AN899

Brushless DC Motor Control Using PIC18FXX31 MCUs

Author: Padmaraja Yedamale
Microchip Technology Inc.

INTRODUCTION

The PIC18F2331/2431/4331/4431 family of microcontrollers have peripherals that are suitable for motor control applications. These peripherals and some of their primary features are:

- Power Control PWM (PCPWM)
  - Up to 8 output channels
  - Up to 14-bit PWM resolution
  - Center-aligned or edge-aligned operation
  - Hardware shutdown by Fault pins, etc.
- Quadrature Encoder Interface (QEI)
  - QEA, QEB and Index interface
  - High and low resolution position measurement
  - Velocity Measurement mode using Timer5
  - Interrupt on detection of direction change
- Input Capture (IC)
  - Pulse width measurement
  - Different modes to capture timer on edge
  - Capture on every input pin edge
  - Interrupt on every capture event
- High-Speed Analog-to-Digital Converter (ADC)
  - Two sample and hold circuits
  - Single/Multichannel selection
  - Simultaneous and Sequential Conversion mode
  - 4-word FIFO with flexible interrupts

In this application note, we will see how to use these features to control a Brushless DC (BLDC) motor in open loop and in closed loop. Refer to the Microchip application note, *"AN885, Brushless DC (BLDC) Motor Fundamentals*" (DS00885), for working principles of Brushless DC motors and basics of control. Also, to obtain more information on motor control peripherals and their functions, refer to the PIC18F2331/2431/4331/4431 Data Sheet (DS39616).

HARDWARE

A PICDEM™ MC demo board was used to develop, test and debug the motor control code. The PICDEM MC has a single-phase diode bridge rectifier, converting AC input to DC and a power capacitor bank that keeps a stable DC bus. A 3-phase IGBT-based inverter bridge is used to control the output voltage from the DC bus. Figure 1 shows the overall block diagram of the hardware.

The control circuit and power circuits are optically isolated with respect to each other. An on-board fly-back power supply generates +5VD, with respect to the digital ground used for powering up the control circuit, including the PICmicro® device. +5VA and +15VA are generated with respect to the power ground (negative of DC bus). The feedback interface circuit is powered by +5VA, while +15VA supplies power to the IGBT drivers located inside the Integrated Power Module (IPM).

With the optical isolation between power and control circuits, programming and debugging tools can be plugged into the development board when main power is connected to the board. The board communicates with a host PC over a serial port configured with an on-chip Enhanced USART. The on-board user interface has two toggle switches, a potentiometer and four LEDs for indication.

In this application note, the switch SW1 is used to toggle between motor Run and Stop and SW2 is used to toggle between the direction of motor rotation. Each press of these buttons will change the state. A potentiometer is used for setting the speed reference. The LEDs are used for indication of different states of control.

Reference copies of the PICDEM™ MC schematics can be found in **Appendix B: "Circuit Schematics"**.

OPEN-LOOP CONTROL

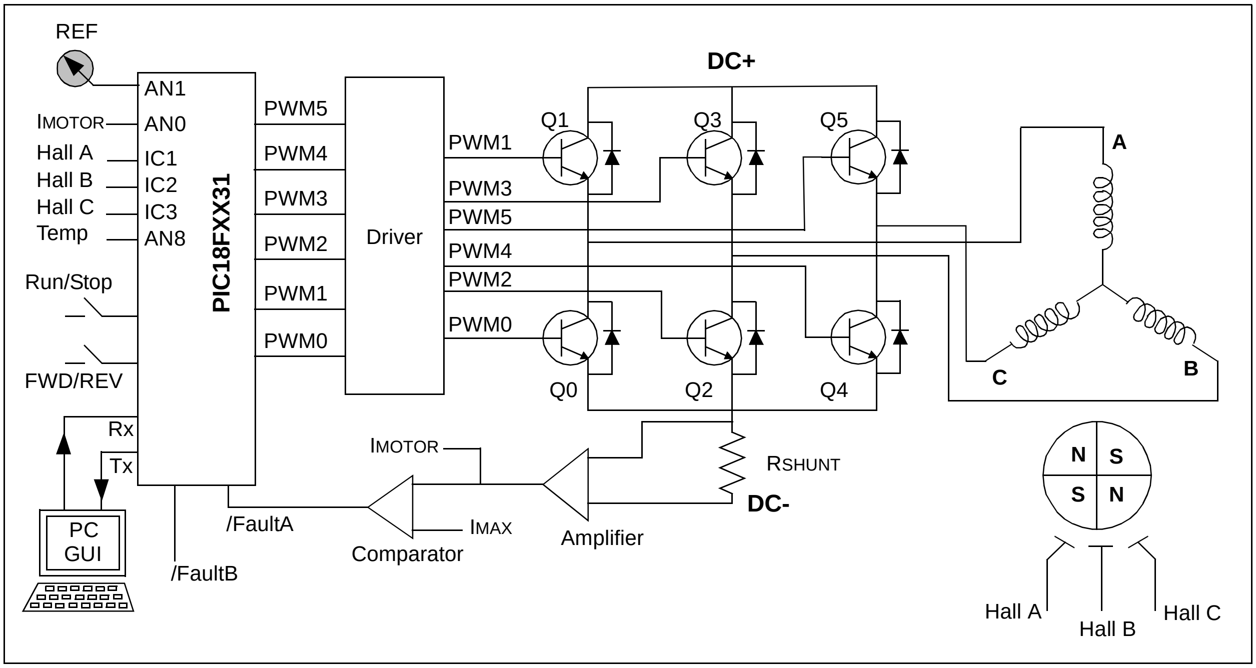
As seen in AN885, BLDC motors are electronically commutated based on the rotor position. Each commutation sequence has two of three phases connected across the power supply and the third phase is left open. Using PWMs, the average voltage supplied across the windings can be controlled, thus controlling the speed. In this section, we will see how the peripherals on the PIC18FXX31 can be used to control a BLDC motor.

Figure 1 shows a typical control block diagram for controlling a BLDC motor.

The PWM outputs from the PIC18FXX31 control the power switches, Q0 to Q5. A matching driver circuit should be used for supplying the required gate current drive for the power switches. As we have seen in AN885, the Hall Sensor signals may have 60-degree, or 120-degree, electrical phase difference to each other. A sequence table is entered in the program memory based on the type of Hall Sensor placement. The sequence can be taken from the motor data sheet. The sequence may be different for clockwise and counterclockwise rotations.

The following section explains how PCPWM, IC and ADCs are used for open-loop control.

FIGURE 1: BLDC MOTOR CONTROL BLOCK DIAGRAM



USING THE INPUT CAPTURE MODULE

Hall Sensors A, B and C are connected to IC1, IC2 and IC3, respectively, on the Input Capture (IC) module. The Input Capture module is used in “Input Capture on State Change” mode. In this mode, the IC module interrupts every transition on any of the IC pins. Also, Timer5 is captured on every transition and cleared at the beginning of the next clock cycle. The captured Timer5 value is useful in determining the speed of the motor. Measuring the speed and controlling the motor in closed loop is discussed in detail in the section **“Closed-Loop Control Using Hall Sensors”**.

Upon IC interrupt, in the IC Interrupt Service Routine, the status of all three input capture pins is read and the combination is used to pick up the correct sequence from the table.

Table 1 shows a typical switching sequence used to run the motor in the clockwise direction and Table 2 shows the counterclockwise sequence. These tables are taken directly from the motor data sheet[(1)].

Note 1: Motor Data Sheet
Manufacturer: Bodine Electric Company
Type Number: 22B4BEBL
Series: 3304
Web Site: www.bodine-electric.com

If the motor you have uses a different sequence, it should be entered in the firmware. Figure 2 shows the relationship between the motor phase current and the Hall Sensor inputs and the corresponding PWM signals to be activated to follow the switching sequence, which in turn, runs the motor in the clockwise direction.

TABLE 1: SEQUENCE FOR ROTATING THE MOTOR IN CLOCKWISE DIRECTION WHEN VIEWED FROM NON-DRIVING END

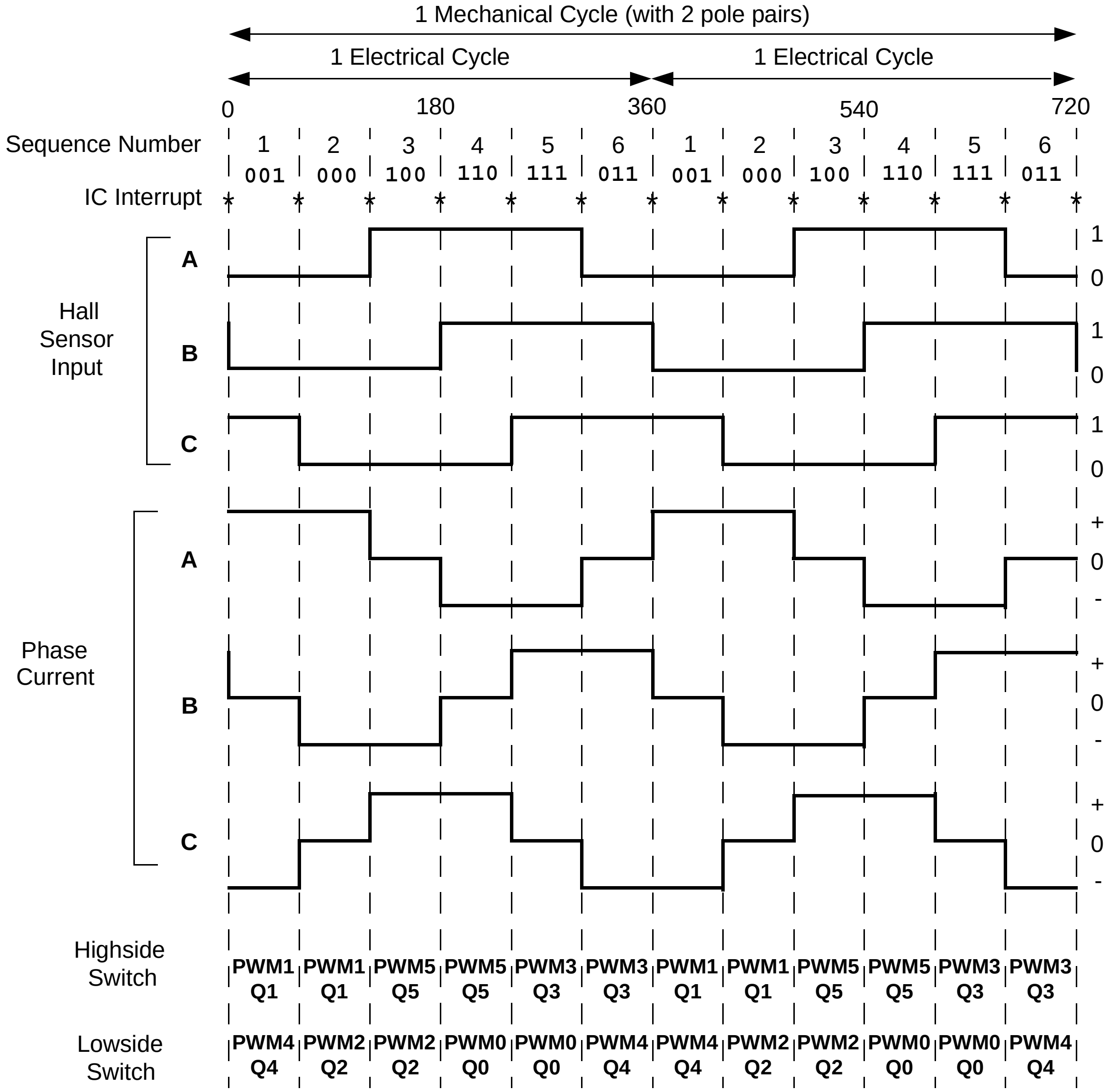
Sequence Number	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	0	1	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
2	0	0	0	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	0	0	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
4	1	1	0	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+
5	1	1	1	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	1	1	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-

TABLE 2: SEQUENCE FOR ROTATING THE MOTOR IN COUNTERCLOCKWISE DIRECTION WHEN VIEWED FROM NON-DRIVING END

Sequence Number	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	1	1	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
2	1	1	1	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	1	0	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
4	1	0	0	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-
5	0	0	0	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	0	1	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+

Figure 2 is drawn with respect to Table 1. The sequence number in Table 1 corresponds to 60 degrees of the electrical cycle shown in Figure 2. For example, as seen in Sequence 1 in Table 1, the Hall Sensor input is set at '`001`', which should activate Q1 and Q4. The corresponding PWMs (PWM1 and PWM4) are active during this 60-degree cycle. For the next 60-degree cycle, the Hall Sensor input is '`000`' and Q1 (PWM1) and Q2 (PWM2) are active.

