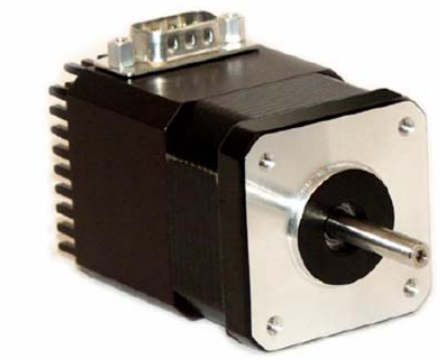
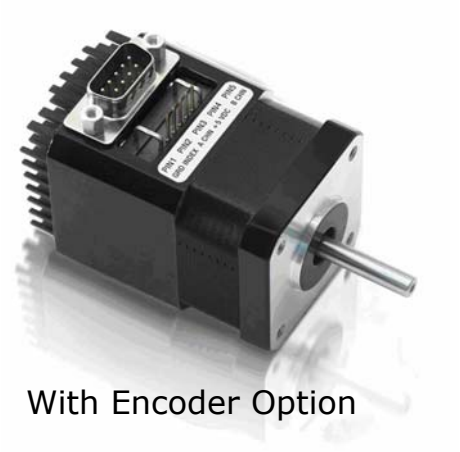
IMC17 / IMCE17

INTEGRATED STEP MOTOR, DRIVER, AND CONTROLLER





With Encoder Option

USER MANUAL

Version 1.14

RMS Technologies
2533 N. Carson St. #4698, Carson City, NV 89706-0147
1-877-301-3609
www.rmsmotion.com sales@rmsmotion.com

Thank you for purchasing the IMC17 - Integrated Motor Driver Controller. This product is warranted to be free of manufacturing defects for one year from the date of purchase.

PLEASE READ BEFORE USING

Before you start, ensure that there is a suitable DC power supply. A current limited lab supply is recommended for first time users to guard against the possibility of miswiring. In addition, in order to prevent any harm to the controller board, **do not disconnect the unit while power is still being supplied**. Do not exceed 40VDC under any circumstances.

DISCLAIMER

The information provided in this document is believed to be reliable. However, no responsibility is assumed for any possible inaccuracies or omissions. Specifications are subject to change without notice.

RMS Technologies reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. RMS Technologies does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

There are known issues involving the Halt command (i.e., H01) when stored in memory location zero. Upon power up, the remaining command string after the Halt command might be executed if the user types in a new command. If memory location zero is not being used, the user is advised to always clear everything in memory by typing /1?9. Otherwise, the user may terminate the remaining command string in the buffer by issuing a /1T.

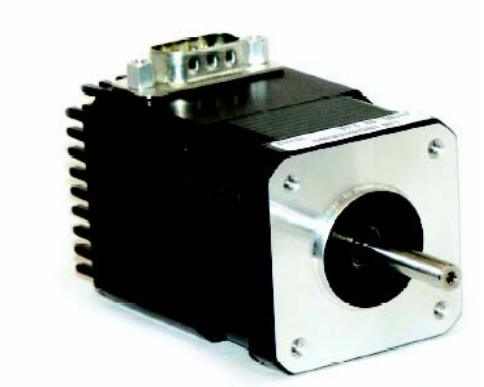
TABLE OF CONTENTS

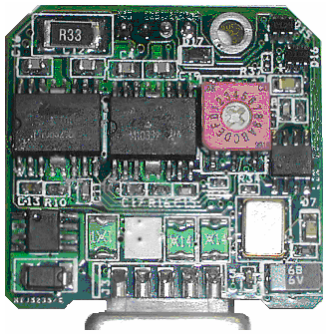
1. INTRODUCTION

RMS Technologies sells integrated solutions for your motion control needs; including motor and driver, motor and driver with an encoder, motor controller/drivers, and motor controller/drivers w/ encoder. The IMC comes in two sizes: NEMA 17 and 23.

This manual will cover the IMC17, Integrated NEMA 17 Step Motor and Controller/Driver solution.

The IMC17 is designed to allow for rapid implementation of stepper motors in products requiring automation.





With a fully intelligent controller integrated with the step motor, the IMC17 accepts high level commands from an RS232/RS485 link. The controller only adds 0.5" to the length of the motor which otherwise is a standard NEMA Size 17 High Torque Step Motor.

To help introduce users to our IMC line of Integrated Solutions, RMS Technology offers a Designer's Kit, which enables first-time users to easily implement their own programs. The kit includes an RS485 to RS232 converter card, a CD-ROM with software and manuals, along with a DB-9 connector cable and accessory pieces. The converter card allows you to connect the IMC17 to your PC via a standard Serial Port. Commands can be issued from any serial terminal program such as HyperTerminal (Windows 98/NT/2000/XP versions are compatible). The command set for the IMC17 is easy to learn and very intuitive. For example the command /1A10000R will move the step motor at Address 1 to Absolute Position 10000 (steps). This communications protocol is compatible with devices that use the Cavro DT or OEM protocol.

The IMC17 is also capable of stand alone operation with no connection to a PC. It can be set to execute a preset string of commands upon power up. Commands include nested loops, execution halt pending a switch closure, and much more. It is possible to daisy chain up to 16 different units at once.

