______ *PIC-SERVO SC* Motion Control Board ____________________

For Brush-type DC Motors (P/N: KAE-T0V10-BDV1)

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

- ***PIC-SERVO SC*** motion control chip providing servo control of DC motors with incremental encoders, including trapezoidal profiling, velocity profiling and support for coordinated multi-axis motions.
- LMD18200 amplifier capable of driving 3 amps continuously, 6 amps peak at up to 48vdc. Built-in thermal, overcurrent and undervoltage protection.
- Current sensing, active current limiting, and overvoltage protection.
- PWM and DIR signals are provided for use with other external amplifiers.
- 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 (2" 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 SC Motion Control Board
Z232-485 Converter Board, ***USB-NMC*** Adapter, or equivalent
DC Motor (48v max., 3 amp continuous current max.) with TTL compatible encoder
Motor power supply (12v min. - 48vdc max.)
Logic power supply (7.5 - 12vdc, 500 ma)
Motor/encoder cable (DB15 male connects to your motor)
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)

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, you only need to connect M+, M-, Encoder A, Encoder B, Encoder +5v and Encoder GND. Other connections can be made as

CAUTION

The ***PIC-SERVO SC*** 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.

needed. Note that when testing, you may have to swap the M+ and M- leads to correct for the polarity of your motor.

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 is supplied on connector JP6. If you are using a different type of serial port adapter, you may attach power to the pins of JP8 on the ***PIC-SERVO SC*** board.

Learning: JP4 dikonek aja

On the ***PIC-SERVO SC*** 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 screw terminals, with 12 - 48vdc connected to the terminal towards the edge of the board and GND connected to the terminal towards the center as shown in *Figure 1*.

We recommend that both your logic power supply and your motor power supply have floating outputs. The ***PIC-SERVO SC*** board creates a common ground between the (-) side of the logic supply, the (-) side of the motor supply and the communications ground. With your power supply outputs floating, you will have a single ground referenced to your PC's ground, thus avoiding 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. ***PIC-SERVO*** 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 ***PIC-SERVO*** 1 and spin the motor shaft by hand. See that the position changes accordingly in the status panel.

Before enabling the motor servo make sure that the motor is disconnected from any mechanism which might be damaged. To test the motor, first turn on the motor power. You should see the

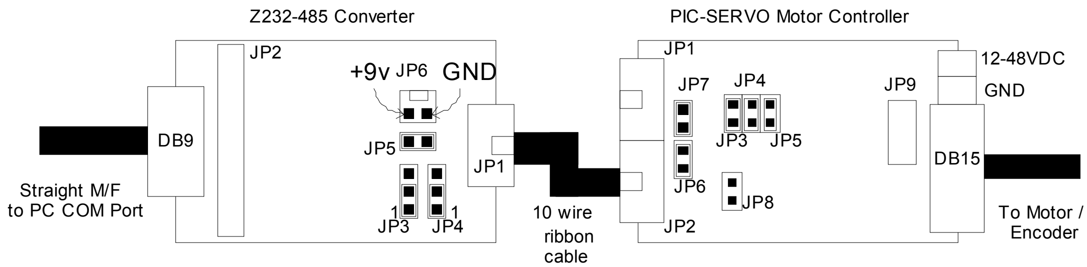
Motor Power box checked in the status panel. Next, click on the Enable Amplifier box in the Motion Command panel. Now click on the STOP! button. Try turning the motor shaft by hand. If the motor jerks and stops, or spins out of control, turn off the motor power and try swapping the M+ and M- leads on the motor. Turn the power back on, click on STOP! again. 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 a bunch of 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 (DC) value to 1 or 2 to compensate for non-linearity in the amplifier at close to zero output.