FIGURE 2: HALL SENSOR INPUT VERSUS PHASE CURRENT



USING THE PCPWM MODULE

The PCPWM module is used in Independent mode to control the PWM output. In this mode, three duty cycle registers control 6 PWM outputs, with two each having the same output; meaning the duty cycles on PWM0 and PWM1 are controlled by the PDC0H:PDC0L registers, the duty cycles on PWM2 and PWM3 are controlled by PDC1H:PDC1L registers and so on. Looking at the sequence in Table 1 and Table 2, PWM0, PWM2 and PWM4 should be OFF any time that PWM1, PWM3 and PWM5 are ON and vice versa.

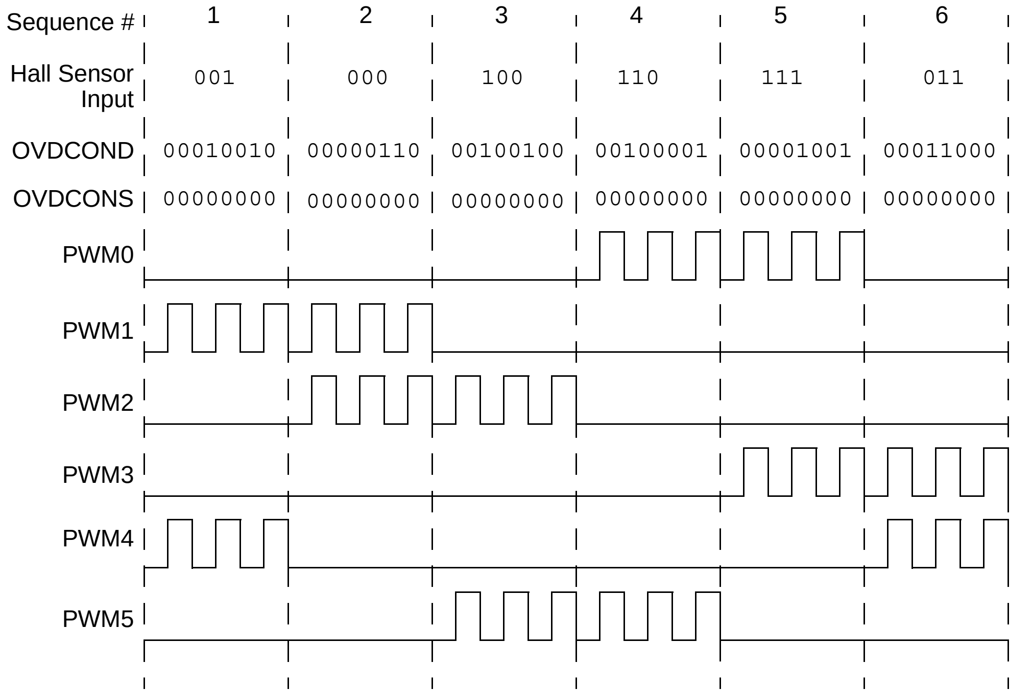
In order to keep the required PWMs active and to inhibit other PWMs from becoming active, the PWM override feature is used. The PCPWM module has a feature of overriding the PWM outputs based on the bit setting in the Special Function Register, OVDCOND. The bits in the OVDCOND register correspond directly to the PWM channel it is controlling. When the corresponding bit is set to ‘1’, the set duty cycle appears on the pin. When the bit is set to ‘0’, the output state is determined by the register, OVDCONS. If the corresponding bit in OVDCONS is set to ‘1’, then the corresponding output is ‘active’; if it is ‘0’, the output is ‘inactive’.

Figure 3 shows an example of setting OVDCOND and OVDCONS registers and PWM outputs corresponding to Table 1.

As shown in Figure 3, the value loaded to the OVDCOND register is determined by the Hall Sensor and the switching sequence. When the PWM needs to be active, the corresponding OVDCOND bit is set to ‘1’ and vice versa. To vary the motor speed, in addition to the OVDCONx registers, PWM duty cycle registers also should be calculated and reloaded based on the set speed.

Note:	Refer to the configuration bits, HPOL and LPOL, in **Section 22.0 “Special Features of the CPU”** of the PIC18F2331/2431/4331/4431 Data Sheet to define the ‘active’ and ‘inactive’ states for the PWM outputs.

FIGURE 3: OVDCOND VERSUS PWM OUTPUT



PWM DUTY CYCLE CALCULATION

PWM duty cycle depends mainly upon three factors: motor rated voltage, DC bus voltage and the speed reference setting. Normally, the DC bus voltage would be at least 10% more than the motor rated voltage to achieve complete speed range. The ratio of motor voltage to the DC bus voltage determines the maximum allowed PWM duty cycle. There can be different ways of inputting speed reference to the controller. It may be from a potentiometer connected to one of the AD Channels, as shown in Figure 1, or it may be a digital value from a host PC or from another controller, or a PWM input with varying duty cycle indicating varying speed. In this application note, speed reference is taken from a potentiometer connected to AD Channel 1 of the PIC18FXX31.

The PWM duty cycle is calculated as shown in Equation 1.

100% of duty cycle corresponds to 4*PTPER register. The value in the PTPER register is responsible for setting the PWM frequency. In order to get the maximum benefit out of PWM, a ratio of the maximum allowed value in duty cycle in relation to the maximum speed reference value is taken and multiplied by Equation 1. Equation 1 is then modified as shown in Equation 2.

Assuming the PWM frequency is not changed on the fly, the only run time variable in Equation 2 is the speed reference. The remaining term can be defined as a compile time constant.

AD Channel 1 is read at a fixed interval and the PWM duty cycle is calculated and loaded to PDCx registers. Example 2 and Example 1 show the code to access the table and determine the sequence based on the Hall inputs. Example 3 shows PWM duty cycle calculation.

EQUATION 1: THEORETICAL PWM DUTY CYCLE

$$\text{PWM Duty Cycle} = \frac{\text{Motor Rated Voltage}}{\text{DC Bus Voltage}} \text{ x Speed Reference}$$

EQUATION 2: ACTUAL PWM DUTY CYCLE

$$\text{PWM Duty Cycle} = \frac{\text{Motor Rated Voltage}}{\text{DC Bus Voltage}} \text{ x } \frac{\text{PTPER x 4}}{\text{Maximum Speed Reference}} \text{ x Speed Reference}$$

Software Functions

Figure 4 shows the simplified flow chart of the main loop and Figure 5 shows the flow chart of the Interrupt Service Routine (ISR).

Main Loop: The Main Loop has the initialization routine, Fault display and key detection and decoding.

Initialization Routine: This routine initializes all peripherals used in this application. PWM is initialized to output in Independent mode with a selectable PWM frequency. Fault input is configured in Cycle-by-Cycle mode. In this mode, PWM outputs are driven to an inactive state until the Fault exists. In the next PWM cycle, the outputs are resumed to active state.

Key Activity Monitoring: Both SW1 and SW2 are monitored and each press of either button toggles the state corresponding to the keys. SW1 is used to toggle the states between Run and Stop of the motor. SW2 is used to toggle between two directions. When SW2 is pressed, the motor is decelerated to stop and accelerated in the opposite direction.

Fault Signals: There are three Faults being monitored: Overcurrent, Overvoltage and Overtemperature.

Overcurrent Fault: A shunt resistor in the negative DC bus gives a voltage corresponding to the current flowing into the motor winding. This voltage is amplified and compared with a reference. The current comparison setting allows a current up to 6.3 Amps. If the current exceeds 6.3 Amps, the Fault A pin goes low, indicating the Overcurrent. The firmware is configured in Cycle-by-Cycle Fault mode. If the Fault occurs more than 20 times in 256 PWM cycles, then the motor is stopped and an Overcurrent Fault is indicated by blinking LED1.

Overvoltage Fault:: The DC bus voltage is attenuated using potential dividers and compared with a fixed reference. If jumper JP5 is open, the Overvoltage is set at 200V on the DC bus. If jumper JP5 is short, the Overvoltage limit is 400V. The Fault B pin is used to monitor the Overvoltage condition. If the Overvoltage persists for more than 20 times in 256 PWM cycles, then the motor is stopped and an Overvoltage Fault is indicated by blinking LED2.

Overtemperature: The power module has an NTC thermal sensor, outputting 3.3V at 110°C on the junction of IGBTs. The NTC output is connected to AN8 through an opto-coupler. The temperature is continuously measured and if it exceeds 80°C, then the motor is stopped and an Overcurrent Fault is indicated by blinking LED3.

ISR Loop: In the ISR loop, mainly the Hall Sensor transition and AD Channel conversion are monitored.

Hall Sensor: Any transition on Hall Sensor inputs will read the corresponding value from the sequence table corresponding to the direction. This value is loaded into the OVDCOND register. OVDCONS is maintained cleared always. Also, LED1, 2 and 3 indicate the state of the Hall Sensor inputs.

A/D Channel Conversion: AN0, AN1 and AN8 Channels are converted in every cycle. The AN1 result is used for determining the speed reference input. The PWM duty cycle is calculated using Equation 2. AN0 is the motor current. The motor current value is compared with a value determined by the motor rated current. If the limit exceeds 1.5 times the rated motor current, then the motor is stopped and an Overcurrent Fault is indicated by blinking LED1.

