Serial Motor Interface (SMI) Circuit Kit - #3-511

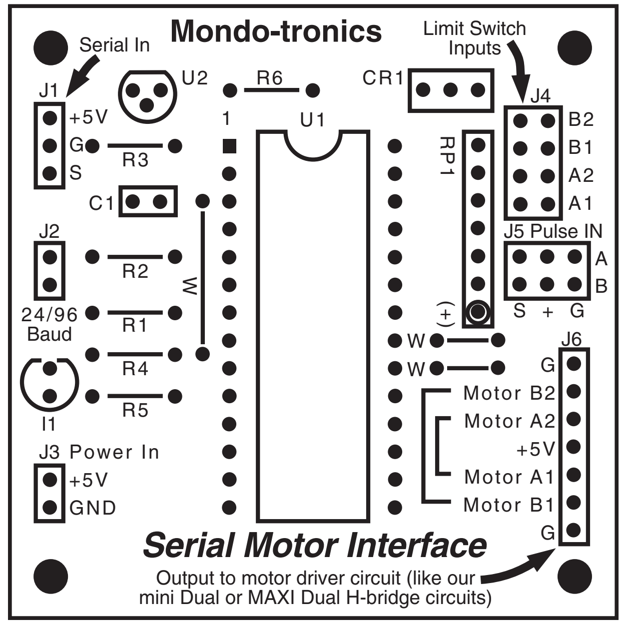
Section 1 - Introduction

The Serial Motor Interface Circuit Kit lets you control one of our MAXI or Mini Dual H-Bridge Motor Driver Kits. This gives you full serial based control of two DC motors from any PC, BASIC Stamp or other 2400 or 9600 baud serial source.

Run motors in forward and reverse with variable speed using Pulse Width Modulation. Separate encoder inputs and limit switch inputs permit "smart control" of each motor. Includes detailed instructions, schematic, application notes and sample control program in Basic.

Features standardized output connector - pin compatible with our Mini and MAXI Dual H-bridge kits. Plus, chain up to 127 SMI boards together on the same serial line to control up to 254 motors! Plus, its fully compatible with Scott Edward's Mini Serial Servo Controller 2 as well - letting you mix servos and DC motors on the same port!

Requires 5V DC power (just 10 mA, which can be borrowed from a Stamp's 5 volt supply), soldering, basic tools, plus a MAXI or Mini Dual H-bridge driver kit, serial cable parts, and a PC with serial port or other serial source.



Component side of the circuit board. (Note correct labeling of "A"s and "B"s along right side.)

Section 2 - Parts List

The kit includes the following:

Item	Quan	Location	Description
1.	3	R1, R3, R4	Resistor 100K Ω 1/4w 5% (brn blk yel gold)
2.	7	R2, RP1†	Resistor 22K Ω 1/4w 5% (red red org gold)
3.	1	R5	Resistor 470 Ω 1/4w 5% (yel vio brn gold)
4.	1	R6	Resistor 220 Ω 1/4w 5% (red red brn gold)
5.	1	C1	Capacitor ceramic mono 0.1μF 50v 20% radial Z5U
6.	1	CR1	Resonator 4MHz with caps
7.	1	I1	LED T-1 green
8.	1	U1	Socket 28-pin DIP
9.	1	U2	IC MN1381 Low Voltage Detect & Reset
10.	1	J1	Header, Male 0.100" 1 x 3
11.	2	J2, J3	Header, Male 0.100" 1 x 2
12.	1	J4	Header, Male 0.100" 2 x 4 (8 pins total)
13.	1	J5	Header, Male 0.100" 2 x 3 (6 pins total)
14.	1	J6	SIP Strip, female, 7 pos, 0.100"
15.	1	U1	IC. Microprocessor, PIC16C55-XT/P, Programmed
16.	1	J2	Shorting Block (Jumper) 0.1"
17.	1	-	PCB, Serial Motor Interface, Rev -
18.	1	-	Instructions, Serial Motor Interface Kit (these!)

Notes:

*The three locations marked "W" require short lengths of jumper wire such as the wire lead clipped from a resistor after installation. Insert wire lead from the component side of the board and solder in place.

† Resistor Pack RP1 is made from six individual resistors. See 4.08 below.

<u>Important Note About The Labeling of the PC Board:</u>

After producing the first run of the SMI circuit board, we realized that there was a better way to identify and label the motors and their associated inputs and outputs. The illustration above and the references through out this document refer to this improved labeling, but differ from the board included with this kit.

Please mark corrections to your PC board with a fine point permanent marker so that they match the drawing above (ten changes total - start at the top right by changing 2B to B2, and end at the bottom right by changing 2A to B1). Sorry about that!

Section 3 - Other Parts, Tools and Materials List

In addition, assembly and operation will require:

1. Soldering iron for electronics
2. Solder for electronics
3. Moist sponge to clean soldering iron tip
4. Wire stripper
5. Side cutter
6. PC computer with serial port and QBASIC application software
7. Connectors and wire for interface cable to PC serial port
8. MAXI or Mini Dual H-bridge driver kit
9. DC motor(s), motor power supply and wiring for motors.
10. Parts for pulse input sensors and limit switches, depending on your configuration. (See Section 6 below for more details.)

Section 4 - Build It

If you have not soldered or assembled electronics before, please consult a book on basic electronic assembly or get the assistance of an experienced board assembler.

Generally, assemble the Printed Circuit Board starting with components having the lowest height above the board, and work upwards in height.