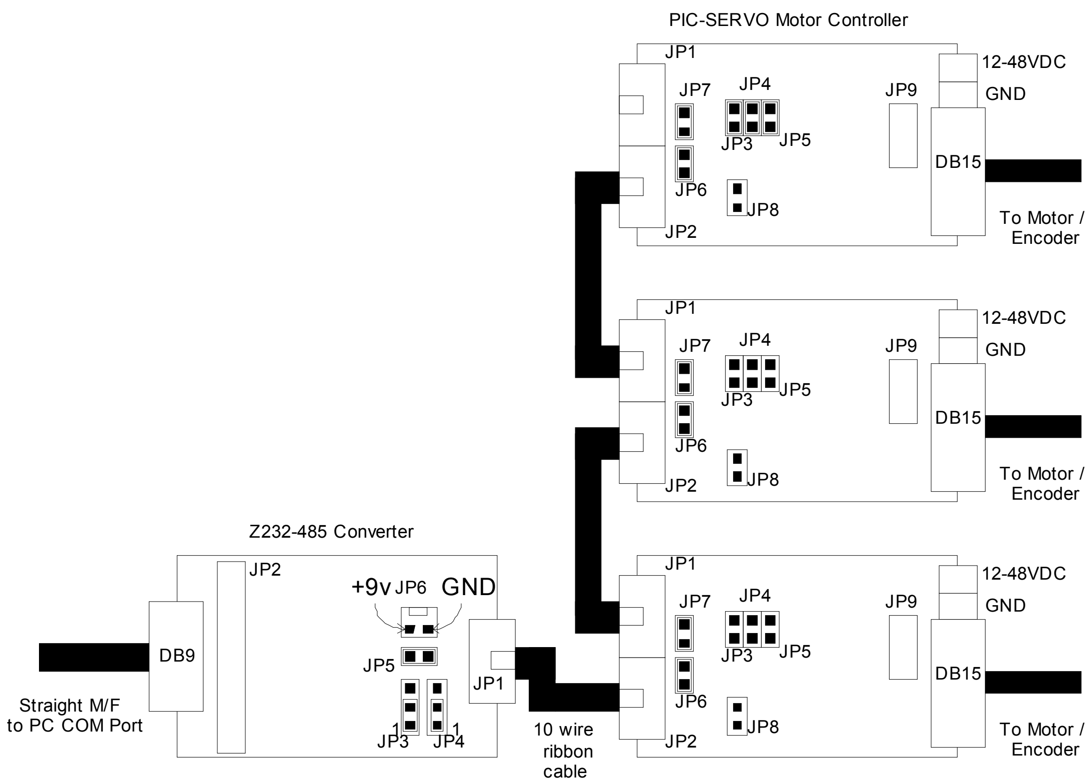
Otherwise, the control gains, and maximum velocities and accelerations are set to default values which are reasonable for most small motors. Please refer to the ***PIC-SERVO SC*** chipset data sheet for details on the values for the gains, velocities and accelerations. The NMCTest online help also has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

At this point, you will want to read the ***PIC-SERVO SC*** controller chip data sheet in detail to discover all of the features of your ***PIC-SERVO SC*** motion controller and for information on how to start developing your application.

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 SC* or other NMC controller chip!

Figure 1 - Basic Interconnections.

2. Connectors and Jumpers

2.1 Pinouts

Motor Connector P1 - DB15 Male

Pin	*Definition*
1	Motor Output (M+)
2	Motor Output (M+)
3	LED power - pulled up to 5v with a 330 ohm resistor (for use with opto-interrupt type switches)
4	Limit Switch 1 (pulled up to 5v with a 4.7k resistor)
5	Encoder Channel A (pulled up to 5v with a 4.7k resistor)
6	Encoder Channel B (pulled up to 5v with a 4.7k resistor)
7	Limit Switch 2 (pulled up to 5v with a 4.7k resistor)
8	Encoder Index (pulled up to 5v with a 4.7k resistor)
9	Motor Output (M-)
10	Motor Output (M-)
11	GND
12	GND
13	GND (supplied to encoder)
14	+5v (supplied to encoder)
15	GND

Network Connectors JP1, JP2 - 10 Pin Flat Ribbon IDC Sockets

Pin	*Definition*
1	***PIC-SERVO*** RCV+
2	***PIC-SERVO*** RCV-
3	***PIC-SERVO*** XMT+
4	***PIC-SERVO*** XMT-
5	***PIC-SERVO*** ADDR_IN on JP1, ADDR_OUT on JP2
6	GND
7	Logic power (7.5 - 12vdc)
8	GND
9	Logic power (7.5 - 12vdc)
10	GND