EXAMPLE 1: SEQUENCE TABLE INITIALIZATION

```
;Commutation definition. This should be loaded to OVDCOND to realize the sequence
;The Hall Sensor makes a transition every 60 degrees

#define    POSITION1          b'00010010'    ;PWM1 & PWM4 are active
#define    POSITION2          b'00000110'    ;PWM1 & PWM2 are active
#define    POSITION3          b'00100100'    ;PWM5 & PWM2 are active
#define    POSITION4          b'00100001'    ;PWM5 & PWM0 are active
#define    POSITION5          b'00001001'    ;PWM3 & PWM0 are active
#define    POSITION6          b'00011000'    ;PWM3 & PWM4 are active
#define    DUMMY_POSITION     b'00000000'    ;All PWM outputs are inactive
;----------------------------------------------------------------------------------
;Table initialization, Table values are loaded to RAM
;Forward sequence
    MOVLW   POSITION2                  ;When Hall Sensor = 000,
    MOVWF   POSITION_TABLE_FWD         ;PWM1 & PWM2 should be active
    MOVLW   POSITION3                  ;When Hall Sensor = 001,
    MOVWF   POSITION_TABLE_FWD+1       ;PWM1 & PWM4 should be active
    MOVLW   DUMMY_POSITION             ;When Hall Sensor = 002,
    MOVWF   POSITION_TABLE_FWD+2       ;All PWM outputs should be inactive
    MOVLW   POSITION4                  ;When Hall Sensor = 003,
    MOVWF   POSITION_TABLE_FWD+3       ;PWM3 & PWM4 should be active
    MOVLW   POSITION1                  ;When Hall Sensor = 004,
    MOVWF   POSITION_TABLE_FWD+4       ;PWM5 & PWM2 should be active
    MOVLW   DUMMY_POSITION             ;When Hall Sensor = 005,
    MOVWF   POSITION_TABLE_FWD+5       ;All PWM outputs should be inactive
    MOVLW   POSITION6                  ;When Hall Sensor = 006,
    MOVWF   POSITION_TABLE_FWD+6       ;PWM5 & PWM0 should be active
    MOVLW   POSITION5                  ;When Hall Sensor = 007,
    MOVWF   POSITION_TABLE_FWD+7       ;PWM3 & PWM0 should be active

;Reverse sequence
    MOVLW   POSITION5                  ;When Hall Sensor = 000,
    MOVWF   POSITION_TABLE_REV         ;PWM3 & PWM0 should be active
    MOVLW   POSITION6                  ;When Hall Sensor = 001,
    MOVWF   POSITION_TABLE_REV+1       ;PWM5 & PWM0 should be active
    MOVLW   DUMMY_POSITION             ;When Hall Sensor = 002,
    MOVWF   POSITION_TABLE_REV+2       ;All PWM outputs should be inactive
    MOVLW   POSITION1                  ;When Hall Sensor = 003,
    MOVWF   POSITION_TABLE_REV+3       ;PWM5 & PWM2 should be active
    MOVLW   POSITION4                  ;When Hall Sensor = 004,
    MOVWF   POSITION_TABLE_REV+4       ;PWM3 & PWM4 should be active
    MOVLW   DUMMY_POSITION             ;When Hall Sensor = 005,
    MOVWF   POSITION_TABLE_REV+5       ;All PWM outputs should be inactive
    MOVLW   POSITION3                  ;When Hall Sensor = 006,
    MOVWF   POSITION_TABLE_REV+6       ;PWM1 & PWM4 should be active
    MOVLW   POSITION2                  ;When Hall Sensor = 007,
    MOVWF   POSITION_TABLE_REV+7       ;PWM1 & PWM2 should be active
```

EXAMPLE 2: SEQUENCE TABLE DEFINITION/ACCESS

```
;Hall Sensors are connected to IC1,IC2 and IC3 on PORTA<4:2>.
;IC module is initialized to capture on every transition on any of the IC pins.
;This is the ISR for IC
UPDATE_SEQUENCE
    BTFSS   FLAGS1,FWD_REV          ;Check for direction command
    BRA     ITS_REVERSE             ;Branch if it is reverse
    LFSR    0,POSITION_TABLE_FWD    ;If forward, point FSR0 to the first location on the
    BRA     PICK_FROM_TABLE         ;forward table
ITS_REVERSE
    LFSR    0,POSITION_TABLE_REV    ;If reverse, point FSR0 to the first location on the reverse
                                    ;table
PICK_FROM_TABLE
    MOVF    PORTA,W                 ;Read PORTA and discard other bits
    ANDLW   0x1C                    ;
    RRNCF   WREG, W
    RRNCF   WREG, W                 ;Readjust the result to LSBits
    MOVF    PLUSW0, W               ;Read the value from table offset by the Hall input value
    MOVWF   OVDCOND                 ;Load to OVDCOND
    RETURN
```

EXAMPLE 3: PWM DUTY CYCLE CALCULATION CODE EXAMPLE

```
;Defining the PWM duty cycle constant based on the Motor voltage, DC bus voltage and PWM period
#define MOTOR_VOLTAGE d'130'
#define AC_INPUT_VOLTAGE d'115'
#define MAX_SPEED_REF ‘256’
PWM_CONSTANT =((MOTOR_VOLTAGE*PTPER_VALUE*4')/(1.414*AC_INPUT_VOLTAGE*MAX_SPEED_REF))*d’16’
;Multiplication factor of 16 is used to scale the result.
;-------------------------------------------------------------------------------------------
CALCULATE_PWM
;PWM = PWM_CONSTANT * SPEED_REF(read from ADC, only 8 MS bits are taken for simplicity)
    MOVF    SPEED_REF,W
    MULLW   (PWM_CONSTANT)              ;PWM_CONSTANT*SPEED_REF
    SWAPF   PRODL,W
    ANDLW   0x0F
    MOVWF   PDC_TEMPL
    SWAPF   PRODH, W
    ANDLW   0xF0
    IORWF   PDC_TEMPL,F
    SWAPF   PRODH, W
    ANDLW   0x0F                        ;Divide the result in PRODH:PRODL by 16 and load to the
    MOVWF   PDC_TEMPH                   ;Duty cycle registers
    MOVFF   PDC_TEMPH,PDCxH
    MOVFF   PDC_TEMPL,PDCxL
    RETURN
```

FIGURE 4: MAIN LOOP

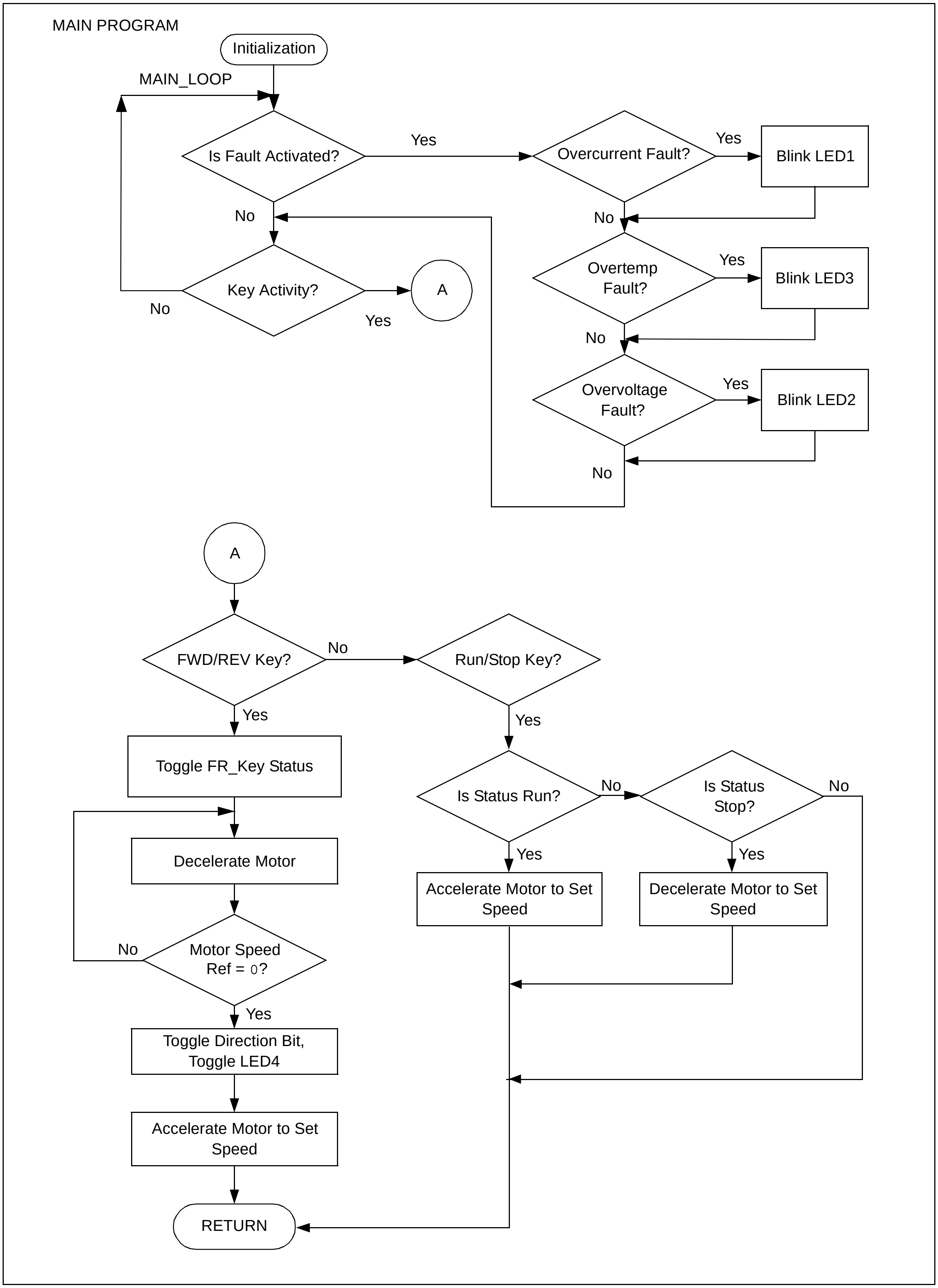
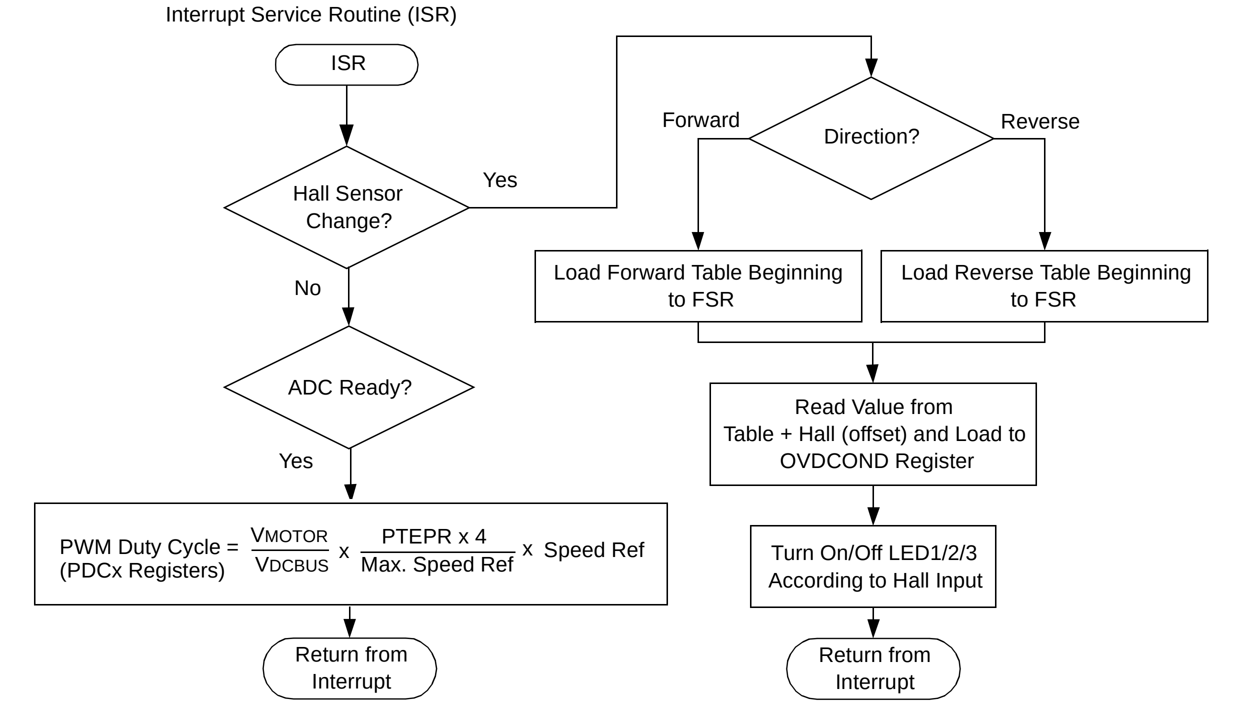


