
PIC-SERVO SC 3PH Motion Control Board

For 3-Phase Brushless & Brush-type DC Motors

(P/N: KAE-T0V10-BD3PHV1)

The **PIC-SERVO SC 3PH** Motor Control board is a complete servo control system with the following features:

- **PIC-SERVO SC** motion control chip providing servo control for motors with incremental encoders, including trapezoidal profiling, velocity profiling and support for coordinated multi-axis motions.
- On-board amplifier for 3-Phase brushless motors (with 3 hall effect sensors) *or* conventional brush-type motors.
- Amplifier capable of driving 6 amps continuously at up to 48vdc. Built-in thermal, overcurrent and undervoltage protection.
- Current sensing, active current limiting, and overvoltage protection.
- RS485 serial interface allows up to 32 **PIC-SERVO** controllers to be controlled from a single serial port. Connects to an RS232 port through commonly available full-duplex adapters or using the **Z232-485** converter board.
- Step & Direction inputs for control from stepper based indexing systems.
- Two limit switch inputs for overtravel protection.
- Its small size (3" x 3") allows it to be mounted near motors, reducing noise and simplifying wiring.
- Windows test software provided including 32 bit Windows DLL and C source code. DOS based C code and Basic code are also available.

1. Quick Start

What you will need:

PIC-SERVO 3PH Motor Control Board

Z232-485 Converter Board, **USB-NMC** Adapter, or equivalent

Brushless Motor, 3-phase with hall effect sensors and incremental encoder

-or-

Conventional DC brush motor with incremental encoder

Motor power supply (12v min. - 48vdc max.)

Logic power supply (7.5 - 12vdc, 500 ma)

Motor & encoder cables

10 pin flat ribbon cable with standard IDC socket connectors at both ends

Straight DB9 male / DB9 female cable to PC COM port

PC compatible computer running Windows

Test software - NMCTest for Windows95/98/2000/NT/XP

(available for download from www.jrkerr.com)

CAUTION

The **PIC-SERVO SC 3PH** Motion Control Board does not incorporate safeguards for fail-safe operation. As such, this board should not be used in any device which could cause injury, loss of life, or property damage. **JEFFREY KERR, LLC** makes no warranties whatsoever regarding the performance, operation, or fitness of this board for any particular purpose.

Most of the cables are available from computer or electronics stores. However, you will probably have to make your own motor/encoder cable to connect to your particular motor. Refer to *Section 2.1* for the connector pin definitions.

To start off, connect your encoder to connector P3 and your hall effect sensors to connector P5. If you are using a brush-type DC motor, do not connect anything to connector P5. Your motor leads will connect to the three-position screw terminals marked P2. If you are using a brush-type motor, you will connect the leads to just the first two positions on this screw terminal. Please see *Section 2.1* for connection details.

Because standards vary, you may have to re-arrange your motor leads by trial and error once you start testing. Even with brush-type motors, there is a 50-50 chance you will have to reverse the motor leads to get the polarity right.

Interconnections and Jumpers:

Basic interconnections and jumpers are shown in *Figure 1* for both a single controller and for a multiple controller configuration. On the **Z232-485** converter, jumpers JP3 and JP4 are installed in the 1-2 position for use as a simple converter. (Please refer to the **Z232-485** documentation for use with the optional standalone processor cards.) Jumper JP5 is installed to distribute logic power to the controller boards over the communications cable. Logic power (7.5 - 12vdc) is supplied on connector JP6. (If you are using a different type of serial port adapter, you may supply power to connector JP8 on the **PIC-SERVO SC 3PH** board.)

On the **PIC-SERVO SC 3PH** controller board, jumpers JP6 and JP7 are installed to connect logic power supplied by the communications cable to the board's logic supply. In the *single* controller configuration, the three jumpers labeled JP3, JP4 and JP5 should be installed as shown. In the *multiple* controller configuration, these jumpers should only be installed on *last* controller, furthest from the PC host. On all intermediate controllers, jumpers at JP3, JP4 and JP5 should be left *uninstalled*.