For all parts note the part (watch the resistor color codes), the location of the part (don't put it in the wrong place), and its orientation. Many parts, like light emitting diodes and integrated circuits, can be damaged or destroyed if powered "the wrong way".

Suggested order of assembly:

4.01) Six resistors, R1 to R6 (note values & locations, orientation matters only for appearance).

4.02) Three jumper wires at locations marked "W" (use cut off leads from resistors above).

4.03) One socket at location U1 (note orientation of indent printed on board)
NOTE: Use care when soldering the socket pins as some of the traces on the PC board are not fully covered by the green solder mask and can get short circuited. Future boards will fix this.

4.04) One capacitor at location C1 (orientation does not matter).

4.05) One ceramic resonator at locations CR1 (insert in either direction).

4.06) One green LED at locations I1 (direction critical - note orientation of flat side).

4.07) One Low Voltage Detector at location U2 (direction critical - note orientation of flat side).

4.08) Build the six-resistor array at location RP1 as follows:
A) Insert six 22K Ω resistors (use correct ones - color coded red red orange gold and put them all in the same way for good appearance). Bend one so it also goes into the hole marked "(+)", as shown below. Solder in place on the board.
B) Bend the extended lead from first resistor to cross all others and contact the other bent lead.
C) Solder all top leads together and trim.

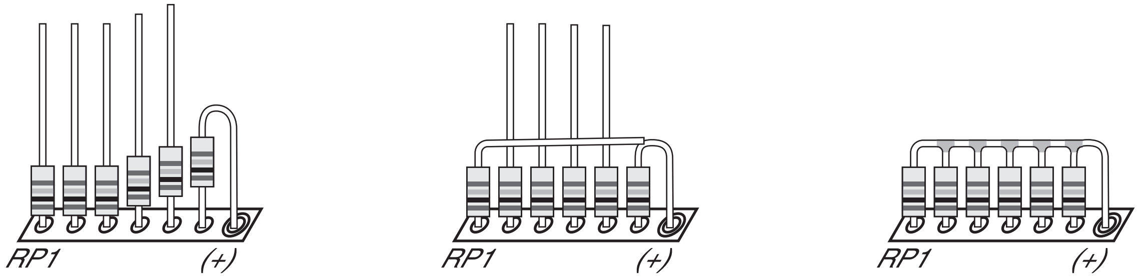


FIG. 4.08 - A. Insert & solder to board B. Bend lead across C. Solder all together & trim

4.09) Five male headers (two 2 pin, one 3 pin, one 6 pin, one 8 pin) and one 7 pin female header in locations J1 through J6 (note locations and insert short ends into board and solder).

4.10) One PIC16C55 integrated circuit, press into socket IC1 (note orientation - indent end matches socket)

4.11) One shorting block, press on to both pins of header J2 to select 9600 baud, or press on only 1 pin (or leave off) of J2 to select 2400 baud.

This completes the assembly of your Serial Motor Interface circuit. Double check all connections and solder points. Make sure all parts are in the correct locations and orientations, that the solder joints are bright and clean, that leads are trimmed, that there are no stray solder bridges or other connections to interfere with the circuit's operation.

Section 5 - Hook It Up

In connecting power and serial data to the Serial Motor Interface, you have two options:

5.1) Power & Serial Option A - Single Source (such as a BASIC Stamp)

Build a cable for connecting a serial port and power from a circuit (such as from a BASIC Stamp) as shown. using a 3 pin, 0.100" spacing connector at J1, and connectors as needed by the serial and power supply source.

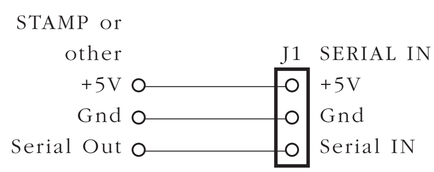


FIG 5.10 - Power and serial data arrive on the same cable from a Stamp 2 (or other computer such as Next Step, HC11, etc.) and provide both power and data to the Serial Motor Interface.

Note: if you are using a Mini Dual H-Bridge Driver Kit, we recommend using a separate (or a well isolated) 5V supply for the motors, in order to avoid electrical noise problems from the motors.

5.2) Power & Serial Option B - Separate Source (as with a PC and a 5V Power Supply)

Alternately, you may connect only serial data and ground on J1 from a PC, Mac or other serial source, and provide a separate 5V DC source to connector J3. Note that the computer, the 5V supply, and the SMI board all share a common Ground connection via the SMI board.

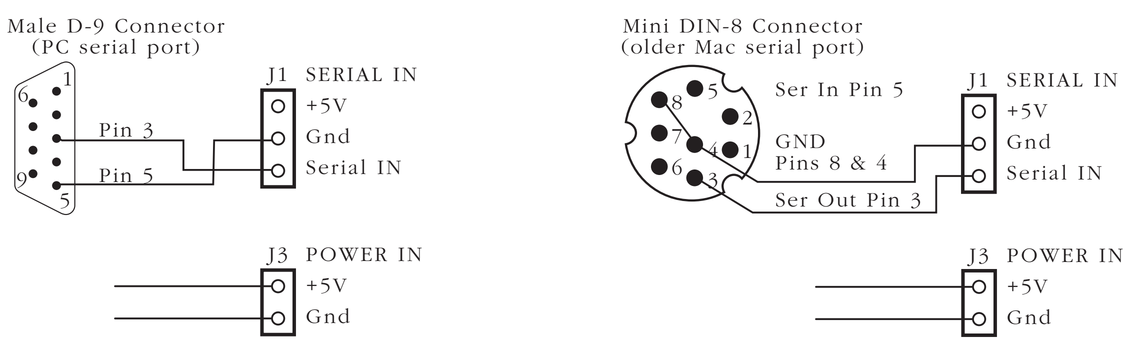


FIG 5.2 - Connect serial data from a PC serial port or an older Macintosh serial port. For newer iMacs with USB ports, use the PC cable above left, and a USB-Serial converter like a Keyspan USA-19 (www.Keyspan.com).