FIGURE 5: INTERRUPT SERVICE ROUTINE (ISR)


Interrupt Service Routine (ISR)
ISR
Hall Sensor Change?
Yes
No
Direction?
Forward
Reverse
Load Forward Table Beginning to FSR
Load Reverse Table Beginning to FSR
Read Value from Table + Hall (offset) and Load to OVDCOND Register
ADC Ready?
Yes
PWM Duty Cycle = (PDCx Registers) VMOTOR / VDCBUS x PTEPR x 4 / Max. Speed Ref x Speed Ref
Turn On/Off LED1/2/3 According to Hall Input
Return from Interrupt
Return from Interrupt

CLOSED-LOOP CONTROL USING HALL SENSORS

As we have seen in an earlier section, Timer5 is captured on every transition on Input Capture used for Hall Sensor inputs. Given this, the Timer5 value is captured 6 times in one electrical cycle. This electrical cycle repeats as many times as the number of rotor pole pairs to complete a mechanical rotation. For example, if the rotor has 4 poles or 2 pole pairs, the electrical cycle repeats twice for one mechanical rotation of the shaft, as shown in Figure 2. Timer5 is captured 12 times per one shaft rotation. The Timer5 value is averaged over one rotation and this value is taken for determining the motor speed.

TIMER5 VALUE VERSUS MOTOR SPEED

Translating Timer5 value into motor speed is dependant upon the following factors:

- Operating frequency
- Timer5 prescaler
- Number of rotor pole pairs

Rotor pole pairs may vary from 2 to 20, depending upon the motor chosen for the application. Based on the number of rotor pole pairs, the number of Timer5 samples taken for averaging will vary to get the best result. Equation 3 shows the speed calculated from the Timer5 value in Revolutions Per Minute (RPM).

The actual value calculated in firmware may be Revolution Per Second (RPS) or scaled version of the absolute number.

Similarly, the speed reference input is translated into a speed value in order to have both reference and feedback in the same platform. Equation 4 shows converting speed reference from a potentiometer setting read through an AD channel.

Speed reference is in RPM, if the rated speed entered is in RPM. Example 4 shows code used to calculate speed reference taken from the potentiometer. Only the eight Most Significant bits are taken for simplicity.

EQUATION 3: MOTOR SPEED FROM TIMER5

$$\text{Speed in RPM} = \left(\frac{\text{Operating Frequency/4}}{\text{Timer5 Count x Timer5 Prescale x Number of Pole Pairs x 6}}\right) \text{x } 60$$

EQUATION 4: SPEED REFERENCE CALCULATION

$$\text{Speed Reference} = \text{Rated Motor Speed x } \frac{\text{ADC Value}}{\text{Maximum ADC Value}}$$

EXAMPLE 4: SPEED REFERENCE CALCULATION CODE EXAMPLE

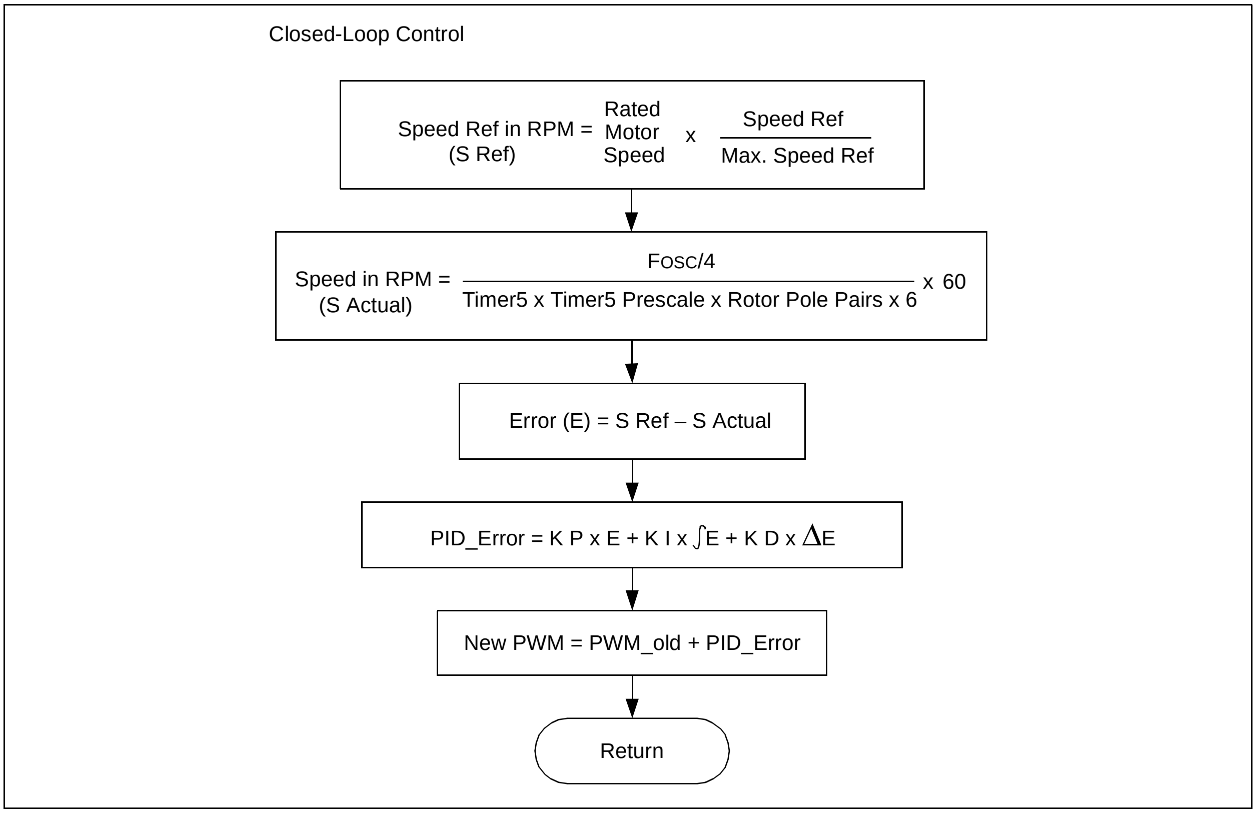
```
#define   MOTOR_RATED_SPEED ‘3500’
#define   MAX_SPEED_REFERENCE ‘256’

SPEED_REF_RATIO = MOTOR_RATED_SPEED* 0xFF / MAX_SPEED_REFERENCE
;0xFF is a multiplication factor, divided when actual speed ref is calculated

CALCULATE_SPEED_REF
    MOVLW   LOW(SPEED_REF_RATIO)
    MULWF   SPEED_REFH             ;SPEED_REF_RATIO* speed reference read
    MOVFF   PRODH,TEMP             ;from ADC (SPEED_REFH = 8 MSB’s pf speed reference)
    MOVLW   HIGH(SPEED_REF_RATIO)
    MULWF   SPEED_REFH             ;For simplifying calculation only 8 bits are taken
    MOVF    PRODL,W
    ADDWF   TEMP,F
    CLRF    WREG
    ADDWFC  PRODH, W               ;Lower 8 bits are discarded = divide result by 0xFF
    MOVWF   SPEED_REF_RPMH         ;Speed reference loaded in
    MOVFF   TEMP,SPEED_REF_RPML    ;SPEED_REF_RPM<H:L>
    RETURN
```

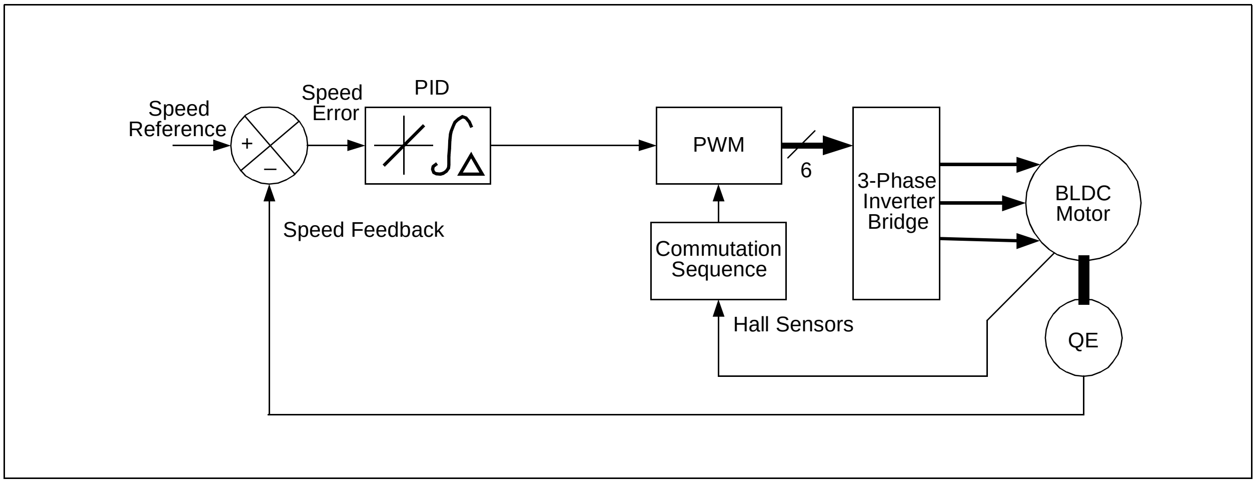
A simplified flow chart of the speed error calculation and updating the PWM duty cycle is shown in Figure 6.

FIGURE 6: SPEED ERROR CALCULATION


Closed-Loop Control
Speed Ref in RPM = Rated Motor Speed x Speed Ref / Max. Speed Ref
(S Ref)
Speed in RPM = Fosc/4 / Timer5 x Timer5 Prescale x Rotor Pole Pairs x 6 x 60
(S Actual)
Error (E) = S Ref – S Actual
PID_Error = K P x E + K I x ∫E + K D x ΔE
New PWM = PWM_old + PID_Error
Return

The difference between the speed reference and actual speed values give the error in speed. The error may be positive or negative, indicating the speed is more or less than the set reference. This error is passed through a PID algorithm to amplify the error. The amplified error is used to readjust the PWM duty cycles originally calculated as per Equation 2. Figure 7 shows a block diagram of a control loop for a closed-loop application. **Appendix A: “PID Controller”** gives some insight on step response and tuning PID gains.

FIGURE 7: CONTROL BLOCK DIAGRAM



CURRENT CONTROL

Motor phase current is measured using on-board current sensors, U6, U9 and U10 (optional). The Hall current transformer isolates the current signals with respect to the power circuits. These signals are connected to three Analog-to-Digital Converter Channels on PIC18F4431.