Motor power should be connected to the two position screw terminals, P1, with 12 - 48vdc connected to the terminal towards the upper edge of the board and GND connected to the terminal towards the center as shown in *Figure 1*. Because the logic power supply, the motor power supply, and your host computer are all connected with a common ground, we recommend that your motor power supply and your logic power supply have floating outputs to avoid ground loops.

Loading and Running Software:

First unzip NMCTEST.ZIP into a single directory. Before starting up the test code, make sure all of your jumpers and interconnections are as shown in *Figure 1*. Also make sure you have logic power supplied to the **Z232-485** converter.

Run the program NMCTest.exe. Select the correct COM port when prompted (leaving the default baud rate at 19200 for now). If you are using a different COM port, you will get an error message saying no modules were found. If this is the case, click on the Reset Network button and set the COM port to the correct value. The program will attempt to locate controllers on the RS485 network and will respond with the number of controllers found. If the number of controllers reported does not match the number connected, re-check the interconnections, jumpers and power, and then try again.

The list box on the left side of the window will display the list of motors found. Module 1 will be the last controller which is furthest from the host PC. Clicking on different controllers will display the status and controls for that particular motor. Click on the **PIC-SERVO** module you are testing and spin the motor shaft by hand. See that the position changes accordingly in the status panel.

Before testing the servo, make sure that the motor is disconnected from any mechanism so that it is able to spin freely. To test the servo, first turn on the motor power. You should see the Motor Power box checked in the status panel. Next, click on the Enable Amplifier box in the Motion Command panel.

Next, you will need verify that your brushless motor leads are connected in the proper order, and rearrange them if they not. (Skip to the next paragraph if you are using a brush motor.) Select PWM mode, type in a value of 64 (about 25% of full voltage), and click on the GO button. If the motor does not spin, click the Motor Off button and try a different arrangement of the motor leads (there are a total of 6 possible arrangements). If the motor does spin, try entering a value of -64, and see if the motor spins at approximately the same speed in the opposite direction. If it does not, you still don't have it right - try a different lead arrangement. When the motor is connected correctly, it will spin smoothly with equal ease in both directions* .

At this point, the commutation is correct, but the motor still may be rotating backwards in relation to the encoder. You can check this by looking at the motor velocity when you run in PWM mode with a PWM value of +64: the velocity reading should be *positive*[#] . If the velocity value reads *negative*, the polarity is reversed and you have two choices for fixing it: 1) you can swap the encoder Channel A and Channel B wires, effectively making the encoder count in the other direction, or 2) rearrange both the hall effect sensor wires *and* the motor lead wires until the commutation is correct *and* the polarity with respect to the encoder is correct. (For brush motors, you can simply reverse the encoder wires or reverse the motor leads to correct any polarity mismatch.)

Finally, click on the STOP! button to enable the position servo. Try turning the motor shaft by hand. (If the motor jerks and stops, or spins out of control, you probably still have the motor and encoder polarities reversed.) The motor should attempt to hold a fixed position. If it does, click on Pos mode, type in a position value of 1000, and then click on GO. The motor should move to position 1000 (or close to it, depending on how the gains are set). Try moving to several different positions until you are satisfied that the motor is moving as it should. (Note that if your motor has a gearhead, the motion of 1000 counts may produce an imperceptibly small motion, and you should use a larger number instead.)