This user manual is a complete guide to setting up the IMC17 unit. It also contains information on various types of inputs and outputs that this unit can use. In addition, there is a separate document with the full Command List for programming the IMC17. The most up to date versions of both these documents is available online at www.rmsmotion.com → Downloads → IMC17

2. FEATURES

- Single 4 wire bus linking up to 16 stepper motors
- 1.50 Amp Chopper (PWM) Driver
- Operates from +12V to 40V
- Stand alone operation with no connection to a PC
- Execution Halt pending switch push button
- Pre-wired for Opto Switch inputs
- 1/2, 1/4, 1/8, 1/16, 1/32, 1/64 step resolution
- Homes to an Opto or Switch closure with a single command
- Fully programmable ramps and speeds
- Two digital I/O and two fixed input channels
- Switch selectable address
- Software selectable "Move" and "Hold" currents
- Hold Current automatically selected upon move completion
- Simple DB9 connection

DESIGNER'S KIT

If you have purchased the optional Designer's Kit, here is the list of the additional components:

- IMC17 - Integrated Motor and Controller/Driver
- RS485 to RS232 Converter Card
- A DB-9 female connector cable, a switch push button, Opto Sensor, and extra wiring for I/O
- User Manuals and Software on CD

DEFAULT SETTINGS

Running Current	30%
Holding Current	10%
Step Resolution	Half Step
Start Velocity	200 pps
Top Running Velocity	3700 pps
Position	0

Table 1: Default Settings

3. ELECTRICAL SPECIFICATIONS

Power Supply Requirements

Voltage	+12 VDC to +40 VDC

Driver

Peak Current:	0.1 to 1.5 Amps Peak

Motor Specifications

NEMA Size 17

Motor Rated Current:

IMC17-S01	0.9 Amps RMS
IMC17-M01	1.4 Amps RMS
IMC17-L01	1.05 Amps RMS

Holding Torque:

IMC17-S01	43.0 oz-in
IMC17-M01	61.4 oz-in
IMC17-L01	86.3 oz-in

Steps per Revolution (1.8° Motor)

400, 800, 1600, 3200, 6400, 12800

Digital I/O Specifications

Number of I/O	2
Number of Inputs	2
Input Voltage	+0 VDC to +24 VDC
Input Current	700 mA
Pull-up Resistors	10k Ω
Protection	Static Protection to the microprocessor

4. OPERATING SPECIFICATIONS

Maximum Step Frequency	10,000 pulses per second (pps)
Operating Temperature Range	0° to 50° C
Storage Temperature Range	-20° to 70° C

5. COMMUNICATION SPECIFICATIONS

Interface Type	RS485 to RS232 converter card
Baud Rate	9600 bps
# Bits per character	8 Data
Parity	None
Stop Bit	1
Flow Control	None

6. MECHANICAL SPECIFICATIONS

A. Motor Front Shaft Extension Length
Standard length is 0.94". Customized length is available.

B. Motor Shaft Diameter
Standard shaft diameter is 0.1968".

C. Overall Body Length
Motor body length is available in various lengths
IMC17-S (2.69")
IMC17-M (2.92")
IMC17-L (3.24")

Dimensions

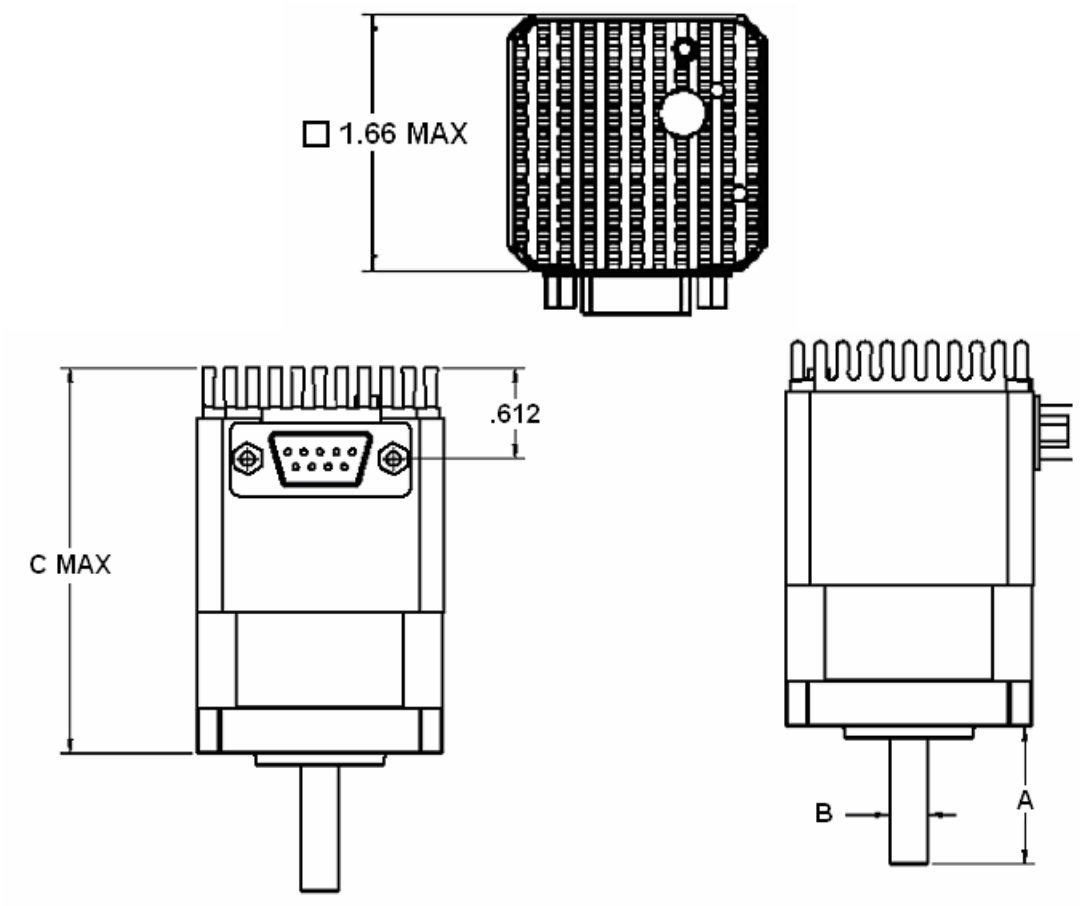


Figure 1: Dimensions Diagram

7. CONNECTORS

A DB-9 female connector cable receives power and provides the control connections for the IMC17 Unit. On the opposite end of the DB-9 female connector cable, there is a 4 pin connector provided for the converter card in order for the driver to communicate with the PC. This allows the user to solder and program the switch push button and the Opto Sensor, enabling several options. The two I/O wires are colored blue and black. This will allow for options such as solenoids, relays, opto isolators, LED's and many other input and output connections. See Section 10 for details on using the digital I/O.