Motor currents are read every fixed interval of time. For constant torque application, the actual current is compared with the set torque reference. The error is amplified using PID algorithm. The proportional, integral and derivative gains are adjusted to get the best transient and steady state responses. This amplified error is used to readjust the PWM duty cycle, calculated earlier, for speed control.

At time of publication, the code included with this application note is Version 1.0. This version of the code does not include a closed current loop operation example as it is being considered as a future enhancement; however, future versions of the code may include this update. The example code is available from the Microchip web site (www.microchip.com).

OVERCURRENT PROTECTION

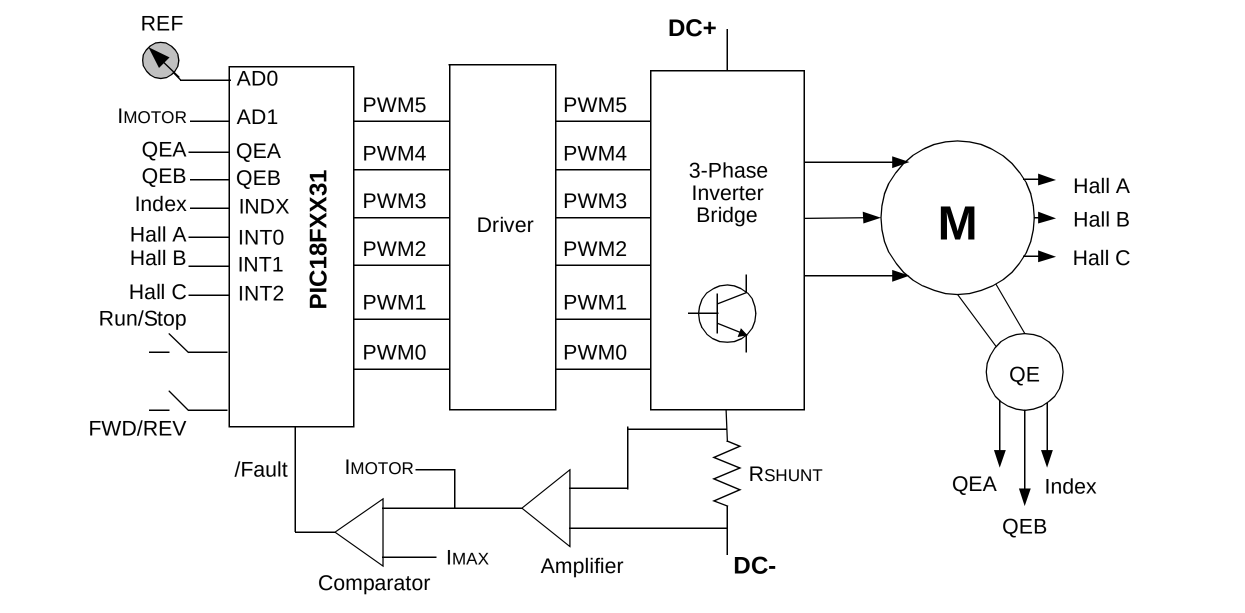
In addition to this, these three currents are added together and compared with a predefined voltage using a comparator. Output of this comparator is connected to the Fault A (/FaultA) pin on the PIC18F4431. The Fault input to the PCPWM module has the capability of putting PWM outputs to an inactive state upon detection of a Fault (Fault signals are active-low). The Fault input has two modes of operation: the first is Catastrophic mode, where the PWM is placed into an inactive state upon a Fault detection until the firmware clears the Fault status bit. The second mode is Cycle-by-Cycle mode. In this mode, the output will be inactive as long as a Fault exists; when the Fault is cleared on the pin, the PWM outputs becomes active in the following PWM cycle. When the system is operational, due to instantaneous current changes, the condition may look like an overcurrent; however, the condition may prevail a few hundredths of a microsecond to a few milliseconds. This condition is harmful if it repeats many times within a short duration of time. The firmware checks for Overcurrent Fault, as explained in the section **"Software Functions"**. With this, any spurious overcurrent signals due to noise can be eliminated and protection to the power circuit and motor is given in case of current exceeding the limit.

CLOSED-LOOP SPEED CONTROL USING OPTICAL ENCODER

An Optical Encoder (also known as the Quadrature Encoder) mounted on the motor shaft can give speed, relative or absolute position and direction information. This information can be used for improving the performance of BLDC motor control. Encoders give 3 signals, Channel A (QEA), Channel B (QEB) and Index. QEA and QEB are 90 degrees out of phase and Index is one single pulse per revolution, which can be used for homing and relative positioning. The PIC18FXX31 family of microcontrollers have a built-in Quadrature Encoder Interface (QEI) module in the motion feedback peripheral.

Figure 8 shows a block diagram showing closed-loop control of a BLDC motor using the Quadrature Encoder. Hall Sensors are used for commutation. Pins for the IC module and the QEI module are shared, so these can be used mutually exclusive of one another.

FIGURE 8: BLOCK DIAGRAM FOR CLOSED-LOOP CONTROL USING QUADRATURE ENCODER



USING EXTERNAL INTERRUPT PINS FOR HALL SENSOR

Hall Sensors can be alternatively connected to the external interrupt pins (INT0, INT1 and INT2). These pins can cause interrupts on the rising or falling edge, based on the respective “Interrupt Edge Select” bits (INTEDG<2:0> in the INTCON2 register). In external interrupt ISR, the interrupt edge select bit should be toggled in the correct direction to get interrupts on both the falling edge and rising edge on all three INT pins.

QUADRATURE ENCODER INTERFACE PERIPHERAL

The Quadrature Encoder Interface has two main modes: Position Measurement mode and Velocity Measurement mode. Position Measurement modes are used for measuring the position of shaft with respect to index pulse, or with respect to a count loaded in the MAXCOUNT register. The position counter can be updated every QEA transition or every QEA and QEB transition. Upon the position being reached, an interrupt is generated.

In Velocity Measurement mode, Timer5 is counted between two QEA transitions or every QEA and QEB transition, and transferred to the Velocity register (VELR<H:L>). This VELR register value is used for determining the speed of the motor. When the motor is running at very low speeds, or if the number of Pulses Per Revolution (PPR) of the encoder used are very low, the Timer5 count may overflow. Timer5 has a software selectable input pulse prescaler, up to 1:8. In addition to this, a pulse reduction ratio of up to 1:64 can be given to the Timer5 count to avoid repeated overflows. An ERROR bit in the QEICON register indicates the overflow/underflow of the count.

Speed can be calculated from the Timer5 count using Equation 5. Speed depends upon the encoder PPR, Velocity Measurement Update mode, velocity pulse reduction ratio, Timer5 prescale and operating frequency.

The reference speed is calculated as previously shown in Equation 4. Error in speed is the difference between the reference speed and the actual speed. Care should be taken to have both reference and feedback in the same platform. This error is amplified using a PID algorithm. The amplified error is used to calculate a PWM duty cycle and is added or subtracted to the duty cycle calculated previously from the speed reference.

Example 5 shows calculating the speed from the Timer5 count. Example 7 shows calculating the speed error.

EQUATION 5: CALCULATING SPEED FROM VELOCITY REGISTER VALUE

$$\text{Speed in RPM} = \left(\frac{\text{Operating Frequency/4}}{\text{PPR x Velocity Update Rate x Pulse Reduction Ratio x Timer5 Prescale x VELR<H:L>}}\right) \text{x } 60$$

EXAMPLE 5: SYSTEM PARAMETER DEFINITIONS CODE EXAMPLE

```
#define    OSCILLATOR                d'20000000'    ;Define oscillator frequency
#define    ENCODER_PPR               d'1024'        ;PPR of Encoder on the motor
#define    TIMER5_PRESCALE           d'1'           ;Timer5 prescaler
#define    QEI_X_UPDATE              d'2'           ;Define the QEI mode of operation.
;If the velocity counter is updated only on QEA transition, then enable 2x mode
;If the velocity counter is updated every QEA and QEB transition, then enable 4x mode
;Define Velocity pulse decimation ratio
#define    VELOCITY_PULSE_DECIMATION d'16'
INSTRUCTION_CYCLE = (OSCILLATOR)/d'4'
RPM_CONSTANT_QEI = ((INSTRUCTION_CYCLE)/
(ENCODER_PPR*QEI_X_UPDATE*VELOCITY_PULSE_DECIMATION*TIMER5_PRESCALE)) * 60 ;In RPM
```

EXAMPLE 6: SPEED CALCULATION FROM VELOCITY REGISTER CODE EXAMPLE

```
CALCULATE_SPEED
;Velocity register value is loaded in VELOCITY_READ<H:L> registers
;Actual speed = RPM_CONSTANT_QEI/ VELOCITY_READ<H:L>
    MOVFF   VELOCITY_READH,ARG2H       ;Timer5 count is loaded to divisible
    MOVFF   VELOCITY_READL,ARG2L
    MOVLW   HIGH(RPM_CONSTANT_QEI)     ;Constant count is loaded to divisor
    MOVWF   ARG1H
    MOVLW   LOW(RPM_CONSTANT_QEI)
    MOVWF   ARG1L
    CALL    DIVISION_16BY16            ;16 bit/16bit division performed
    MOVFF   RESL,SPEED_FEEDBACKL       ;Result is the actual speed in RPM
    MOVFF   RESH,SPEED_FEEDBACKH       ;Stored in the SPEED_FEEDBACK
    RETURN                             ;registers
```

EXAMPLE 7: SPEED ERROR CALCULATION CODE EXAMPLE

```
;Speed Error = SPEED_REF_RPM - SPEED_FEEDBACK
    BSF     STATUS,C
    MOVF    SPEED_REF_RPML, W
    SUBFWB  SPEED_FEEDBACKL, W
    MOVWF   SPEED_ERRORL
    MOVF    SPEED_REF_RPMH, W
    SUBFWB  SPEED_FEEDBACKH, W
    MOVWF   SPEED_ERRORH
    BCF     FLAGS,NEGATIVE_ERROR       ;error is negative?
    BTFSS   SPEED_ERRORH, 7
    BRA     POSITIVE_ERROR             ;yes, complement the error
    COMF    SPEED_ERRORH, F
    COMF    SPEED_ERRORL, F
    BSF     FLAGS,NEGATIVE_ERROR       ;set the error flag to indicate negative error
POSITIVE_ERROR
;Calculate error PWM based on the speed error
;Error PWM = Error_PWM_constant(8bit) * Error(16bit)
    MOVLW   (ERROR_PWM_CONSTANT)       ;calculate the error in PWM
    MULWF   SPEED_ERRORL
    MOVFF   PRODH,TEMP
    MOVFF   PRODL,ERROR_PWML
    MOVLW   (ERROR_PWM_CONSTANT)
    MULWF   SPEED_ERRORH
    MOVF    PRODL, W
    ADDWF   TEMP, W
    MOVWF   ERROR_PWMH

    CALL    PID_ALGORITHM              ;call PID controller
RETURN
```