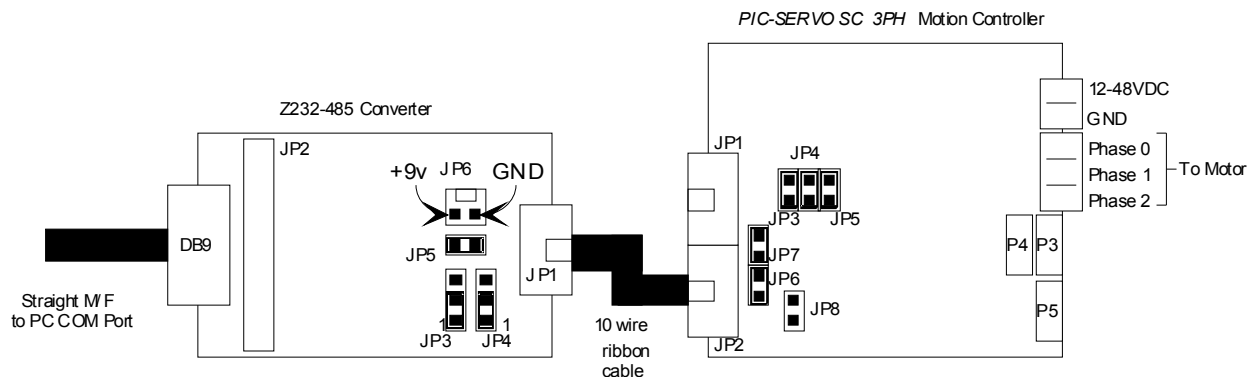
If the motor seems to buzz when it is stationary, try setting the Deadband Compensation value (DC) to 1 or 2 to compensation for amplifier non-linearity near zero output.

Otherwise, the control gains, and maximum velocities and accelerations are set to default values which are reasonable for most small motors. Please refer to the **PIC-SERVO SC** controller chip datasheet for details on the values for the gains, velocities and accelerations. The NMCTest online help also has a great deal of information about the **PIC-SERVO SC** controller.

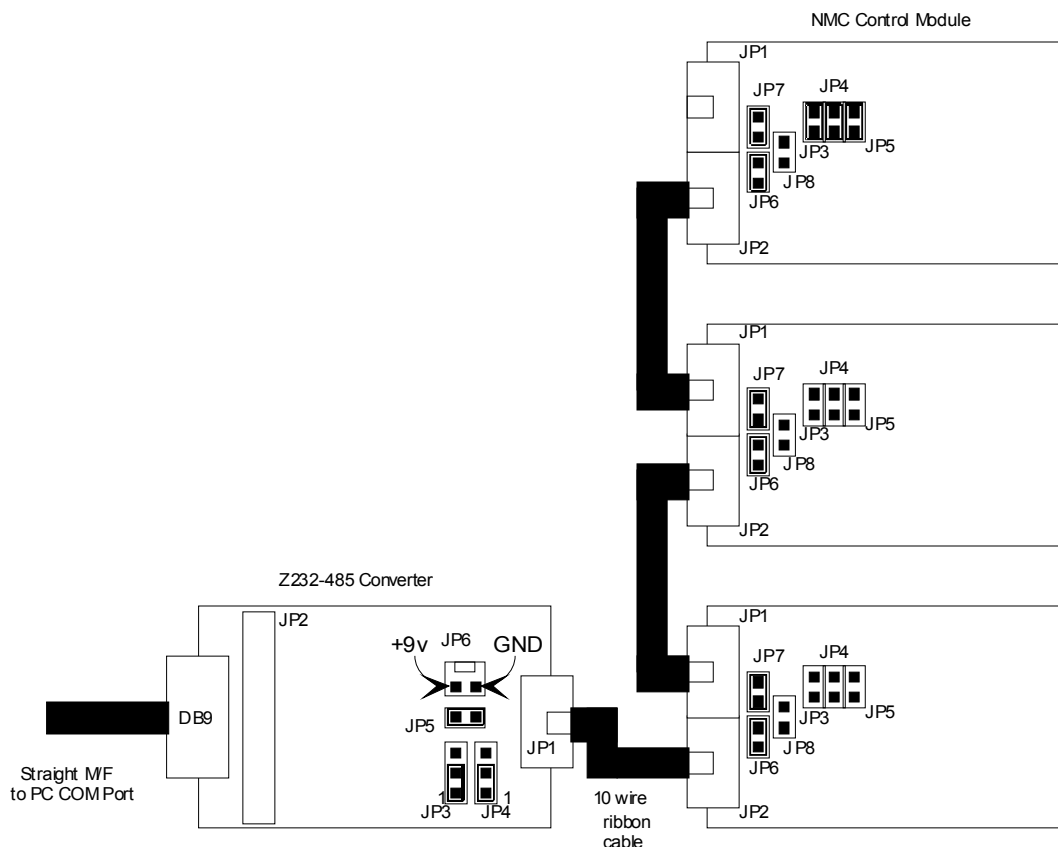
* If your motor does not spin smoothly with any of the six possible lead configurations, it may be that the **PIC-SERVO SC** is not set for 3-phase commutation. Click on the "Configure EEPROM" and check the 3-Phase Commutation option only and then click "Save data in EEPROM". Next click on "Reset Network" and start again.