WARNING: With either power option above, be sure that the Power Supply wires are correctly connected. Even a momentary reversal of polarity (switching +5 and GND) can cause permanent damage.

5.3) Select the Baud Rate

You may send serial data to the SMI via TTL, RS-232 or other similar signal levels. You may select either baud rates of 2400 or 9600 baud.

To select 2400 baud, leave off jumper at J2. To select 9600 baud, install jumper at J2.

Note: The SMI looks at the status of J2 the Baud Select jumper only at startup. If you change the jumper, be sure to turn off the circuit and restart to enable the new baud rate setting.

5.4) Motor Driver Option One - MAXI Dual H-Bridge Driver, Motors and Power

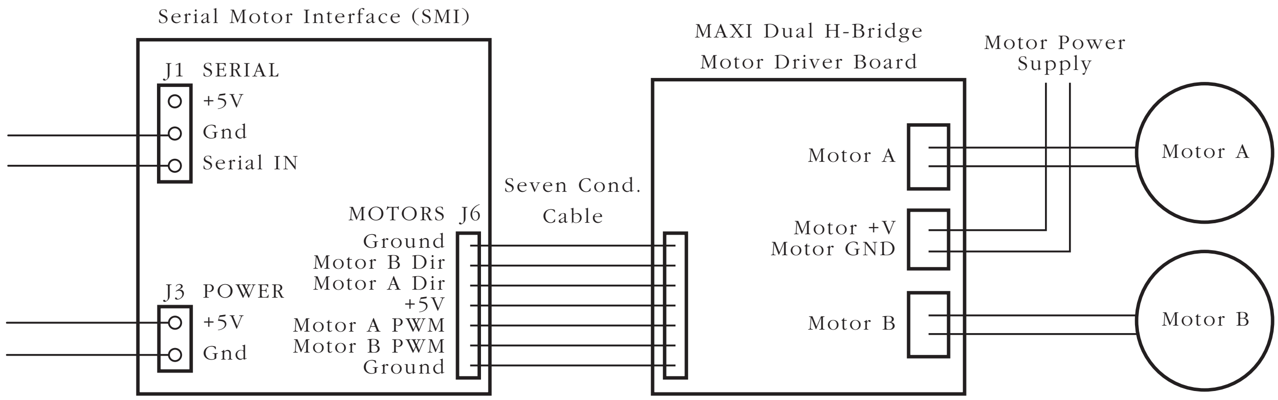


FIG 5.4 - Connecting MAXI Dual H-Bridge Driver Kit, DC Motors and Power

The MAXI Dual H-Bridge Motor Interface Kit (#3-773) provides control of two large DC motors, from 10 to 40 volts, and up to 10 Amps continuous each.

Be sure to use adequate heat sinking on the power transistors, per the kits instructions. The MAXI Dual circuit works with the Active Braking feature of the SMI for quick stops (see section 5.7 below).

5.5) Motor Driver Option Two - Mini Dual H-Bridge Driver, Motors and Power

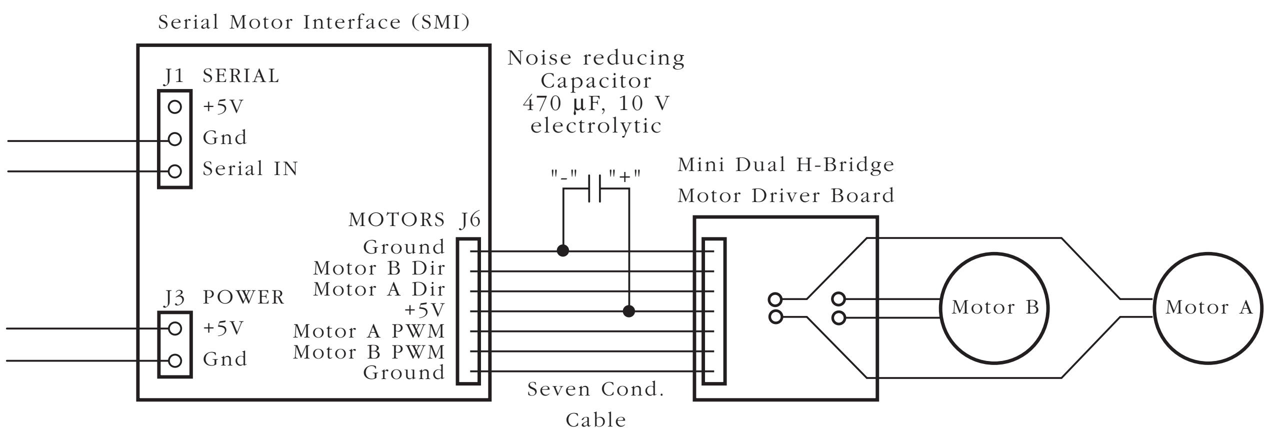


FIG 5.5 - Connecting Mini Dual H-Bridge Driver Kit, DC Motors and Power

The Mini Dual H-Bridge Motor Interface Kit (#3-301) provides control of two small DC motors when powered by 5V. For protection from electrical interference, be sure to include the 470µF capacitor as shown (either in the cable, or mounted directly to the back of the Mini Dual circuit board.

