRLA2 = 1
CTRLB1 = 1
CTRLB2 = 1

Imax Reached?

No

CTRLBx = 0

goto Main

Yes

CTRLBx = 0

goto Main

State6

Disable P1A
Enable P1B
CTRLA1 = 0
CTRLA2 = 1
CTRLB1 = 0
CTRLB2 = 1

Imax Reached?

No

CTRLAx = 0

goto Main

Yes

CTRLAx = 0

goto Main

State7

Disable P1A
Enable P1B
CTRLA1 = 1
CTRLA2 = 0
CTRLB1 = 0
CTRLB2 = 1

Imax Reached?

No

CTRLAx = 0

goto Main

Yes

CTRLAx = 0

goto Main
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.