External Amplifier Connector JP9 - 8 Pin Single Row Locking Header

Pin	*Definition*
1	PWM output - 20 KHz square wave magnitude signal or Antiphase signal
2	Direction output (not needed if in Antiphase mode)
3	Amplifier Enable output (active high)
4	Limit2 input (same as P1, pin 7) *or* Step input if in Step & Direction mode
5	Limit1 input (same as P1, pin 4) *or* Direction input if in Step & Direction mode
6	MCLR input - can be used to enable/disable controller if in Step & Direction mode - pull low to disable controller (pulled up to 5v with a 4.7k resistor)
7	ADDR_OUT output - can be used as a servo fault indicator if in Step & Direction mode (signal goes high when servo is disabled)
8	GND

Motor Power Connector P2 - Screw Terminals

Pin	*Definition*
1	Motor Power 12 - 48vdc (at edge of board)
2	Motor Power Ground (connected internally to logic ground)

Logic Power Connector JP8 - 2 Pin Locking Header
(Use only if logic power is **not** supplied via the network communications cable.)

Pin	*Definition*
1	7.5 - 12vdc (pin towards the lower edge of the board)
2	Ground (pin towards the center of the board)

2.2 Jumpers

PIC-SERVO Motor Control Board Jumpers:

Jumper	*Description*
JP3	Connects ADDR_IN to GND. Insert jumper for the last ***PIC-SERVO*** on the network (or if only 1 ***PIC-SERVO*** is used)
JP4, JP5	Enables termination resistors on RX and TX. Insert these jumpers for the last ***PIC-SERVO*** on the network (or if only 1 ***PIC-SERVO*** 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 SC* Motion Control Board Description

The ***PIC-SERVO SC*** Motion 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/RS422 converter if necessary).

3.1 *PIC-SERVO SC* Motion Control Chip

The ***PIC-SERVO SC*** motion control chip forms the core of the controller, performing all of the tasks of servo control, motion profiling, communications and amplifier control. Please refer to the ***PIC-SERVO SC*** chip data sheet for complete details on the theory of operation of the servo control and motion profiling algorithms. Please see our web pages www.jrkerr.com/docs.html and www.jrkerr.com/software.html for a wide variety of information and software useful for developing your application.

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

3.2 Communications Interface

The ***PIC-SERVO SC*** 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 simple RS242-RS485 converter.

With multiple controllers on a single network, each controller must have a unique address for sending commands. Rather than using dip switches or jumpers to assign addresses, the ***PIC-SERVO SC*** 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 ***PIC-SERVO*** Motor Control board uses an LMD18200 H-bridge amplifier to drive DC brush-type motors with up to 3 amps continuously, 6 amps peak, with a supply voltage of 12 to 48vdc. This amplifier has overcurrent, overtemperature and undervoltage protection. The ***PIC-SERVO SC*** chip provides additional overvoltage protection for the amplifier. Note that if you are driving more than 500 ma, you will likely need to mount a heat sink to the tab of the amplifier. This tab is connected to GND, so it may also be bolted directly to a metal enclosure if your case ground is connected to your power ground.

The LMD18200 provides a current sense output which can be read by the host, and may be also used for current limiting by the ***PIC-SERVO SC*** chip. The board is configured so that the ***PIC-SERVO SC*** will produce an A/D reading of about 39 counts per amp of current. To enable current limiting, you should set the current limit value to an odd, non-zero value (ie, 39, 51, etc.). For example to limit the current to about 2 amps, you should set the current limit value to approximately 2 x 39, or 79.

The ***PIC-SERVO SC*** controller chip provides additional overvoltage and undervoltage protection for the amplifier. As configured on the ***PIC-SERVO SC*** controller board, if the motor supply voltage drops below 10.8v, the servo will be disabled, and the amplifier will be temporarily disabled. If the voltage rises back above 10.8v, the amplifier will be re-enabled (if it had been enabled before the voltage drop), but you must send a command to the board if you want to re-enable the servo.

If the motor supply voltage rises above 54v, the amplifier will be temporarily disabled until the voltage drops back below 54v. The P.I.D. servo, however, will remain enabled. This is primarily to prevent a decelerating motor from acting as a generator and driving the voltage above a safe value.