For even better electrical isolation, provide the Mini Dual with a separate 5V supply (but still have common Ground lines, and include a capacitor to help reduce electrical noise in the circuit.

5.6) Load and run the demonstration software

Boot up the PC, run QBASIC and load the program SMIdemo1.BAS (listed below). For more on QBASIC, see the Notes section below. You may carefully type the program in, or download it from our web site.

Turn on the power to the motors. Run the software. The demo program permits control of both motors, including their speed and direction. Experiment with the Serial Motor Interface and the driver circuits, and use the software listing as a starting point to develop your own programs.

Each change to the motors operation is performed by a single command sequence. The command sequence is compatible with serial data commands used with Scott Edwards' popular controller for servo motors, the Mini SSC2. The command sequence for the Mini SSC2 is easy - send the SMI a sequence of three numbers (using an LPRINT command from Basic, or similar command in C, Linux or the language of your choice) in the form:

255, M, C

where 255 is the "sync byte", M is the motor number (defaults are 0 for Motor A, or 1 for Motor B, and higher motor numbers, up to 254 are possible - see section 6.4 Setting Higher Base Addresses below), and C is the command 0 to 254 for each motor, as follows:

5.7) Operating Mode One - Direction, Speed and Limit Switches

Mode One permits the operation of both motors at any speed across their full range. Plus, limit switches can be added to signal the extreme limits of action of a motor, and the SMI will automatically shut off the motor, and await the next command. Each motor can have two limit switches (connected at Connector J4) - one to signal to limit of motion in each direction.

Command	Motor Action
0	Full Stop (free running)
1-31	Reverse with PWM speed control (1 slowest, 31 fastest), with limit switch auto-stop inputs (Motor A stops if Limit Switch A2 closes, Motor B stops if Limit Switch B2 closes).
32	Full Stop (active braking)
33-63	Forward with PWM speed control (33 slowest, 63 fastest), with limit switch auto-stop inputs (Motor A stops if Limit Switch A1 closes, Motor B stops if Limit Switch B1 closes).

Command 0 and command 32 both stop the motor. However, command 0 stops in a "free running" mode (both outputs for a given Motor at J6 go LOW), permitting motors connected to the MAXI Dual H Bridge to continue turning until they stop on their own.

Conversely, command 32 makes both outputs for a given Motor at J6 go HIGH, causing a motor connected to the MAXI Dual to stop rapidly and remain held in place. (The Mini Dual H always stops in free running mode).

The demo program SMIdemo1.BAS (listed below) uses Mode One, which permits control of speed and direction for two motors.

In addition, when running in Mode One, you may add limit switches that will automatically stop the motor when triggered. This is useful for permitting the automatic protection of your device, without the direct involvement of the controlling computer.

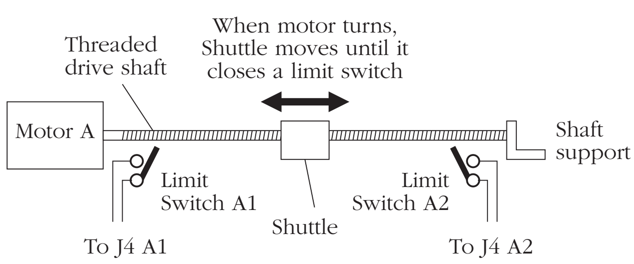


FIG 5.7 - Example of using limit switches in Mode One for Motor A. Create a similar setup for Motor B with two additional switches at inputs B1 and B2.

However, limit switches are optional, and the SMI will function in Mode One without them, simply running the motor in the speed and direction as commanded.

5.8) Operating Mode Two - Direction and Pulse Counts

Mode Two permits the operation of both motors (the automatically ramp up to full speed in about 1 second). This mode requires an external pulse generating encoder. The pulse sensor (a contact switch, an IR sensor, a magnetic sensor or many other devices) attaches to the J5 Pulse IN A for Motor A, or J5 Pulse IN B for Motor B. See figures 5.8a and 5.8b below.

Upon receiving a command 64 through 127, the SMI will automatically ramps the motor up to full speed and begins counting input pulse transitions ("transitions" (up or down) not complete pulses), from 1 up to 31, depending on the command (see listing below). Each motor has just one encoder, connected at Connector J5.

Command	Motor Action
64	Full Stop (free running)
65-95	Reverse, ramp up to full speed, count (Command - 64) transitions on J5 Pulse IN, then stop.
(1 - 31	Note: (Command - 64) equals the total number of transitions (Hi-Lo or Lo-Hi) to count.
pulses)	(Example: to count 10 transitions on input Pulse IN, set Command = (64+10) or 74.)
96	Full Stop (active braking)
97-127	Forward, ramp up to full speed, count (Command - 96) transitions on J5 Pulse IN, then stop.
(1 - 31	Note: (Command - 96) equals the total number of transitions (Hi-Lo or Lo-Hi) to count.
pulses)	(Example: to count 10 transitions on input Pulse IN, set Command = (64+10) or 74.)

Mode Two, if the circuit does not have some form of input pulse signal from a sensor, the SMI will drive the motor in the commanded direction at full speed until receiving a command to stop (command 64 or 96). Limit Switches work only with Mode One, and have no effect with Mode Two commands.