Pin #	Color	Function	Input*
1	Red	+V (Main Power In)	
2	Black	I/O	1
3	Brown	RS485B (-)	
4	Black/White	RS485A (+)	
5	Orange	Switch Closure to GND (IN)	4
6	Green	GND (-V of Main Power IN)	
7	White	Opto Sensor Phototransistor (IN)	3
8	Blue	I/O	2
9	Yellow	Opto Sensor LED (Power Out)	

Table 2: Pin Assignments

*****Inputs are labeled 1, 2, 3 and 4 for programming the 'Halt' and 'Skip' Commands. See Section 6 for more information.

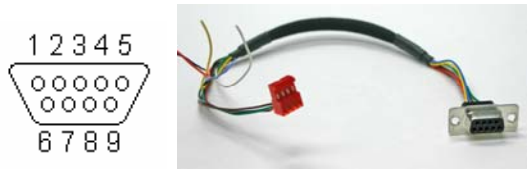


Figure 2: DB-9 Female Cable Connector (Rear View)

8. SETTING THE CURRENT

CAUTION! *DO NOT SET THE CURRENT ABOVE THE MOTOR'S RATED CURRENT, DOING SO WILL DAMAGE YOUR MOTOR AND MAY OVERHEAD THE CONTROLLER*

In order to set the correct current for your motor, you must program the specified amount from HyperTerminal or from the Windows Application.

Current is set based on the Maximum amount of current the controller board can output, which is 1.5 Amps Peak. Below is a table of how much current will be applied to your motor for each setting.

Percent		Motor's Current Rating (Amps)	Driver's Equivalent Current (Amps)
10%	=	0.105	0.15
20%	=	0.21	0.30
30%	=	0.315	0.45
40%	=	0.42	0.60
50%	=	0.525	0.75
60%	=	0.63	0.90
70%	=	0.735	1.05
80%	=	0.84	1.20
90%	=	0.945	1.35
100%	=	1.05	1.50

Table 3: Equivalent Driver Current

Example: If you have a motor that is rated at 0.85 Amps, then you would want to set the current to 80% of the maximum current. This is what you would program; if you are speaking to Address Number 1 (you can change the address using the Red Dial at the back of your unit):

`/1l80m80R`

WARNING! *Setting the Current to a value greater than the Motor's rated current will damage your motor, and may overheat the Controller.*

PLEASE NOTE:
The column labeled "Driver's Equivalent Current" refers to the Peak Current, which is simply 1.4 * Motor's Rated Current. In actuality, you are inputting more current than the Motor's Rated current (1.4 times more). This will give you optimal performance, specifically for microstepping. Microstepping refers to anything more than half-stepping.

9. GETTING STARTED

1. Connect the DB-9 Female Cable to the back of the IMC17 unit.
2. On the opposite end of the DB-9 cable, there is a red 4 pin female connector. Connect this female mating connector to the header of the converter card (RS485 to RS232 Converter Card).
3. Then connect two wires from the converter card to a power supply. Plus and minus signs should already be allocated on the converter card.
4. Connect the Male end of a serial cable to the converter card, and the Female end to the serial port on a PC (This cable is not provided in the kit.)
5. Turn on your Power Supply (See *Figure 3* below).

WARNING! DO NOT DISCONNECT THE DB-9 CABLE FROM THE IMC17 UNIT WHILE POWER IS STILL BEING SUPPLIED. THIS MAY CAUSE DAMAGE TO THE INTERNAL DRIVER BOARD.