If greater than 3 amps is required, the ***PIC-SERVO SC*** Motion Control board can be used with an external amplifier. External amplifiers may be for brush or *brushless* motors. PWM, Direction and Enable signals are provided on connector JP9. The ***PIC-SERVO SC***'s 20 KHz PWM output can be configured as either a PWM magnitude signal with a separate Direction signal, or as a single Antiphase PWM signal. In Antiphase output mode, a 50% duty cycle output will correspond to a *zero* drive output, a 100% duty cycle will correspond to a full *forward* output, and a 0% duty cycle will correspond to a full *reverse* output. This Antiphase output mode is more convenient for connecting to external amplifiers requiring an analog drive input.

The NMCTest program may be used for enabling the Antiphase mode, and also for permanently storing the Antiphase output mode option in EEPROM. On powerup, the Antiphase output mode will be automatically restored.

3.4 Limit Switch Inputs

The ***PIC-SERVO SC*** 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*** 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 & 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" 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 Physical Dimensions

The ***PIC-SERVO*** Motor Control board is 2.1" x 3.1" with four 0.156" dia. mounting holes at 1.8" x 2.45". The overall dimensions of the ***PIC-SERVO SC*** board and the placement of the connectors and amplifier chip are identical to earlier ***PIC-SERVO*** boards.

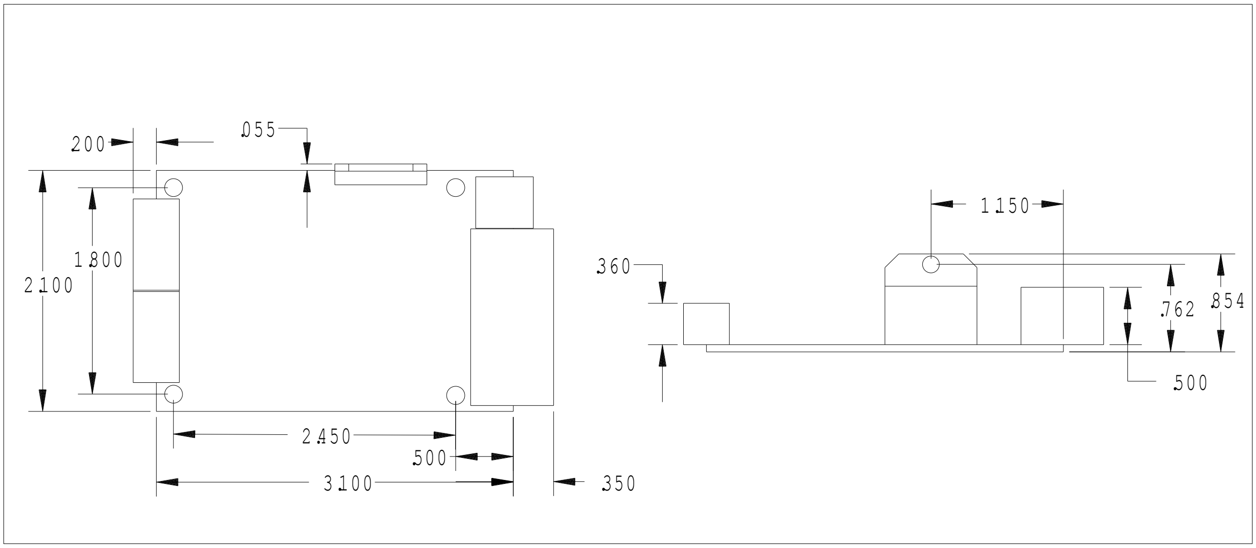


Figure 2 - ***PIC-SERVO SC*** board dimensions.

4. Contact Information

Additional information may be found from these sources:

JEFFREY KERR, LLC Web Site **www.jrkerr.com**
Your source for data sheets, application notes and test code. Technical support is provided via e-mail. Please see *jrkerr.com/contact.html* for contact information.

Microchip **www.microchip.com**
The ***PIC-SERVO SC*** is based on the Microchip PIC18F2331 microcontroller. Please refer to the Microchip data sheet for this devices for complete electrical, timing, dimensional and environmental specifications.