CONCLUSION

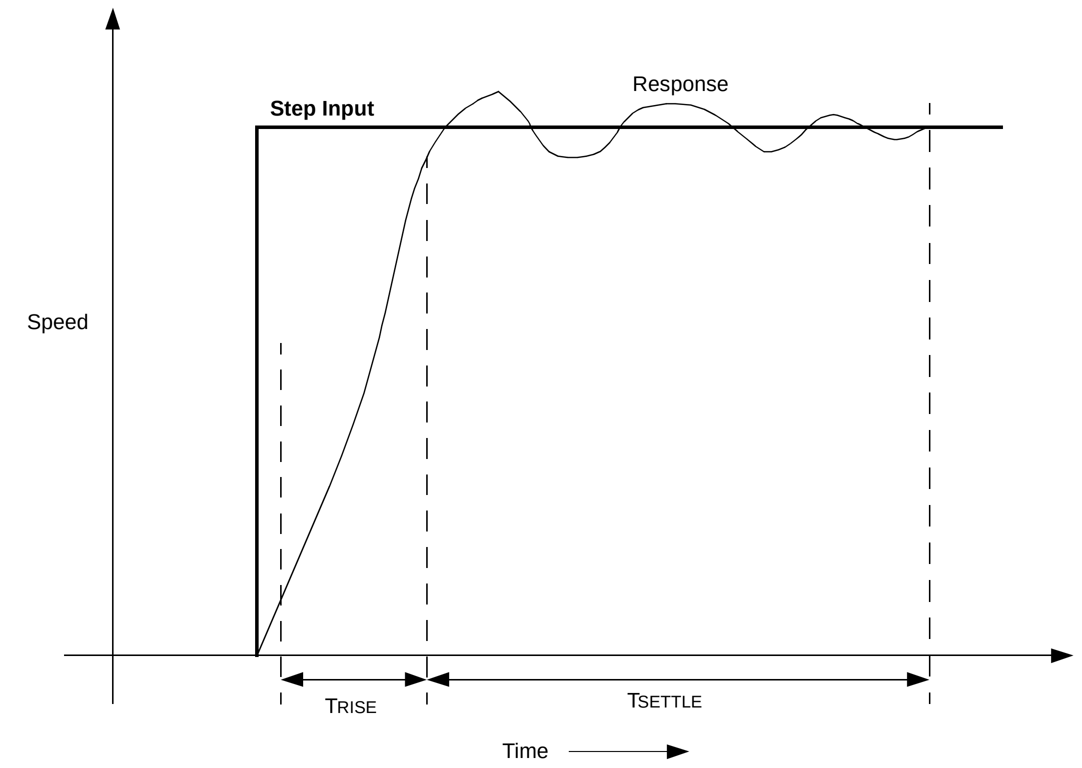
The PIC18F2331/2431/4331/4431 family of microcontrollers have peripherals that are well suited for motor control applications. Using these peripherals, speed control of a BLDC motor can be achieved with less overhead on the firmware. Closed-loop speed control is easy to implement as the microcontroller has a built-in motion feedback module.

APPENDIX A: PID CONTROLLER

The Proportional, Integral and Derivative gains should be adjusted according to the requirement. Figure A-1 shows a typical step response transient and study state for step input reference.

The rise time (TRISE) depends upon the rotor inertia and the load inertia. A typical response would be a 10% overshoot with respect to the input signal. The response should settle in about 2 to 3 subsequent overshoots and undershoots. Increasing the P gain will reduce the rise time and put the system into the steady state condition at a faster rate. But higher P gain will result in higher overshoot, which may put the system into a momentarily unstable situation. Changing the Integral gain will adjust the number of overshoots and undershoots around the steady state condition. Too high I gain may result in putting the system into an unbalanced condition. Too low I gain may make the system slow to reach the steady state position. The D gain slows the system down by adding a damping factor to the system. Normally, Derivative gain is kept at zero for the motor control. If the inertia of the load is too high, adding a small D component may help to put the system into a steady state position. P, I and D gains should be adjusted in such a way that the system has sufficient rise time and are short enough to settle to a steady state without any vibrations.