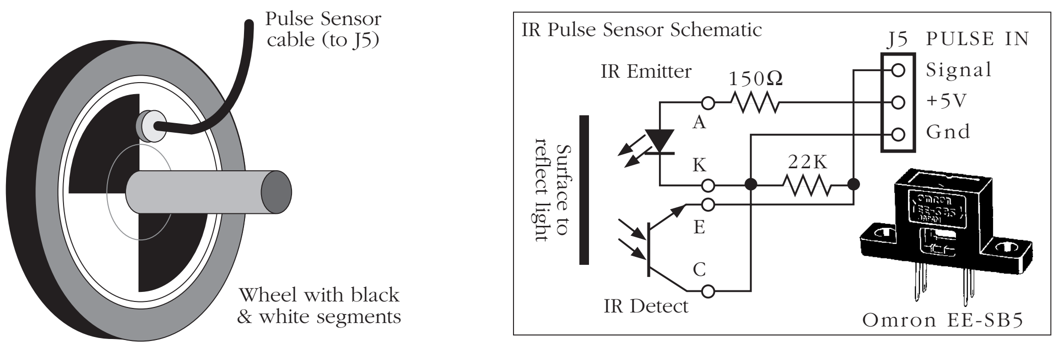


FIG 5.8a - Example of reflective IR optical pulse encoder (Omron EE-SB5, DigiKey #OR500),

Note that the SMI counts pulse "transitions" (up or down) not complete pulses, permitting a higher resolution count from a given pattern. For example, in figure 5.8a above, one full rotation of the black and white wheel would require a count of 5 (hi, lo, hi, lo, hi). Four transitions could be as little as 3/4 of a revolution, whereas 5 transitions could be as 1 and 1/4 revolution. For greater accuracy, use a greater number of black and white segments.

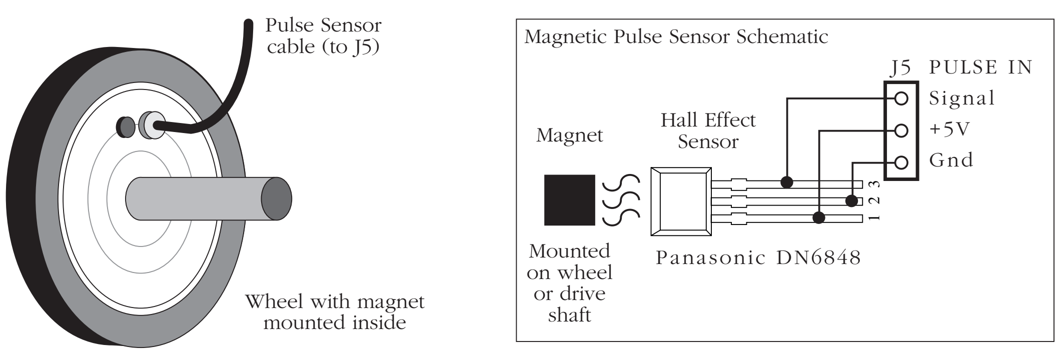


FIG 5.8b - Example of magnetic Hall Effect pulse encoder (Panasonic DN6848, DigiKey #DN6848).

Again, as the SMI counts pulse "transitions" (up or down) not complete pulses, in figure 5.8b above, one full

Again, as the SMI counts pulse "transitions" (up or down) not complete pulses, in figure 5.8b above, one full rotation of the magnet could register in as few as two transitions (hi, lo) given the magnet starting directly under the sensor. For greater accuracy, use multiple magnets, or place the magnet and sensor further "up" on the gear train to detect at greater resolution.

5.9) Higher modes

The Command numbers higher than 127 are not currently used. Let us know what features you most want to see in future versions of the SMI. See our email address at the end of this document.

Command	Motor Action
128-191	Repeats as 64 to 127 above (reserved for future use)
192-254	Repeats as 64 to 127 above (reserved for future use)
255	Do not use (same as sync byte).

Notes:

Do not switch modes during operation. Once powered up, an SMI should be commanded in only one Mode to avoid conflicting actions. Switching modes may result in irregular performance.

Do not mix modes (ie run Motor A in one mode and Motor B in the other mode. This may also result in erratic and unpredictable performance.

Pulse Inputs have no effect on Commands 0 to 63. Limit Switches have no effect on commands 64 and higher.

Section 6 - Technical Details & Options

6.1) If it doesn't work...

a) Double check the location and orientation of all components

b) Double check all solder joints for
- wires in wrong locations
- bridged or short-circuited solder joints
- weak solder joints

c) Double check the cable wiring with a continuity tester and make sure it matches the schematic.

d) Sufficient power - be sure your power supply has the correct voltage and sufficient current

e) Serial port settings. Correct port? Enabled? Other applications interfering?

f) Noise in circuit. If you have a power supply common to both the circuit and motors, be sure to add capacitors to block any interference. Better yet, use separate supplies for motors and logic.

6.2) LED - what does it blink about?

The LED on the SMI board blinks when it is happy. What makes the SMI happy? These:

- Starting up, the program shows that it is alive by turning the LED on.

- Receiving data will cause the LED to toggle states (ie, if the LED is off and the SMI receives a command, it will turn the LED on. The next command will turn it back off).

- Input pulses make the LED toggle states - once for each of the pulse transition detected. This permits you to visually count the pulses as they come in, and easily verify the functioning of the pulse sensor.