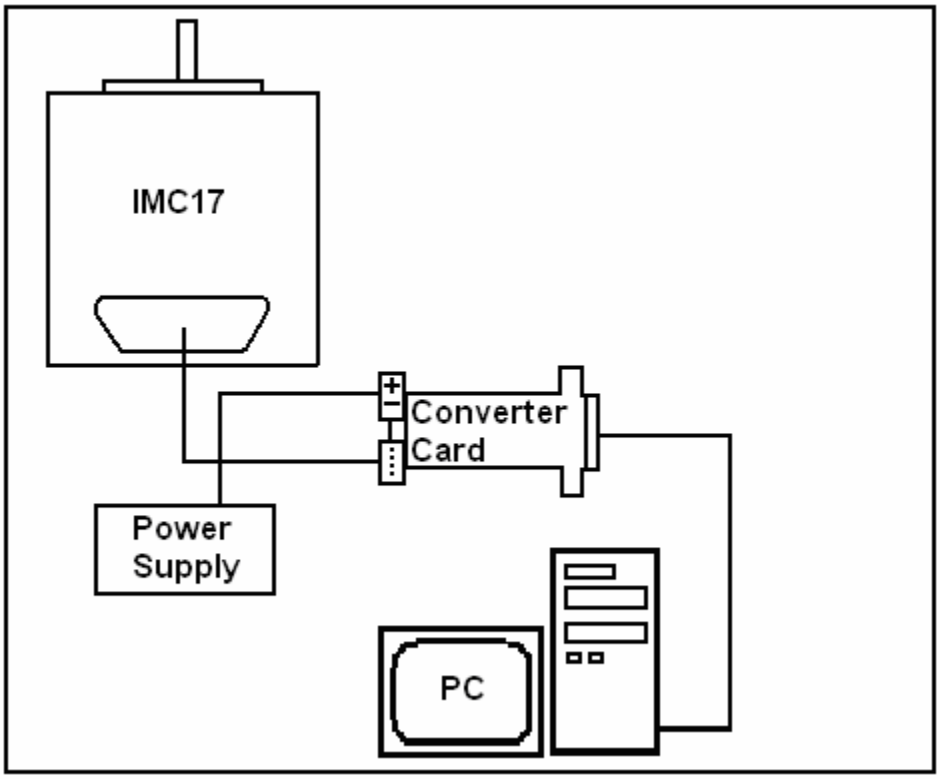


Figure 3: Connecting the IMC17

HYPERTERMINAL SETUP

Please follow these steps in order to properly set up HyperTerminal:

1. Open a terminal from your PC by following these steps:

   Start Menu → Programs → Accessories →
   Communications → HyperTerminal

2. Assign a name for your terminal
3. Make sure your COM connection corresponds to the same COM connection as your PC serial port (i.e. COM 1, COM 2, etc.)
4. Set your Port Settings to default (i.e. 9600 baud, 8 data bits, no parity, 1 stop bit, no flow control) To get to your Port Settings, perform the following steps:
   File → Properties → "Connect Using: COM 1" → Configure
5. Turn on local echo by going to:

   File→ Properties →Settings → ASCII Setup:

   Click on the box for "Echo Typed Characters Locally" and click on the box for "Send Line ends with line feeds". These options will be useful when typing commands in HyperTerminal.

6. Click OK
7. Now you can type your commands
8. Example: **`/1A10000R`**
   - This will run driver 1 to the Absolute position 10000
   - The computer should respond with this string of characters to indicate that the command was received correctly: '/0@ □'
   - A full list of commands is available in the IMC17 Command List document. It is available at www.rmsmotion.com → Download → IMC17.

10. USING THE DIGITAL I/O

The IMC17 contains two fixed inputs and two digital bi-directional I/O's. It is possible to receive input from external devices such as sensors, switches, or PLC outputs. Outputs such as relays, solenoids, LED's, and PLC inputs may be controlled from the IMC17 unit.

All Input and I/O lines feature internal 20k Ω pull-up resistors. The I/O lines in addition feature a switch closure to ground, which can be used to drive loads connected externally of up to 2.4 mA at 24V, or as low as 0.5 mA at 5V. The I/O lines should not be actively driven high, because of the internal switch closure to ground. (Use a pull-up resistor of 300 Ω or greater and an open collector style pull-down on these particular lines).

INPUT CONNECTIONS

Connecting a Switch Push Button

A Red Capped Switch Push Button is provided in the Designer's Kit. After completing the 'Getting Started' section, you are now ready to program an input. Please refer to the 'Soldering Accessory Pieces' on page 16 in order to use your switch push button. Figure 4.1 shows a circuit schematic of how the switch push button is configured.

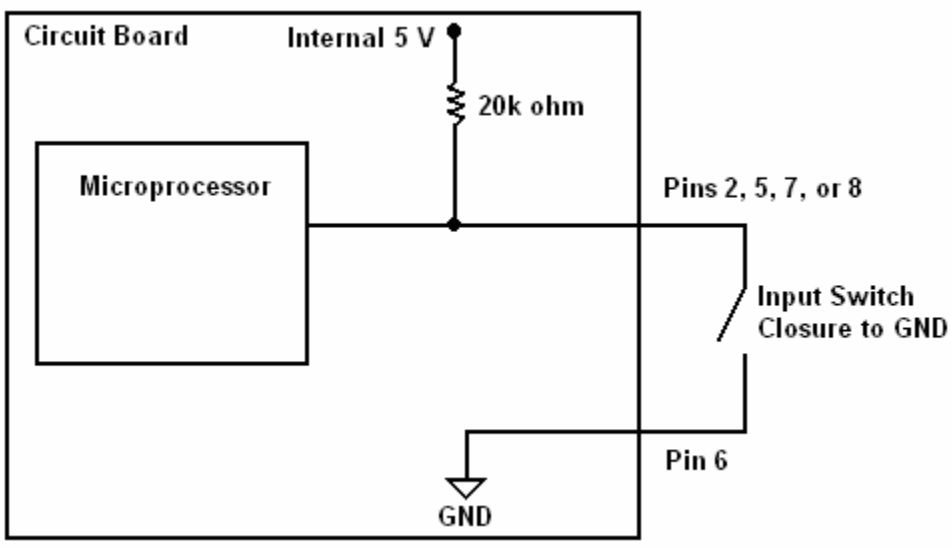


Figure 4.1: Switch Push Button

Using HyperTerminal, here is how to program your Unit to move 1000 steps in the positive direction each time you press the button. This code will work when using Pin 5 as your switch closure to ground. Assuming the motor address is set to 1 (see 'Changing Addresses' in Section 10), type in the following code:

/1gH04P1000G0R

Connecting a Direct Digital Circuit

This enables the user to have one unit's output to be another unit's input. Below, Figure 4.2 shows the input driver on the left, and the output driver on the right.