National Semiconductor **www.national.com**
Datasheet for the LMD18200 PWM amplifiers.

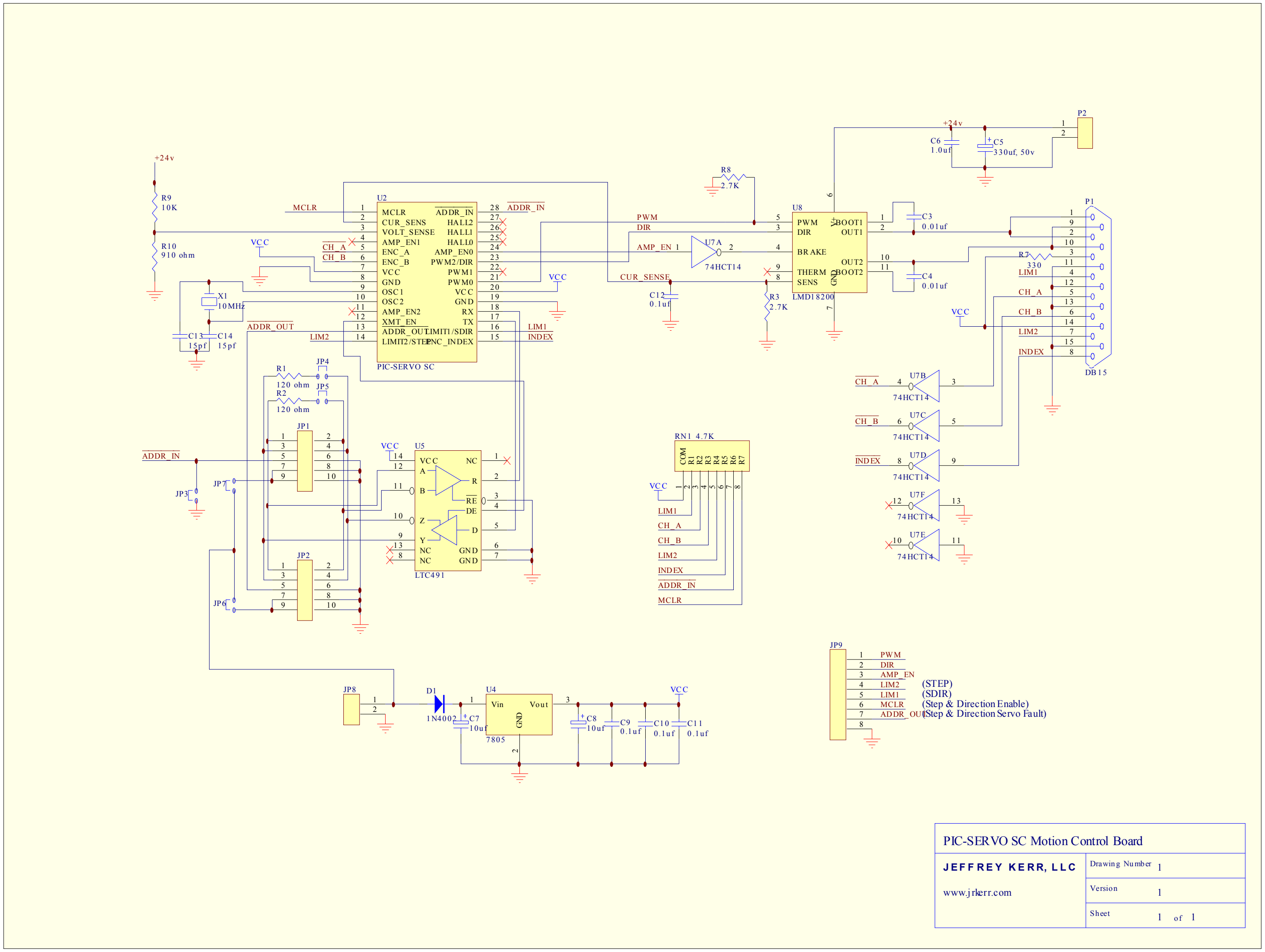


Figure 3 - ***PIC-SERVO SC*** Motion Control Board Schematic.