6.3) Schematic

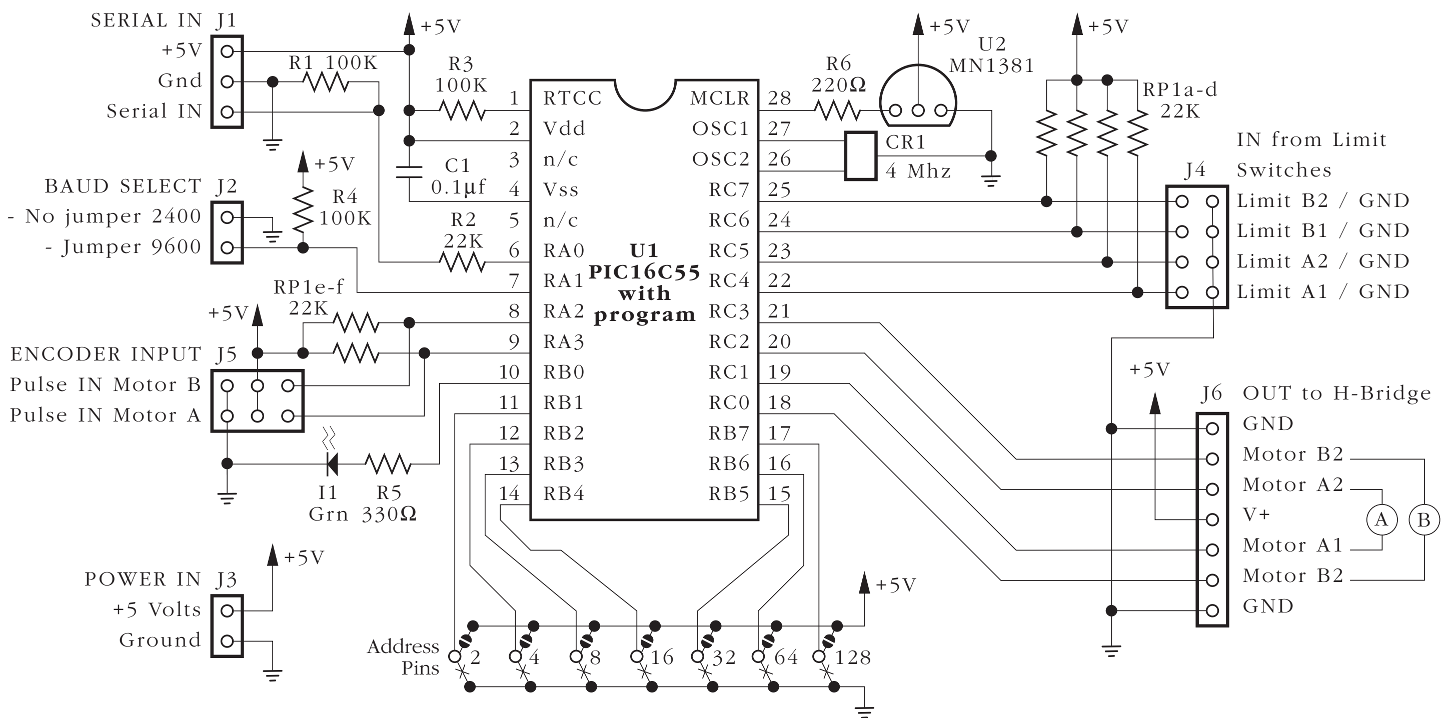


FIG 6.3 - Here are the connections between the parts of the circuit. For connections to external devices and supporting components, see the additional figures above.

6.4) Setting Higher Base Addresses for use with multiple Controllers

Multiple SMI boards can be addressed via a single serial data line. The serial line may also include Scott Edward's popular Mini Serial Servo Controller 2 (Mini SSC2). The SMI and Mini SSC2 use the same data protocol (255, M, C). The address (M) can be any number from 0 to 253 (set by cutting and jumpering special pads and traces on the back of the SMI). The Mini SSC2 come with preset addresses.

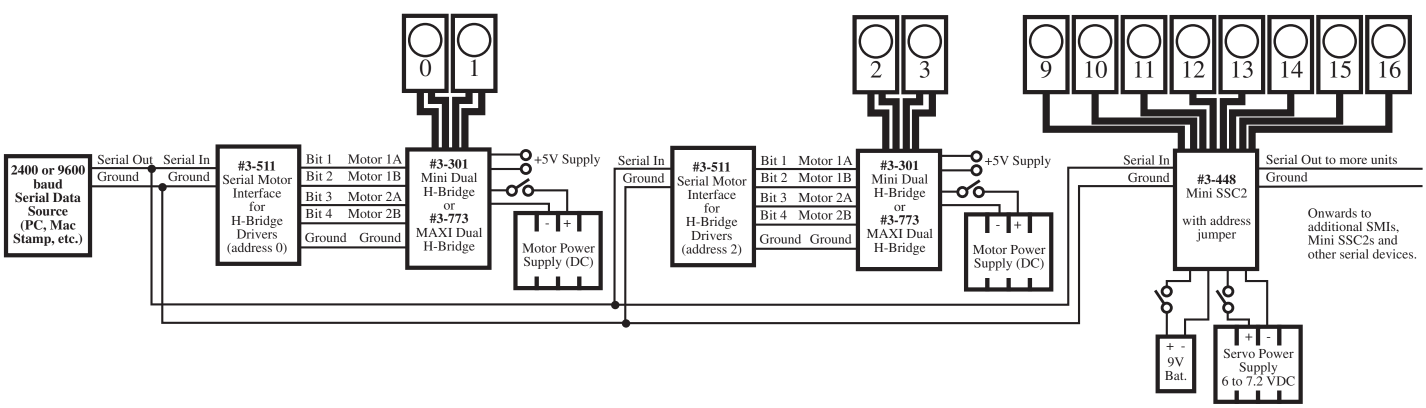


FIG 6.4 - Example of intermixing SMIs and Mini SSC2s to permit the control of DC and servo type motors over the same single pair serial data line.