With older versions of the **PIC-SERVO** chipset, the velocity should read *negative* for positive PWM values.

Single Controller Configuration



Multiple Controller Configuration



CAUTION: Connecting communications cables incorrectly, or installing jumpers JP3, JP4 and JP5 (on the *PIC-SERVO* board) in the wrong location may damage the *PIC-SERVO* or other NMC controller chips!

Figure 1 - Basic Interconnections.

2. Connectors and Jumpers

2.1 Pinouts

Motor Power Connector **P1** (2 position screw terminal)

<i>Pin</i>	<i>Definition</i>
1	Motor Power Supply 12 - 48vdc - <i>near top edge of board</i>
2	Motor Power Supply Ground (connected internally to logic ground)

Logic Power Connector **JP8** (1x2 pin header - 0.100" spacing)

(Use only if logic power is **not** supplied via the network communications cable.)

<i>Pin</i>	<i>Definition</i>
1	7.5 - 12vdc (towards the lower edge of the board)
2	Ground

Motor Connector **P2** (3 position screw terminals)

<i>Pin</i>	<i>Definition</i>
1	Phase 0 for 3-phase motors, M+ for brush-type motors
2	Phase 1 for 3-phase motors, M- for brush-type motors
3	Phase 2 for 3-phase motors

Hall Effect Sensor Connector **P5** (1x5 pin header - 0.100" spacing)

<i>Pin</i>	<i>Definition</i>
1	+5v output
2	Hall sensor 0 (This signal has an internal pull-up resistor to +5v)
3	Hall sensor 1 (This signal has an internal pull-up resistor to +5v)
4	Hall sensor 2 (This signal has an internal pull-up resistor to +5v)
5	Ground

Encoder Connector **P3** (1x5 pin header - 0.100" spacing)

<i>Pin</i>	<i>Definition</i>
1	+5v output
2	Channel A (This signal has an internal pull-up resistor to +5v)
3	Channel B (This signal has an internal pull-up resistor to +5v)
4	Index (optional) (This signal has an internal pull-up resistor to +5v)
5	Ground

Limit Switch Connector **P4** (1x5 pin header - 0.100" spacing)

<i>Pin</i>	<i>Definition</i>
1	Limit Switch 2 (reverse limit) <i>or</i> Step Input (if Step & Direction mode enabled) (This signal has an internal pull-up resistor to +5v)
2	Limit Switch 1 (forward limit) <i>or</i> Dir Input (if Step & Direction mode enabled) (This signal has an internal pull-up resistor to +5v)
3	Enable input used if in Step & Direction mode (connects to MCLR) (This signal has an internal pull-up resistor to +5v)
4	Servo Fault output if in Step & Direction mode (connects to ADDR_OUT)
5	Ground

Network Connectors JP1, JP2 (2x5 pin header - 0.100" spacing)