FIGURE A-1: TYPICAL SECOND ORDER STEP RESPONSE



APPENDIX B: CIRCUIT SCHEMATICS

FIGURE B-1: PIC18F4431 DEMO BOARD SHEET 1 OF 7

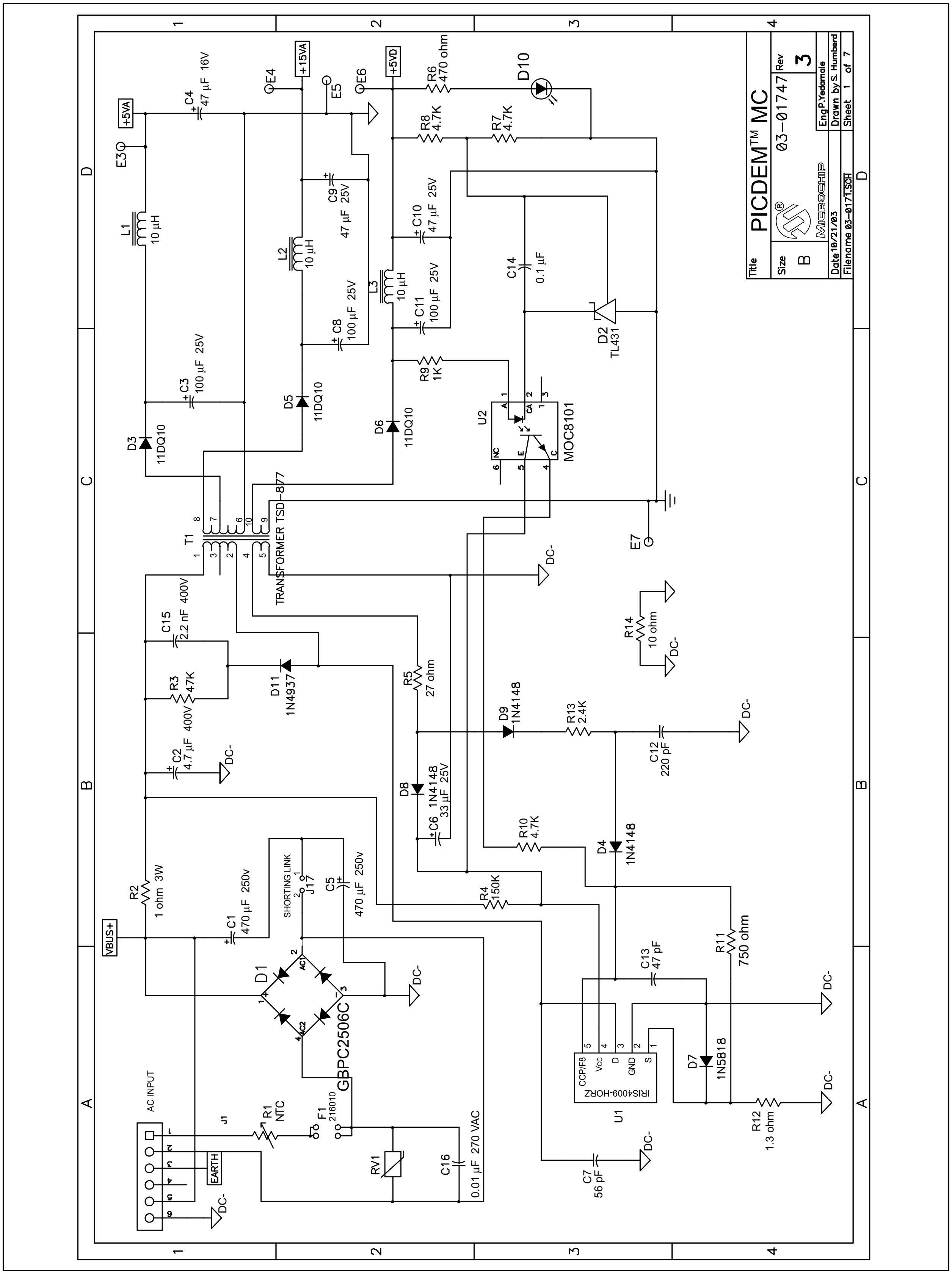


FIGURE B-2: PIC18F4431 DEMO BOARD SHEET 2 OF 7

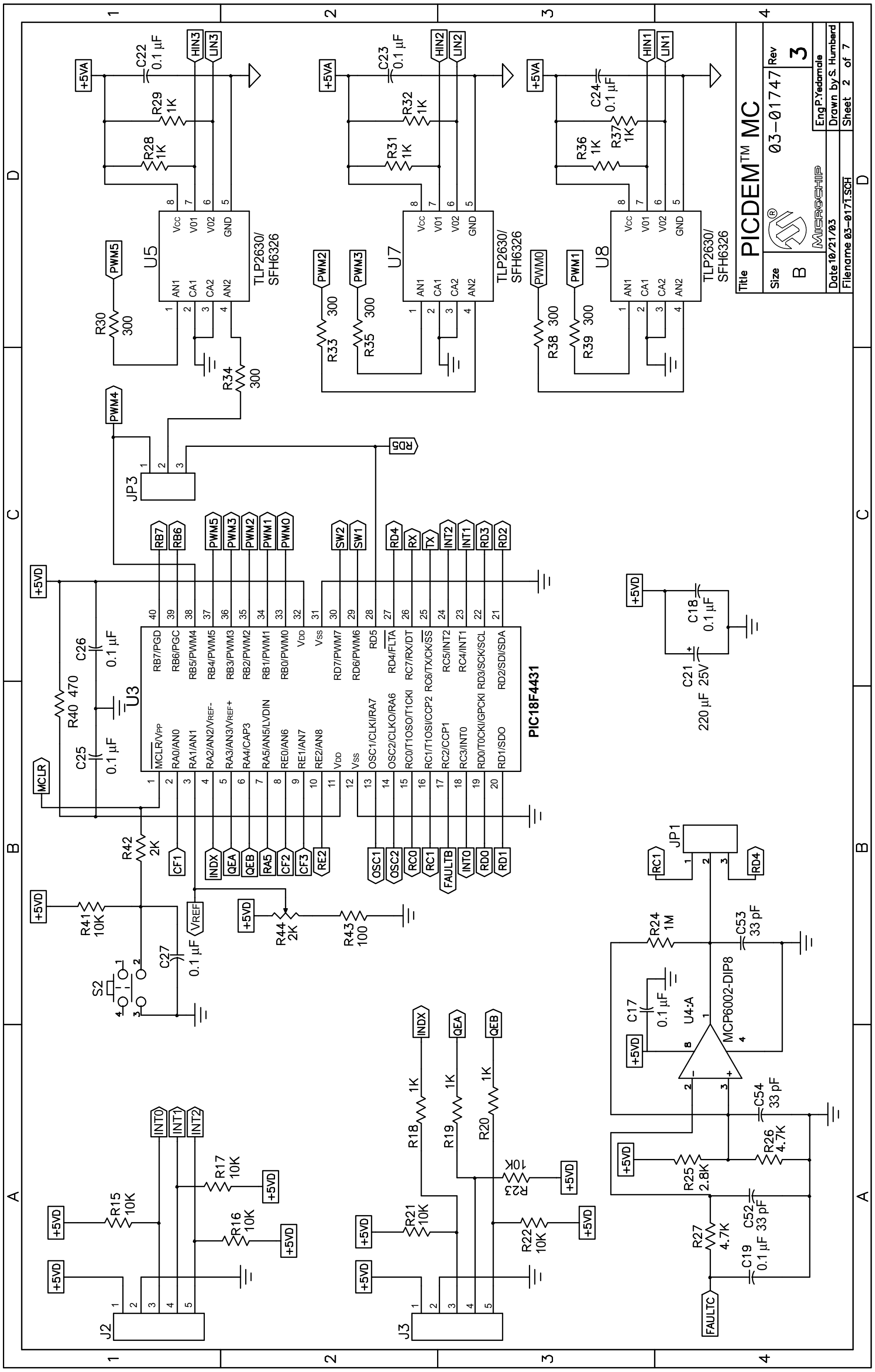

PICDEM™ MC
03-01747
Rev 3
Size B
Date 10/21/03
Filename 03-0171.SCH
Eng P.Yedamale
Drawn by S. Humbard
Sheet 2 of 7
MICROCHIP
U5
U7
U8
TLP2630/SFH6326
AN1 CA1 CA2 AN2 Vcc V01 V02 GND
+5VA
HIN1 LIN1 HIN2 LIN2 HIN3 LIN3
C22 0.1 µF
C23 0.1 µF
C24 0.1 µF
R28 1K
R29 1K
R31 1K
R32 1K
R36 1K
R37 1K
R30 300
R33 300
R34 300
R35 300
R38 300
R39 300
PWM0 PWM1 PWM2 PWM3 PWM4 PWM5
JP3
RD5
U3
PIC18F4431
MCLR/VPP RA0/AN0 RA1/AN1 RA2/AN2/VREF- RA3/AN3/VREF+ RA4/CAP3 RA5/AN5/LVDIN RE0/AN6 RE1/AN7 RE2/AN8 VDD VSS OSC1/CLKI/RA7 OSC2/CLKO/RA6 RC0/T1OSO/T1CKI RC1/T1OSI/CCP2 RC2/CCP1 RC3/INT0 RD0/T0CKI/GPCKI RD1/SDO
RB7/PGD RB6/PGC RB5/PWM4 RB4/PWM5 RB3/PWM3 RB2/PWM2 RB1/PWM1 RB0/PWM0 VDD VSS RD7/PWM7 RD6/PWM6 RD5 RD4/FLTA RC7/RX/DT RC6/TX/CK/SS RC5/INT2 RC4/INT1 RD3/SCK/SCL RD2/SDI/SDA
RB7 RB6 SW2 SW1 RD4 RX TX INT2 INT1 RD3 RD2
+5VD
C25 0.1 µF
C26 0.1 µF
R40 470
MCLR
R42 2K
CF1 INDX QEA QEB RA5 CF2 CF3 RE2
OSC1 OSC2 RC0 RC1 FAULTB INT0 RD0 RD1
R41 10K
VREF
C27 0.1 µF
S2
R44 2K
R43 100
C18 0.1 µF
C21 220 µF 25V
JP1
RC1 RD4
R24 1M
C53 33 pF
C17 0.1 µF
U4:A
MCP6002-DIP8
C54 33 pF
R26 4.7K
R25 2.8K
R27 4.7K
C52 33 pF
C19 0.1 µF
FAULTC
R18 1K
R19 1K
R20 1K
R3 10K
R21 10K
R22 10K
R15 10K
R16 10K
R17 10K
INT0 INT1 INT2
J2
J3
A B C D
1 2 3 4