CHANGING THE SMI BASE ADDRESS

The default Base Address for the SMI board is zero (ie all seven Address lines of the microprocessor are tied to ground, logical low). You need change it only if it will interfere with other devices on the same serial bus.

To set to an alternate Base Address (from 2 to 254): FIRST disconnect desired Address Pin(s) from ground (ie cut correct traces on back of circuit board at "**X**" marks), then bridge corresponding ◖◗ pad together with solder. This jumpers the Address Pin to +5V, logical high. To determine the new Base Address, add the address value (ie. 2, 4, 8, etc.) of all soldered ◖◗ pads .

Example 1: To set Base Address to 28 (ie Motor A to address 28 and Motor B to 29), cut traces at "**X**" marks and solder corresponding ◖◗ pads at the following: 16, 8 and 4 (as 16 + 8 + 4 = 28).

Example 2: To set motors to Addresses 100 and 101, cut and solder 64, 32 and 4 (as 64 + 32 + 4 = 100).

*CAUTION: Be sure to completely cut "**X**" traces for any soldered jumper, otherwise you risk shorting out your 5 Volt power supply and damaging your project!*

6.5) Alternate Addressing Option.

Should your SMI require frequent changing of its Base Address, you may wish to attach your own switches to the board, and eliminate the need for modifying **X**s and ◖◗s . Use seven small single pole double throw (SPDT) switches (such as Jameco #109170), or a single 8 Position Tri State DIP Switch (such as Jameco #ETA108E).

Cut all **X**s on the board, and unsolder any bridged ◖◗s. Solder a wire from each address pin on the IC to the center tap of each switch. Attach a ground line to one side of all switches, and a +5V line to the other side, per the schematic in Figure 6.5.

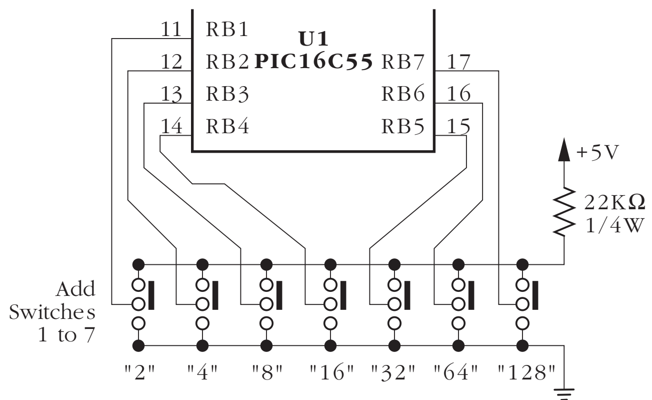


FIG 6.5 - Installing switches for frequent changing of the SMI's Base Address. Wire them to a separate piece of perf board and mount next to the SMI.

Note: the SMI looks at the conditions of the address inputs only at startup. If you change the address, be sure to power down and restart the SMI to enable the new address.

Section 7 - Notes

7.1) Running QBASIC on Your PC

In Windows 95, 98 or NT, look under the Start Menu, choose MS DOS Command Prompt. Then, from the DOS window, type QBASIC.

If you don't have QBASIC installed, follow these steps:

1. Start Windows
2. Put the Windows CD-ROM into your CD-ROM drive.
3. Double-click on My Computer on the Desktop.
4. Right-click on your CD-ROM drive and click on Open.
5. Double-click on Other and then on Oldmsdos.
6. Click on Qbasic.
7. While holding down Shift, use the right cursor (arrow) key to select Qbasic.exe and Qbasic.hlp.
8. Click on Edit and then on Copy.
9. Close all My Computer windows.
10. Double-click on My Computer on the Desktop again.
11. This time, double-click on your hard drive icon.
12. Click on File, drag the cursor to New and then click on Folder.
13. A folder with the name New Folder appears. Type in QBASIC and then press Enter.
14. Double-click on Qbasic, the folder that you just created.

15. Click on Edit and then Paste. Two new items, Qbasic.exe and Qbasic.hlp were created under that folder.
16. Close all My Computer windows.
17. To create an icon for QBASIC, click on the Start button, and then drag the mouse cursor to Settings, next click on Taskbar...
18. Click on the Start Menu Programs tab and then Add...
19. For the Command line:, type in: C:\QBASIC\QBASIC.EXE, click on Next.
20. Click on New Folder..., type in QBASIC, and press Enter.
21. Click on Finish and then click on OK in the Taskbar Properties window.
22. To access QBASIC, click on Start, move the cursor over Programs, then move it over QBASIC, finally, click on Microsoft Quick BASIC.