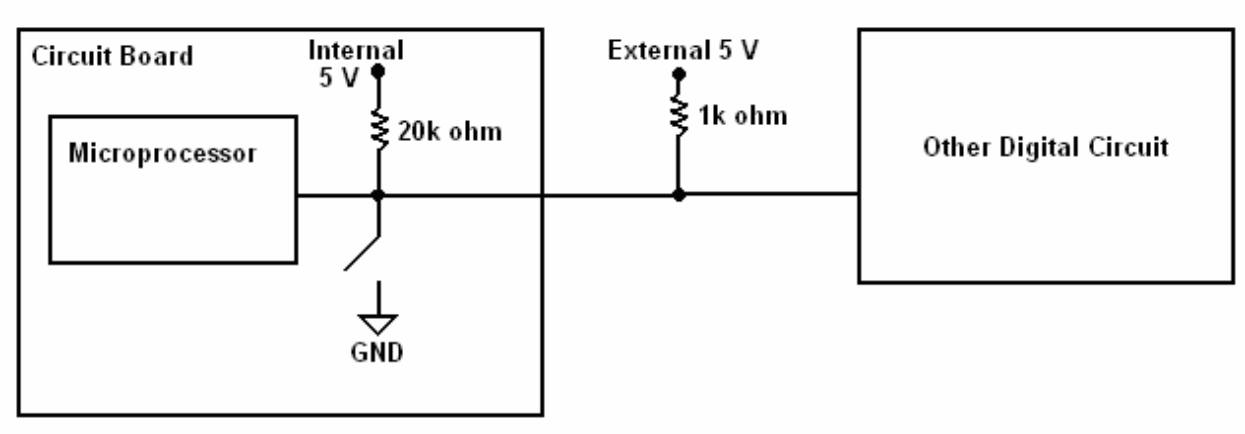


Figure 4.2: Direct Digital Circuit

Be aware of the possibility of a ground shift between the output driver and the input driver. The wiring between the two drivers should be a short 6 inch distance. If your wiring is too long, it is possible that the ground voltage on the input side may be as high as 2 VDC, thus not enabling this connection to work properly. To connect motors further apart, use an opto isolator, as shown in the following section. It is also recommended to connect multiple motors as shown on page 16.

Connecting an Opto Isolator

Connection of two motors is possible using an opto isolator. Be sure to use a transistor opto that has a current, Ic > 1mA at IF = 20mA.

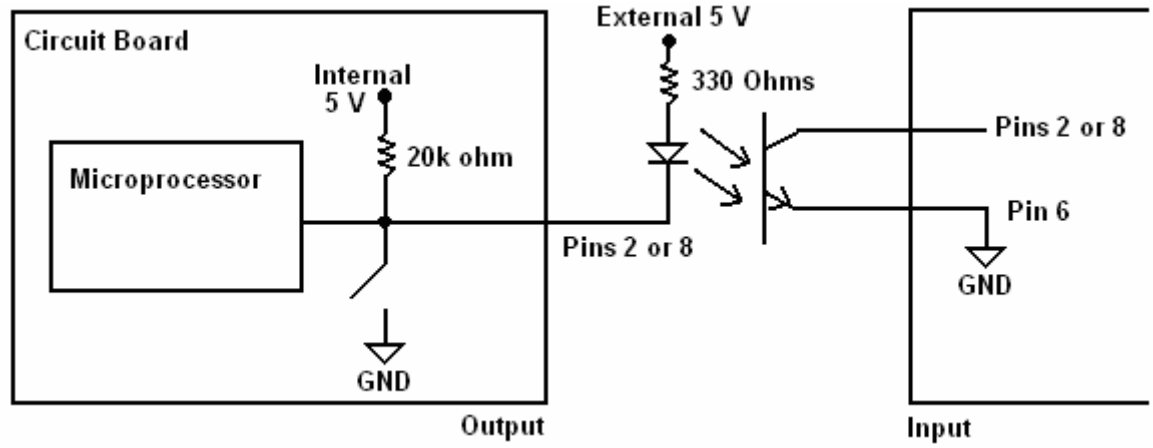


Figure 4.3 Connecting an Opto Isolator

OUTPUT CONNECTIONS

Connecting an Opto Sensor

An Opto Sensor is provided in your developer's kit. After completing the 'Getting Started' Section, you can now program the Opto in HyperTerminal. Please refer to page 18 for Soldering the Opto.

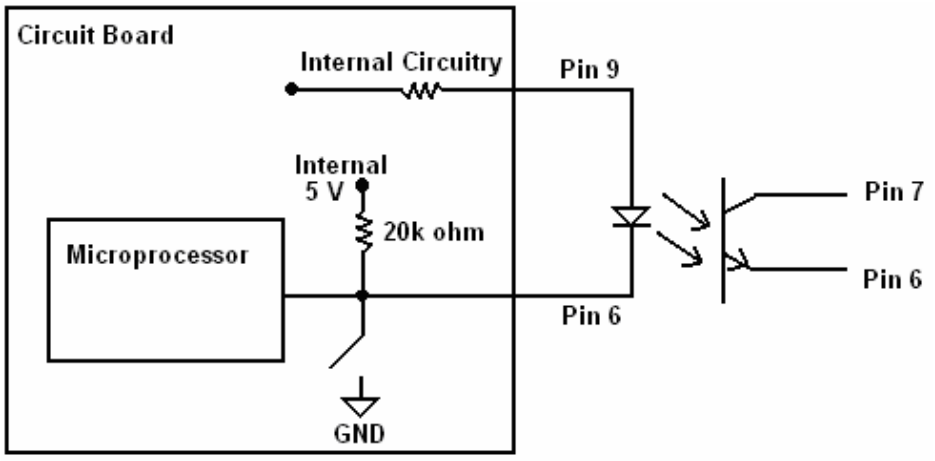


Figure 4.4: Connecting the Opto Sensor

Here is how to program your Unit to wait for a switch closure on pin 2, home the stepper to the opto, move to position 1000, then move to position 0:

/1H01ZA1000A0R

Driving a Relay

It is necessary to place a 1N4001 diode across the relay as shown in order to protect against inductive spikes.