FIGURE B-3: PIC18F4431 DEMO BOARD SHEET 3 OF 7

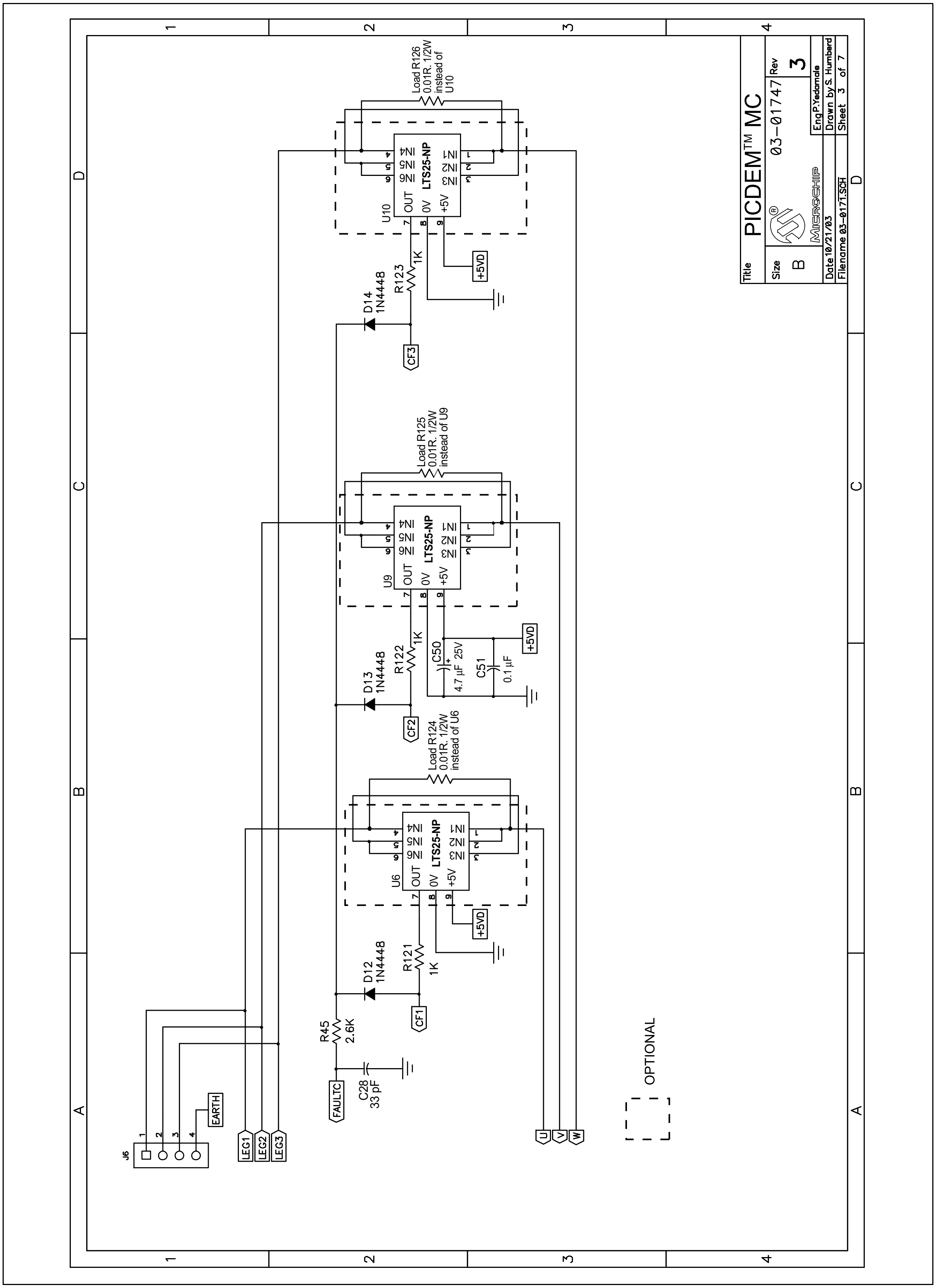


FIGURE B-4: PIC18F4431 DEMO BOARD SHEET 4 OF 7

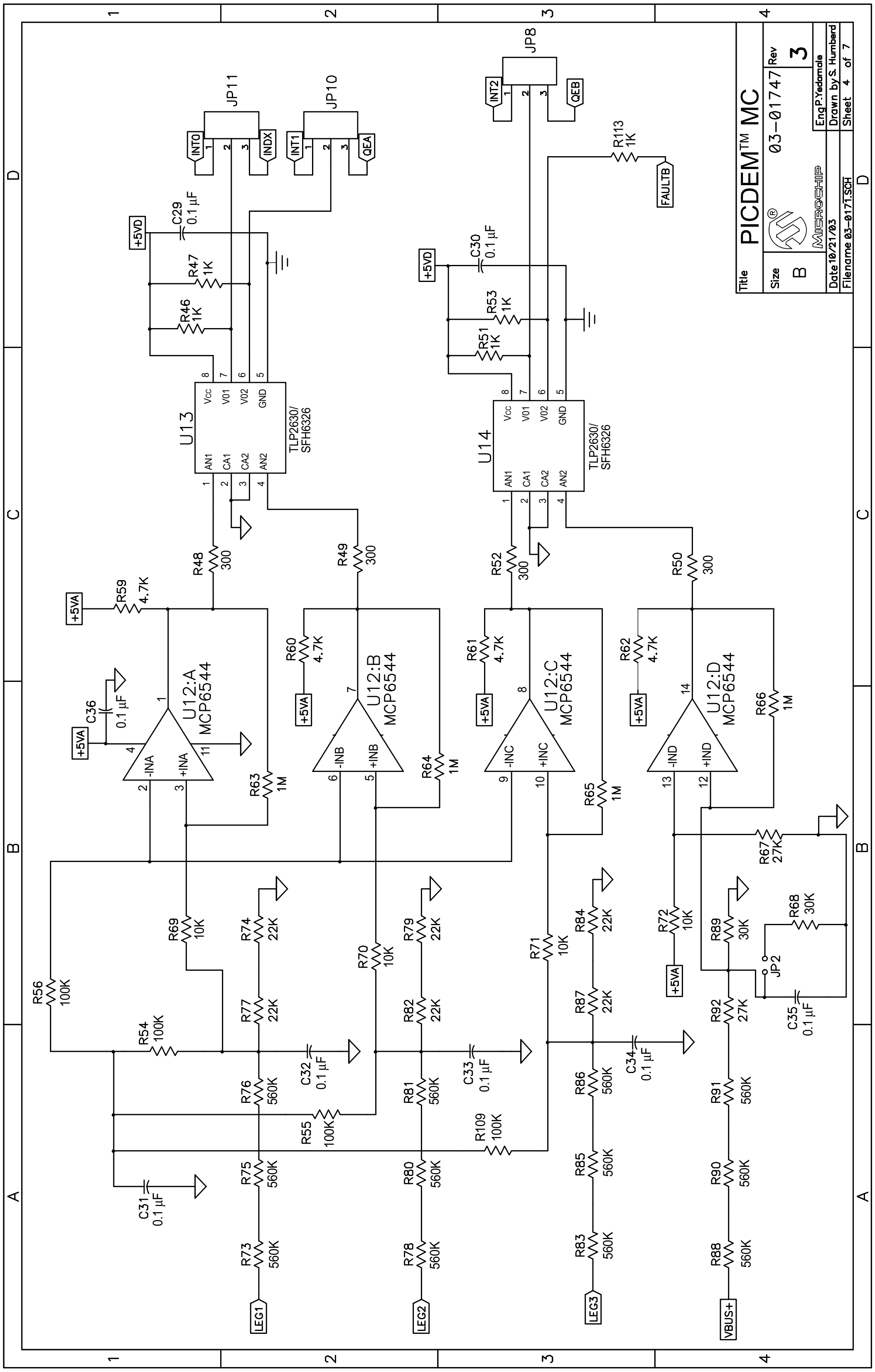


FIGURE B-5: PIC18F4431 DEMO BOARD SHEET 5 OF 7

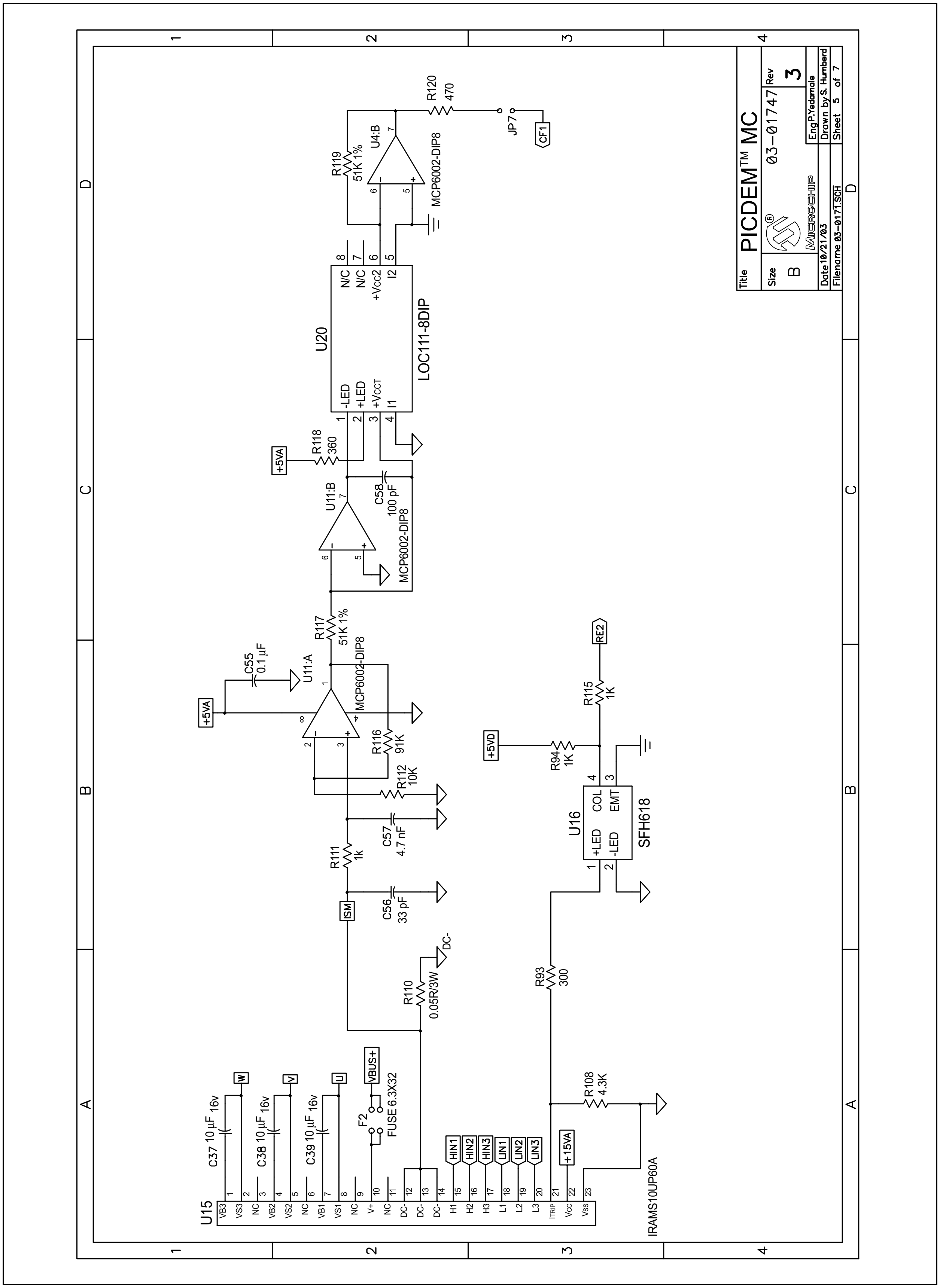


FIGURE B-6: PIC18F4431 DEMO BOARD SHEET 6 OF 7

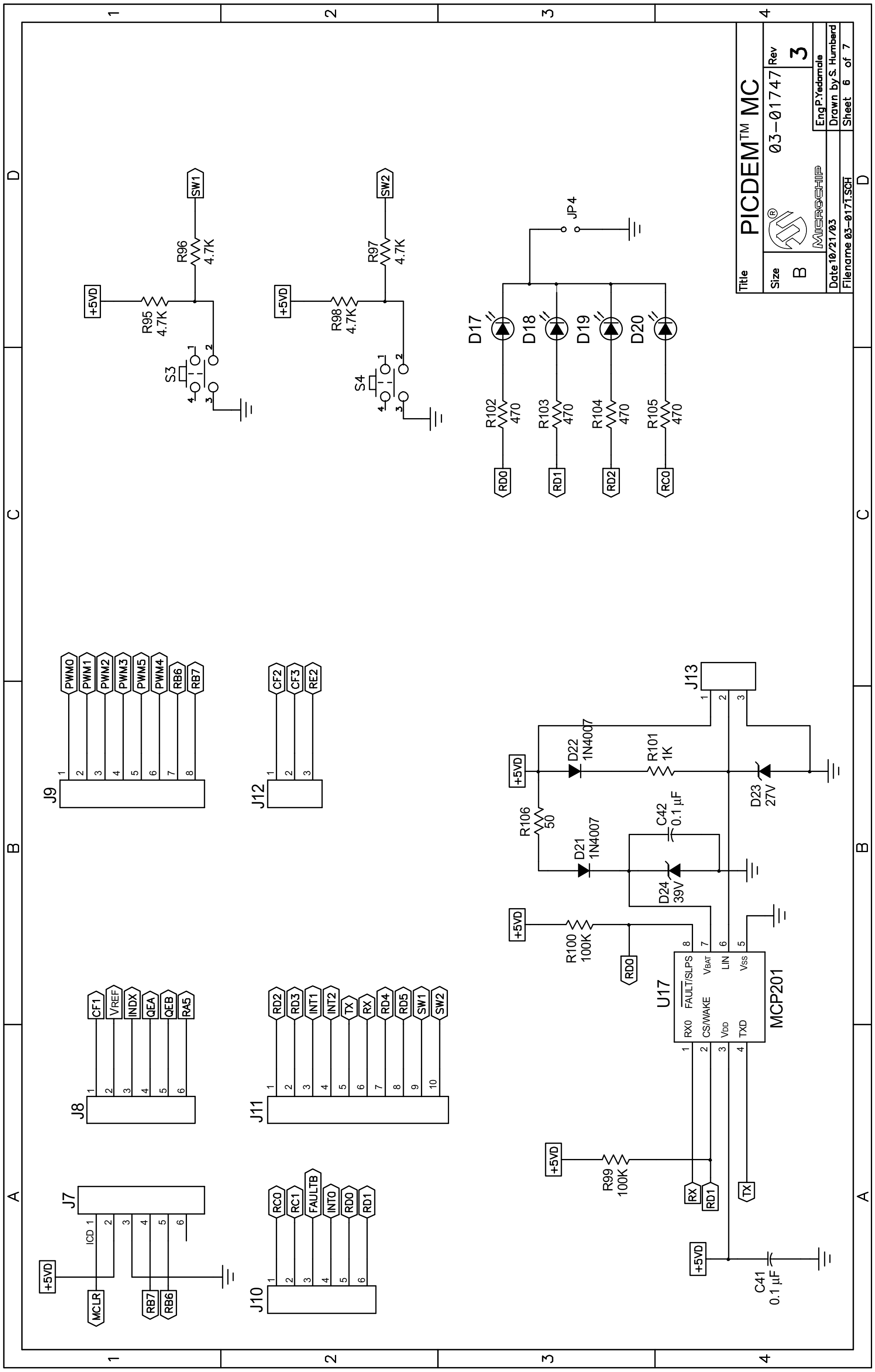
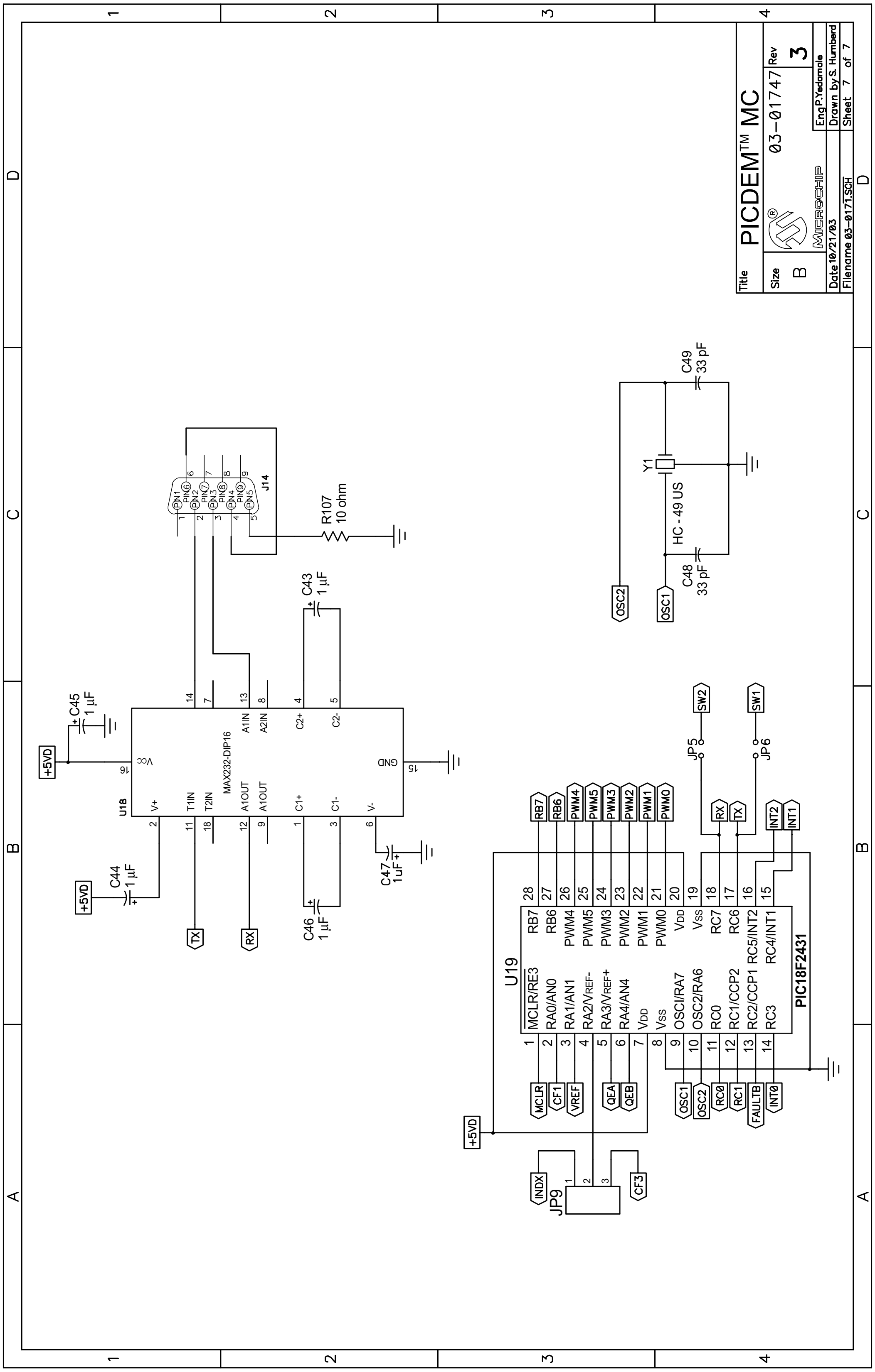


FIGURE B-7: PIC18F4431 DEMO BOARD SHEET 7 OF 7



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.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartShunt and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPIC, Select Mode, SmartSensor, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

♻ Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company’s quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

Phoenix
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966
Fax: 480-792-4338

San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/08/04