For more on QBASIC visit:

http://www.geocities.com/SiliconValley/Park/4504/index.html

Section 8 - Program

The sample program below operates the Serial Motor Interface and an attached Mini or MAXI Dual H-Bridge circuit via the serial port of a PC. Though simple, it provides a good start and permits full, independent control of both motors. You can adapt the circuit and this program to run with BASIC Stamps, or use it to develop your own programs in other languages. These instructions are available as a .PDF file via our website: http://www.RobotStore.com. Get it and avoid typing in this program!

```
'SMIdemo1.BAS
'Serial Motor Interface demo program RG 0102.14
'Control direction and speed of two DC motors via serial port.

'Setup
OPEN "com1:9600,N,8,1,CD0,CS0,DS0,OP0" FOR OUTPUT AS #1
WIDTH #1, 255

'Base Address on board defaults to 0 for Motor A and 1 for Motor B
'Higher Base Address set by modifying jumpers on back of board (see instructions).
'Higher Base Addresses can be 2 through 252
'NOTE: Do not set SMI Base Address to 254

MotorA = 0               '+2+4+8+16+32+64+128 'Set Base Address Motor A
MotorB = MotorA + 1      'Motor B Address is always Base Address + 1

Restart:                 'Clear all speed and direction values
SpeedA = 0
SpeedB = 0
DirA = 0
DirB = 0
ModeA$ = "Coast"         'Set brake modes to Coast
ModeB$ = "Coast"
GOTO Display

BigLoop:                 'Main control loop
A$ = INKEY$              'check keyboard

'UPPER CASE COMMANDS - Motor A Control
'Speed is from 1 (slow) to 31 (full on)
IF A$ = "J" THEN SpeedA$ = "Slower ": SpeedA = SpeedA - 1: IF SpeedA = (-1) THEN SpeedA = 0: GOTO UpdateA
ELSE GOTO UpdateA
IF A$ = "L" THEN SpeedA$ = "Faster ": SpeedA = SpeedA + 1: IF SpeedA > 31 THEN SpeedA = 31: GOTO UpdateA
ELSE GOTO UpdateA

'Direction is 0 for CW, 32 for CCW
IF A$ = "I" THEN ModeA$ = "CW   ": DirA = 0: GOTO UpdateA
IF A$ = "M" THEN ModeA$ = "CCW  ": DirA = 32: GOTO UpdateA

'Brake modes are Coast (Command = 0), Brake (Command = 32)
IF A$ = "K" THEN ModeA$ = "Coast": SpeedA = 0: DirA = 0: GOTO UpdateA      'Stop Coast mode
IF A$ = "B" THEN ModeA$ = "Brake": SpeedA = 0: DirA = 32: GOTO UpdateA     'Stop Brake mode

'lower case commands - Motor B Control
'Speed is from 1 (slow) to 31 (full on)
IF A$ = "j" THEN SpeedB$ = "Slower ": SpeedB = SpeedB - 1: IF SpeedB = (-1) THEN SpeedB = 0: GOTO UpdateB
ELSE GOTO UpdateB
IF A$ = "l" THEN SpeedB$ = "Faster ": SpeedB = SpeedB + 1: IF SpeedB > 31 THEN SpeedB = 31: GOTO UpdateB
ELSE GOTO UpdateB

'Direction is 0 for CW, 32 for CCW
IF A$ = "i" THEN ModeB$ = "CW   ": DirB = 0: GOTO UpdateB
IF A$ = "m" THEN ModeB$ = "CCW  ": DirB = 32: GOTO UpdateB
```

```
'Brake modes are Coast (Command = 0), Brake (Command = 32)
IF A$ = "k" THEN ModeB$ = "Coast": SpeedB = 0: DirB = 0: GOTO UpdateB      'Stop Coast mode
IF A$ = "b" THEN ModeB$ = "Brake": SpeedB = 0: DirB = 32: GOTO UpdateB     'Stop Brake mode

'Reset or Quit
IF A$ = "R" OR A$ = "r" THEN GOTO Restart        'Reset values
IF A$ = "Q" OR A$ = "q" THEN PRINT "Done": PRINT #1, CHR$(255); CHR$(MotorA); CHR$(0); : PRINT #1,
CHR$(255); CHR$(MotorB); CHR$(0); : END

GOTO BigLoop


UpdateA:         'Send MotorA command to SMI board
PRINT 255, MotorA, SpeedA + DirA
PRINT #1, CHR$(255); CHR$(MotorA); CHR$(SpeedA + DirA);
GOTO Display


UpdateB:         'Send MotorB command to SMI board
PRINT 255, MotorB, SpeedB + DirB
PRINT #1, CHR$(255); CHR$(MotorB); CHR$(SpeedB + DirB);
GOTO Display


Display:
'GOTO BigLoop
CLS
PRINT "Mondo-tronics' Serial Motor Interface Demo - SMIdemo1.BAS"
PRINT ""
PRINT "      CAPS - Motor A   lowercase - Motor B"
PRINT ""
PRINT "                    Clockwise "
PRINT "                       I"
PRINT "             Slower  J  +  L  Faster"
PRINT "                       M"
PRINT "                  Counterclockwise"
PRINT ""
PRINT "     K/k for Stop   R for Reset   Q to Quit"
PRINT ""
PRINT "   MODE A: "; ModeA$; "     SPEED A: "; SpeedA$, SpeedA
PRINT "   MODE B: "; ModeB$; "     SPEED B: "; SpeedB$, SpeedB
GOTO BigLoop
END
```

Contacting Us:

Mondo-tronics. Inc.
4286 Redwood Hwy PMB-N
San Rafael, CA 94903

Phone 415-491-4600
Fax 415-491-4696
Support 415-491-4631

Email support@RobotStore.com
Web http://www.RobotStore.com

Serial Motor Interface designed by Zach Radding (www.BuildCoolStuff.com) for Mondo-tronics.

Comments? Errors? Improvements? Compliments?

Help us make this product better with your feedback. We want to hear from you!

Email us at: support@RobotStore.com

Thanks!