<i>Pin</i>	<i>Definition</i>
1	RCV+
2	RCV-
3	XMT+
4	XMT-
5	ADDR_IN on JP1, ADDR_OUT on JP2
6	Ground
7	Logic power (7.5 - 12vdc)
8	Ground
9	Logic power (7.5-12vdc)
10	Ground

2.2 Jumpers

<i>Jumper</i>	<i>Description</i>
JP3	Connects ADDR_IN to GND. Insert jumper for the last controller board on the network (or if only 1 controller is used)
JP4, JP5	Enables termination resistors on RX and TX. Insert these jumpers for the last controller on the network (or if only 1 controller is used).
JP6,JP7	Logic power interconnection. Inserting JP6 connects logic power to network connector JP2. Inserting JP7 connects logic power to JP1. These are used to control the distribution of logic power over the network cables. Normally both these jumpers are installed.

3. PIC-SERVO Motor Control Board Description

The **PIC-SERVO 3PH** Motor Control board is a complete motor servo control system including a servo controller, amplifier, serial communications interface, optical encoder interface, and limit switch inputs. The board is designed so that up to 32 controllers can be connected directly to a single standard serial port (using an RS232-RS485 converter if necessary).

3.1 PIC-SERVO SC Controller Chip

The **PIC-SERVO SC** controller chip is the core of the controller board. It encompasses all of the logic for encoder counting, servo control, motion profiling, phase commutation, fault detection and serial communications. It also has optional Step & Direction inputs for interfacing to stepper-type motion profiling systems, and an on-board EEPROM for storing configuration information. Please refer to the **PIC-SERVO SC** controller chip data sheet for complete information on the servo control operation and command set for your **PIC-SERVO SC 3PH** controller board.

Note that earlier versions of the **PIC-SERVO** chipset (v.5 and earlier) cannot be used in the **PIC-SERVO SC 3PH** board.

3.2 Communications Interface

The **PIC-SERVO SC 3PH** uses an RS485 multi-drop interface for allowing multiple control modules to communicate over the same RS485 communication port. The host computer sends commands out over a dedicated pair of transmit wires, and all status data comes back over a shared pair of receive wires. Because the host has a dedicated transmit line, a standard RS232 serial port can be used with a 4-wire RS242-RS485 (or RS232-RS422) converter.

With multiple controllers on a single network, each controller must have a unique address for receiving commands. Rather than using dip switches or jumpers to assign addresses, the **PIC-SERVO SC 3PH** uses a method of daisy-chaining an ADDR_IN signal and an ADDR_OUT signal for dynamically assigning addresses. With the controllers interconnected as shown in *Figure 1*, the ADDR_OUT signal of one board is connected to ADDR_IN of the next board. The very last board has ADDR_IN jumpered to GND. On power-up, all boards with ADDR_IN held high will have their communications disabled. Therefore, only the last board will be able to communicate with a default address of 0.

To initialize the network, a command is sent to the last controller (with address 0) to change its address to a value of 1. This has the side effect of causing its ADDR_OUT to lower, enabling communications with the next controller. The next command sent to address 0 will now be sent to the second-to-last controller. This process of assigning addresses is repeated until all controllers have been given unique addresses.

3.3 Amplifier

The amplifier on the **PIC-SERVO SC 3PH** board consists of three LMD18200 I.C.'s configured as half-bridge drivers. The amplifier is capable of driving 3-phase brushless motors or conventional brush-type DC motors. The commutation logic will check to see if hall effect sensors are connected to connector P5. If they are, it commutates the three phase outputs for driving a 3-phase brushless motor assuming a 120 degree hall sensor orientation. If nothing is connected to P5 (*i.e.*, if it reads hall sensor input values of 1, 1, 1), the commutation logic will only drive output phases 0 and 1 (pins 1 & 2 of P2) to work with a regular 2-wire brush-commutated DC motor.