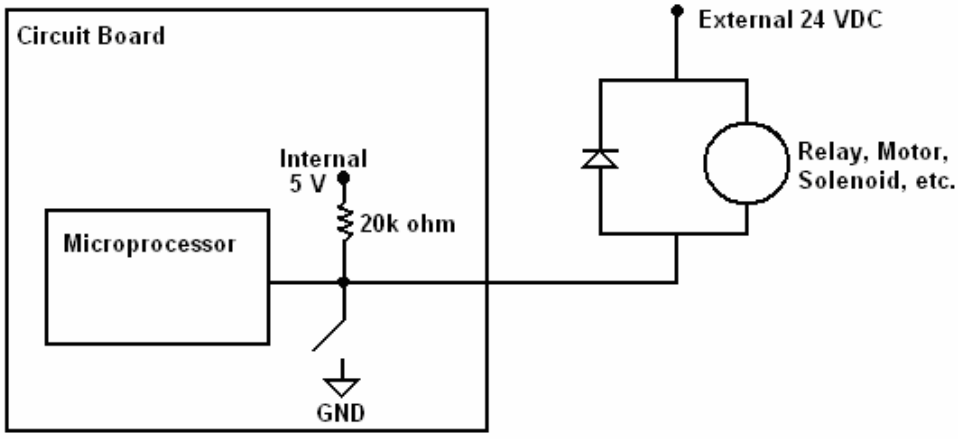


Figure 4.5: Driving a Relay

Example: To activate the relay, then wait for 500 milliseconds, and then deactivate the relay, type in the following commands:

/1gJ3M500J0G5R

CONNECTING A LED TO AN I/O

Since the Inputs and Outputs are tied down together to make a bi-directional I/O line, there is about 0.002 Amps running through the line when it should be off. When connecting an LED, there is a slight chance that the light may appear dim when it is should appear to be completely off. If this happens, an easy solution is to tie the negative side of the LED to ground, and then put a 2.2k Ohm resistor in series up to your External power supply. See figure below:

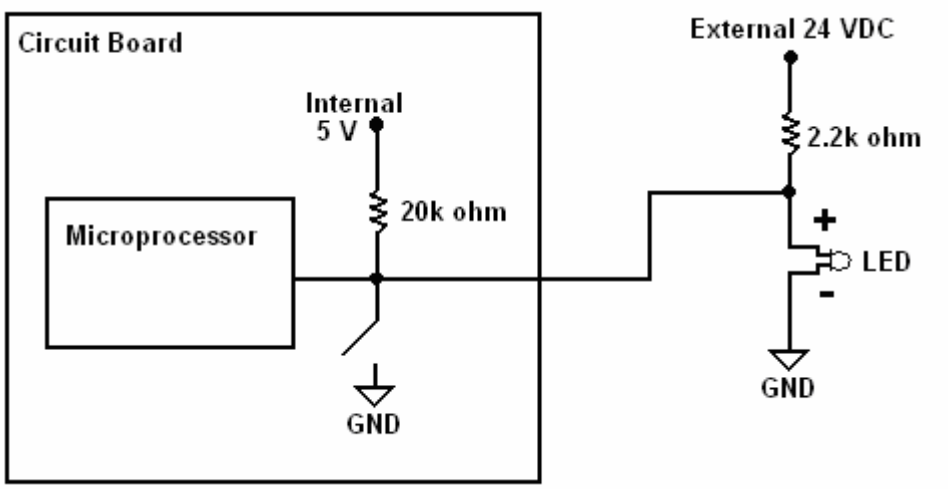


Figure 4.6: Connecting a LED

When using this particular connection, the commands for turning on and off the drivers are switched. /1J0R will turn on both drivers, and /1J3R will turn off both drivers.

Another way to connect the LED is to use a 20V Zener Diode in series with the main external power supply, shown below:

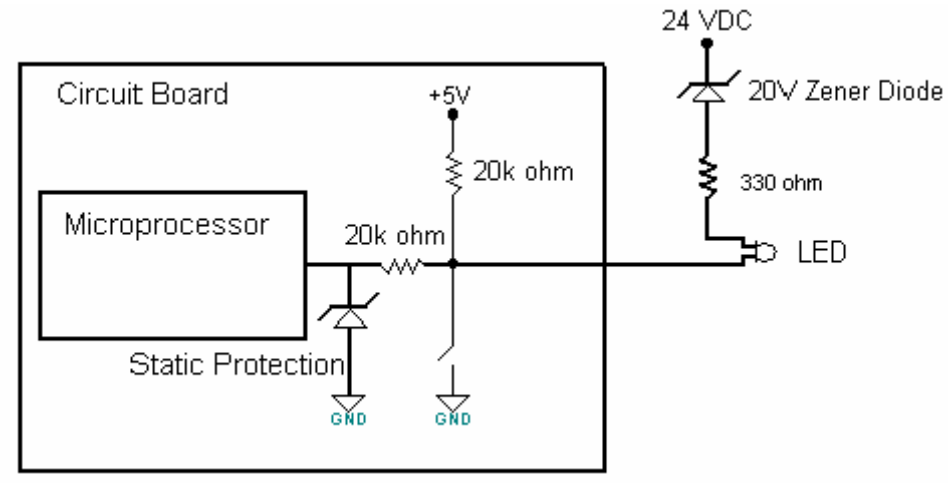


Figure 4.7: Connecting the LED using the Zener Diode

CONNECTING MULTIPLE IMC17 UNITS

- Ensure that each unit has its own unique address if you choose to have different motions per unit. If you choose to run the same motion on multiple units, then the same address for each unit may be used.
- Figure 5 shows a basic schematic of the connections between two motors and the converter card