The three LMD18200 amplifier chips may be screwed to an aluminum heatsink or to a metal chassis for heat dissipation. The metal tabs of these chips are all connected internally to ground. You should use an electrical insulator (Kapton tape works well) between the tabs and your chassis if you do not want the chassis connected to ground. If you are driving less than 1/2 amp, however, you may be able to run the amplifier without heat sinks.

Overcurrent, Overvoltage, Undervoltage & Thermal Protection

The LMD18200 amplifier chips on the **PIC-SERVO SC 3PH** board have internal overcurrent protection circuitry which prevents absolute peak currents above 10 amps. Generally, however, the current is limited to about 7 amps and the thermal limitation on the continuous current is 6 amps. You can also use the current limiting feature of the **PIC-SERVO SC** controller chip to limit the current to any value less than 7 amps by setting the Current Limit value according to the table below. Note that only *odd* current limit values should be used, unless you want to disable the **PIC-SERVO SC** chip's current limiting altogether.

Desired maximum current (amps)	Current limit value
no current limiting (limited to approximately 7A by the LMD18200)	0
1.0	43
2.0	83
3.0	121
4.0	159
5.0	199
6.0	239

The LMD18200 amplifier chips also have built-in thermal protection. Thermal shutdown occurs at 170 degrees C which will disable the amplifier and keep it disabled until the temperature drops below the threshold. Adequate heat sinking, as described above, is required to prevent thermal shutdown.

Undervoltage and overvoltage protection are provided by the **PIC-SERVO SC** chip. If the supply voltage drops below 10.8v, as might happen with a power supply fault, the **PIC-SERVO SC** will disable the amplifier and also disable the PID servo. If the voltage again rises above 10.8v, the PID servo will have to be explicitly re-enabled with a command to the **PIC-SERVO SC**. If the supply voltage rises above 54v, as might happen when a motor decelerates and acts as a generator, the **PIC-SERVO SC** will also disable the amplifier, but it will leave the PID servo enabled. This way, the servo will automatically regain control of the motor as the voltage drops back below 54v.

3.4 Limit Switch Inputs

The **PIC-SERVO SC 3PH** board has two limit switch inputs which can be used for overtravel protection. A normally closed limit switch should be connected between the limit switch input and ground. (The limit switch inputs are pulled-up internally to +5v.) When limit switch 1 is hit (*i.e.*, the switch is opened), forward motion of the motor will be inhibited. When limit switch 2 is hit, reverse motion will be inhibited. Note that the limit switch overtravel protection must be enabled with a command to the **PIC-SERVO SC**.

3.5 Step & Direction Inputs

The **PIC-SERVO SC 3PH** has the option of configuring the limit switch inputs as Step & Direction inputs instead. This is useful when using the **PIC-SERVO SC** controller and servo motor as a high performance replacement for a stepper motor and driver. If the Step and Direction inputs are being used, the limit switch inputs are unavailable, and you should connect your limit switches to your stepper indexing system instead.

When Step & Direction mode is enabled, the ADDR_OUT signal may be used to detect a servo fault condition. During normal operation, the ADDR_OUT output will be low. However, if in Step & Direction mode a servo fault condition is detected and the servo is disabled, the ADDR_OUT signal will go HI.

Also when in Step & Direction mode, it is useful for your stepper indexer system to be able to disable the **PIC-SERVO SC** through a TTL level signal. The MCLR input (internally pulled high to +5v) can be pulled low to disable the **PIC-SERVO SC**.

The Step, Direction, ADDR_OUT and MCLR signals all appear on connector JP9 for connecting to your stepper indexing system.

Lastly, the **PIC-SERVO SC** has an internal EEPROM which can be configured to make the board power-up ready to accept Step & Direction signals. The NMCTest program has a “Configure EEPROM” button which will allow you to store the proper startup parameters in EEPROM. When configuring the EEPROM, you should click on the “Restore Current Address”, “Enable Amplifier”, “Enable Servo”, “Enable 3-phase Commutation” and “Enable step/direction inputs” options and then save the parameters. When the board is next powered up, the servo will be enabled and ready to accept step and direction signals.