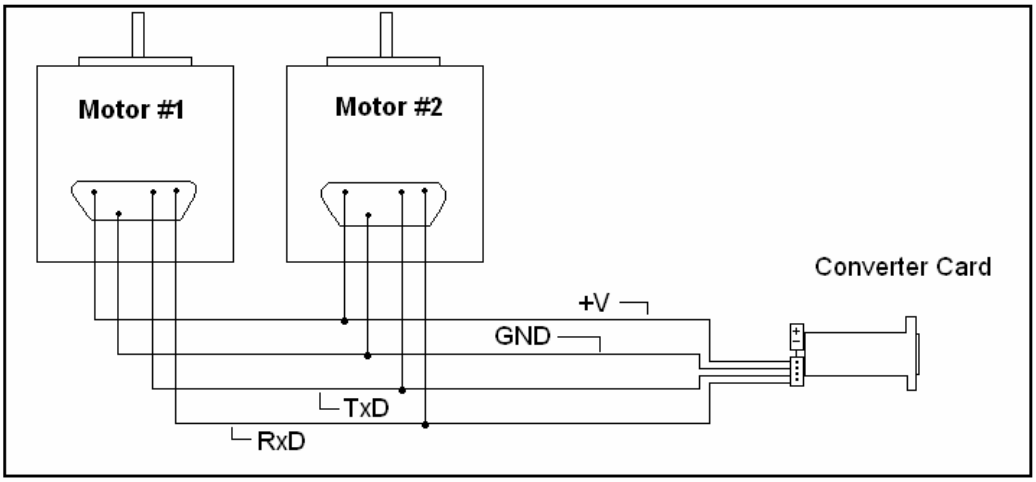


Figure 5: Connecting Multiple IMC17s

CHANGING THE ADDRESS

- Locate the red channel selector dial on the back of IMC17.
- Using a small flat screwdriver, turn dial so the arrow points to the desired channel
- Make sure that the arrow is directly pointing to the desired address, and is not in between 2 addresses.
- See *Figure 6* below

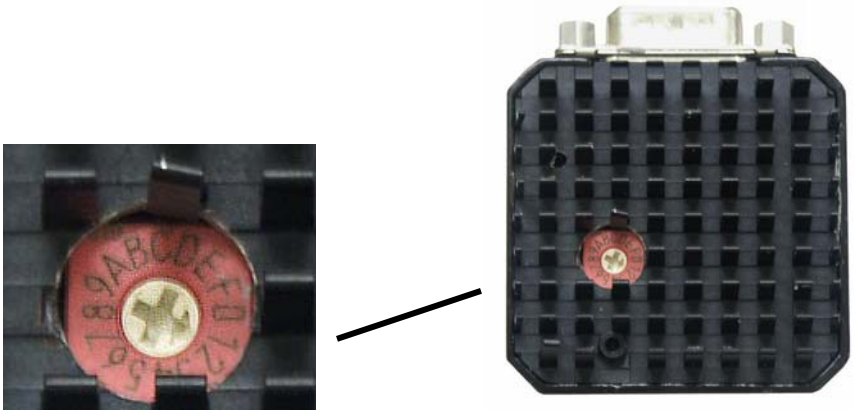


Figure 6: Address Selection Dial

11. SOLDERING ACCESSORY PIECES

PUSH BUTTON

The Push Button needs to be soldered to wire 5 (Orange) and wire 6 (Green). There is no difference in polarity for the Push Button; therefore, wire 5 and 6 can be interchanged with the push button terminals.

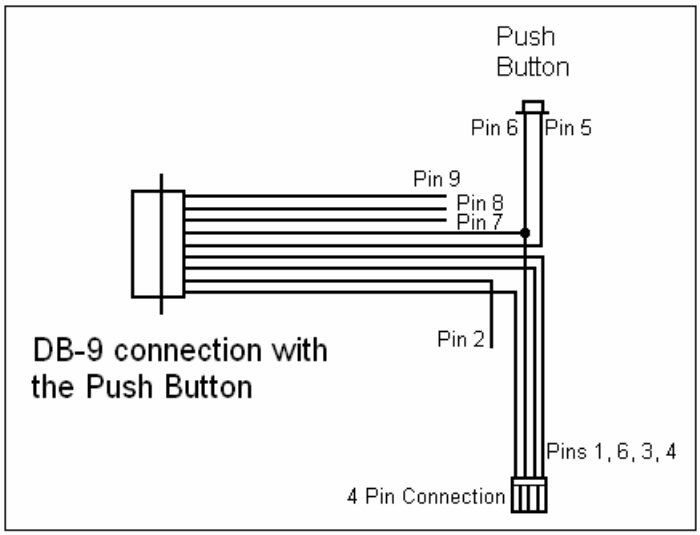


Figure 7.1: Push Button Connection Schematic

Now you can program your switch push button. See 'Examples' in Section 6.

Note: When programming a specific input, note that:
Input 1 corresponds to Pin 2
Input 2 corresponds to Pin 8
Input 3 corresponds to Pin 7
Input 4 corresponds to Pin 5

See the commands list on how to program switches and halt statements.

OPTO SENSOR

The opto sensor uses wire 6 (Green), wire 7 (White), and wire 9 (Yellow). Four wires are already connected to the opto sensor, colored Red, Black, White, and Green. On the opto sensor, wires Green and Black are both Ground, which need to be soldered together, then soldered to wire 6 (Green) on your DB-9 cable. Then the red wire on the opto sensor needs to be soldered to wire 9 (Yellow) on your DB-9 cable. And the white wire on the sensor needs to be soldered to wire 7 (White) on your DB-9 cable.