3.6 EEPROM Configuration

The **PIC-SERVO SC 3PH** board is shipped with the internal EEPROM set for 3-phase mode. (Note that 3-phase mode is always used with this board, even if a brush-type motor is connected.) With the 3-phase mode set in EEPROM, software applications do not need to use the I/O Control command to set this mode. If you ever replace the **PIC-SERVO SC** chip, however, you will probably want to use the NMCTest program (or your own software) to again save the 3-phase option in EEPROM.

3.7 Physical Dimensions

The **PIC-SERVO SC 3PH** board has 6 mounting holes for placing 4 mounting feet in one of two positions. The mounting holes at 2.45" x 1.80" match the pattern for other NMC controller boards, making it simple to stack the boards. The mounting holes at 2.45" x 2.3" provide better stability. The amplifier chips are all mounted along the back edge of the board for convenient attachment to a heatsink.

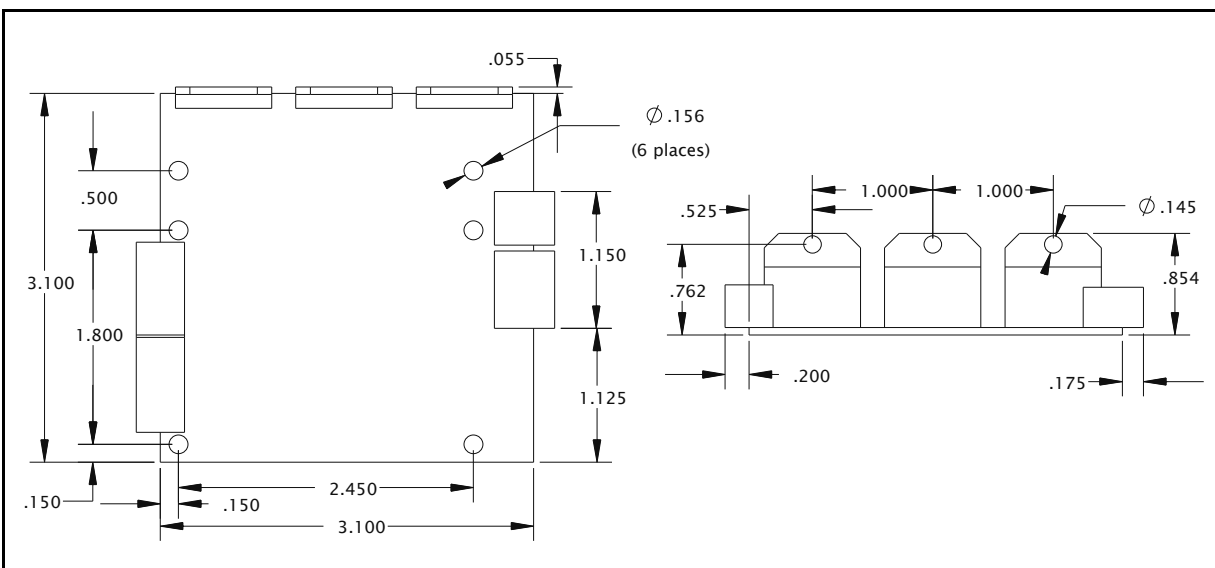


Figure 2 - **PIC-SERVO 3PH** Motor Control Board Dimensions

4. Contact Information

Additional information may be found from these sources:

JEFFREY KERR, LLC

www.jrkerr.com

Data sheets, application notes and test code may be downloaded from www.jrkerr.com/docs.html. Technical support is provided via e-mail. Please see www.jrkerr.com/contact.html for contact information.

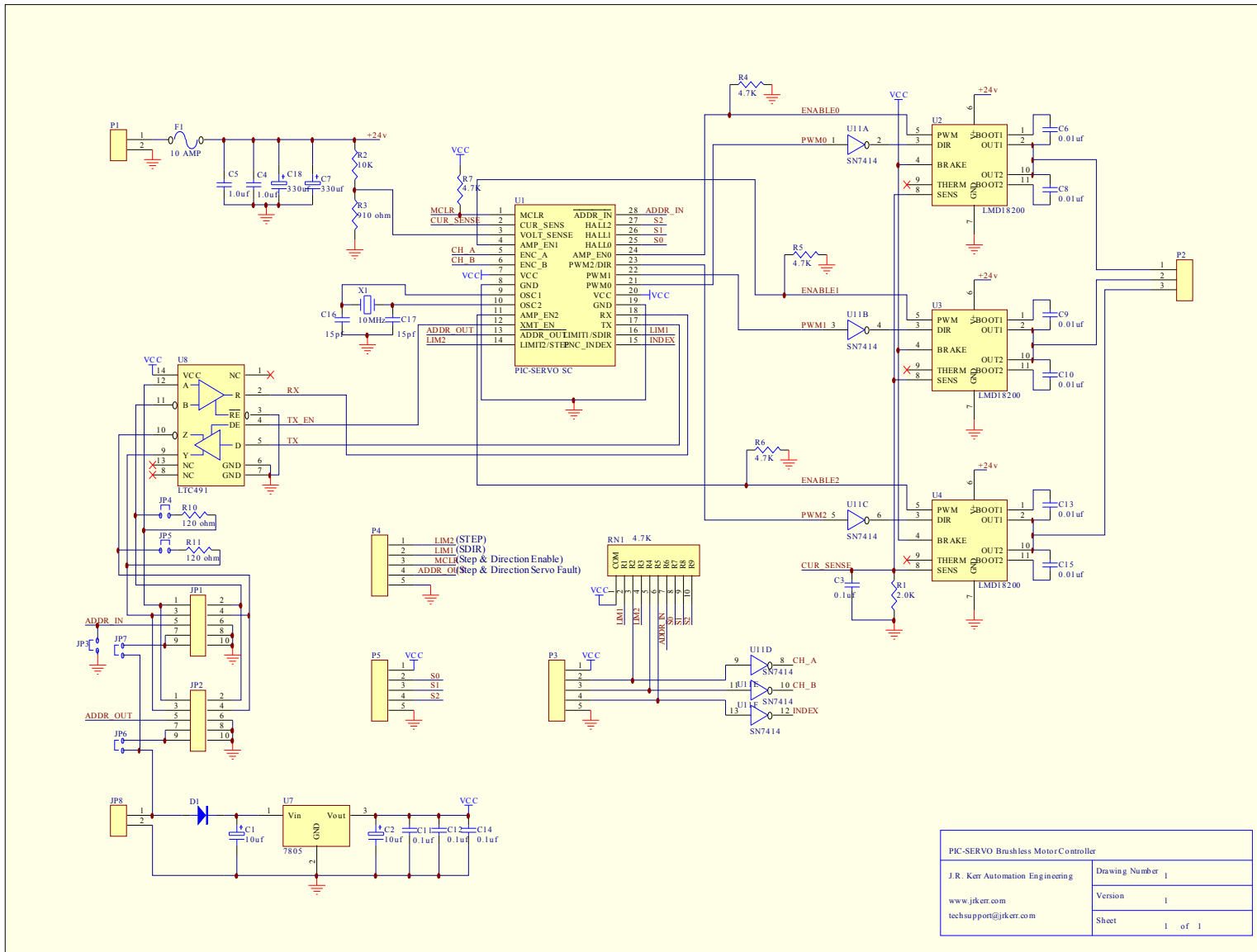


Figure 3 - PIC-SERVO SC 3PH Motor Control Board Schematic