Opto Sensor	Cable
Green →	Green
Black →	Green
Red →	Yellow
White →	White

Table 5.1: Opto Sensor Pins

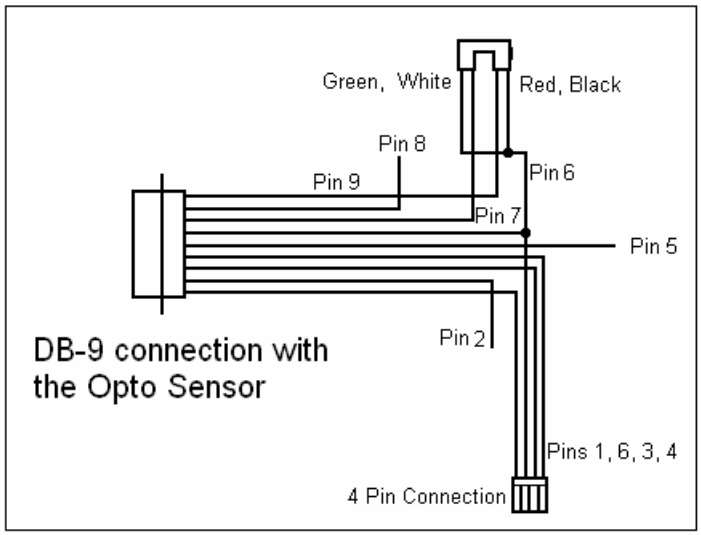


Figure 7.2: Opto Sensor Schematic

Now you can program your Opto Sensor. See 'Examples' in Section 7.

Note: When programming a specific input, note that:
Input 1 corresponds to Pin 2
Input 2 corresponds to Pin 8
Input 3 corresponds to Pin 7
Input 4 corresponds to Pin 5

12. Encoder Option Specifications

Features:

- 400 cycles per revolution (CPR)
- 1600 pulses per revolution (PPR)
- 2 channel quadrature TTL squarewave outputs
- Optional index (3rd channel)
- Single +5V supply

The IMCE17 uses an US Digital E2 Encoder. The pin assignments are as follows:

Pin Number	Function
1	Ground
2	Index
3	Channel A
4	+ 5 VDC
5	Channel B

The E2 encoder requires a separate +5 Volt power supply, the R164 controller cannot provide a strong enough source of power.

In order to use the Indexer as a reference to home, connect Pin 2 from the encoder to one of the inputs on the controller. From the controller's side, for best results use Pin 5, the switch closure to ground.

In addition, use a pull-down resistor (10k Ω) to ensure that the controller will recognize the difference between high and low (4.85V and 0.5V).

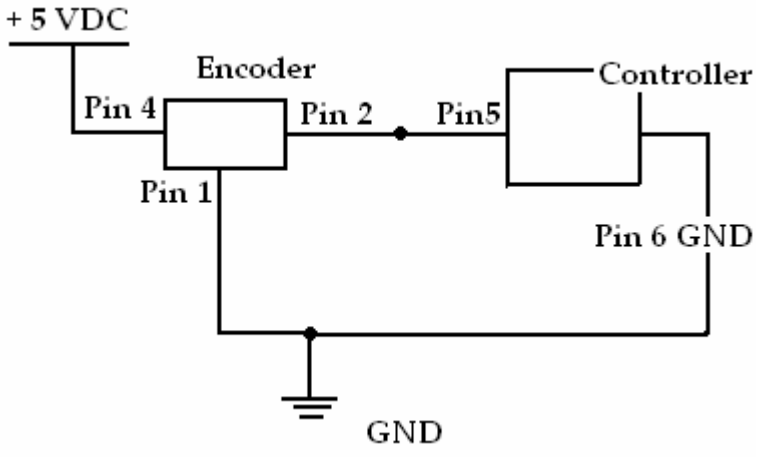


Figure 8: Encoder Pin Connection Schematic

After successfully connecting the encoder to the controller, now you can program the controller to run continuously. The motor will stop movement when the indexer goes high. This will send the high signal to Pin 5 on the controller.

If using channel 1, use this command to start the homing routine:

/1P0R

13. TROUBLESHOOTING

I cannot type anything in HyperTerminal:

Check to make sure that you have the correct COM Port selected. Are you using Windows 95? Windows 95 has had problems with its HyperTerminal. Use an operating system of Windows 98 or higher. Are you working on a Laptop? Sometimes there is a shift in Ground on Laptop Serial Ports. Pin 5 on the Serial Port is Ground. Confirm that this is connected to a true ground.

Did you check the Properties in HyperTerminal? Go to File → Properties → Settings → ASCII Setup, and check the box 'Echo Typed Characters Locally'.

Is the LED flashing on the back of the IMC17 (heatsink side)? A flashing light indicates it is waiting for Commands. The other steady LED indicates that Power is being supplied to it. If neither of these is on, then check your Power Supply.

Check the Converter Card. Use a voltmeter to measure if Power is coming into the card correctly and if power is being supplied to the Controller properly.

I can type in HyperTerminal, but nothing happens:

Check the LED's. If neither is on, check the Power Supply. If one is steady, and the other LED is flashing, then Power is OK.

Is the IMC17 controller set to the correct Address? Check the Red Dial on the back of the unit. The white arrow should be pointing to the corresponding Address. Confirm that the arrow is not between 2 Addresses.

Check the Converter Card. Use a voltmeter to measure if Power is coming into the card correctly and if power is being supplied to the Controller.

Assuming the Properties of HyperTerminal are set as described in the Manual (Getting Started), does HyperTerminal respond with a string of characters?
/0'□ indicates that the command is terminated
/0@□ indicates that the command was received correctly
/0C□ indicates that the command is out of range
/0b□ indicates bad command

The IMC17 is accepting commands, but Motor will stall in the middle of a command:

This means there is not enough current being supplied to the Motor. Use the m and l (lower case L) commands to change the current, or run the Motor at a Lower Speed (V command).