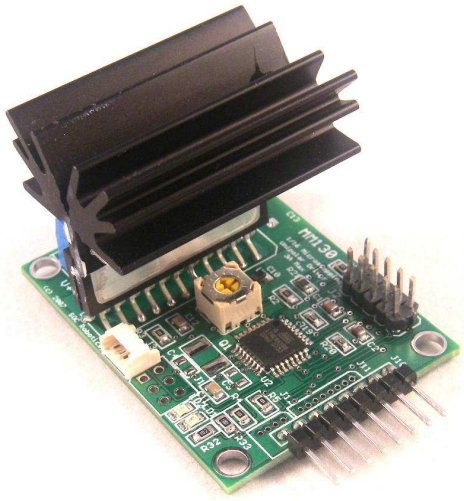
MM130 Single Axis Stepper Motor Driver

1/16th Microstepping 3A Unipolar Driver

Technical Reference Manual

PCB Rev 1.1

Motor Control Software Version 0.98



www.soc-robotics.inc

Manual Revision 0.91
March 2009

Warranty Statement

SOC Robotics warrants that the Product delivered hereunder shall conform to the applicable SOC Robotics Data Sheet or mutually agreed upon specifications and shall be free from defects in material and workmanship under normal use and service for a period of 30 days from the applicable date of invoice. Products that are "samples", "design verification units", and/or "prototypes" are sold "AS IS," "WITH ALL FAULTS," and without a warranty. If, during such warranty period, (i) SOC Robotics is notified promptly in writing upon discovery of any defect in the goods, including a detailed description of such defect; (ii) such goods are returned to SOC Robotics, DDP SOC Robotics facility accompanied by SOC Robotics Returned Material Authorization form; and (iii) SOC Robotics examination of such goods discloses to SOC Robotics satisfaction that such goods are defective and such defects are not caused by accident, abuse, misuse, neglect, alteration, improper installation, repair, improper testing, or use contrary to any instructions issued by SOC Robotics. SOC Robotics shall (at its sole option) either repair, replace, or credit Buyer the purchase price of such goods. No goods may be returned to SOC Robotics without SOC Robotics Returned Material Authorization form. Prior to any return of goods by Buyer pursuant to this Section, Buyer shall afford SOC Robotics the opportunity to inspect such goods at Buyer's location, and any such goods so inspected shall not be returned to SOC Robotics without its prior written consent. SOC Robotics shall return any goods repaired or replaced under this warranty to Buyer transportation prepaid, and reimburse Buyer for the transportation charges paid by Buyer for such goods. The performance of this warranty does not extend the warranty period for any goods beyond that period applicable to the goods originally delivered.

THE FOREGOING WARRANTY CONSTITUTES SOC ROBOTICS EXCLUSIVE LIABILITY, AND THE EXCLUSIVE REMEDY OF BUYER, FOR ANY BREACH OF ANY WARRANTY OR OTHER NONCONFORMITY OF THE GOODS COVERED BY THIS AGREEMENT. THIS WARRANTY IS EXCLUSIVE, AND IN LIEU OF ALL OTHER WARRANTIES. SOC ROBOTICS MAKES NO OTHER WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE SOLE AND EXCLUSIVE REMEDY FOR ANY BREACH OF THIS WARRANTY SHALL BE AS EXPRESSLY PROVIDED HEREIN.

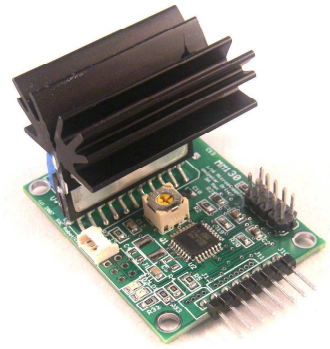
Limitation on Liability

Notwithstanding anything to the contrary contained herein, SOC Robotics shall not, under any circumstances, be liable to Buyer or any third parties for consequential, incidental, indirect, exemplary, special, or other damages. SOC Robotics total liability shall not exceed the total amount paid by Buyer or SOC Robotics hereunder. SOC Robotics shall not under any circumstances be liable for excess costs of re-procurement.

1935A–08/00/5M

Table of Contents

1.0 Introduction



Features:

- Unipolar 3A Stepper Motor Driver
- Maximum motor drive voltage 44V
- Full, half, quarter, eight and sixteenth microstepping
- Maximum 3 A per phase
- Backlash compensation built-in
- On board 20MHz RISC AVR processor – Atmega168
- SPI Port
- TWI I2C Communication Protocol Built-in
- 16K Internal Program Flash
- 512 Internal EEPROM
- 1K Internal SRAM
- ISP Programming Port
- GNU C Compiler, Third Party Commercial C Compiler
- Extremely Small form factor (2.10x1.54 in)
- 5VDC @ 12ma Power input for logic
- Motor drive input 10-44VDC at 3A

Hardware

The MM130 is a 1/16th Microstepping Unipolar 3A per phase stepper motor driver that converts step and direction signals to the appropriate high voltage stepper motor drive signals. The board is designed to operate at 5VDC. Motor voltage can vary from 10-44VDC. The motor drive chip is controlled by a dedicated onboard 8 bit RISC processor (Atmega168) that provides various expansion and enhancement features. Applications can communicate with the board via an I2C interface or using encoded commands on the step and direction inputs. A Limit switch input can be monitored and acted on but are turned off by default. An EStop input is can also be monitored to stop step operation.

The MM130 consumes about 12ma in active state.

Motor Control Software

The MM130 is shipped with a sophisticated software application that controls the SLA7078 driver chip, monitors step/direction inputs, limit switch state, encoded commands on the step/direction input lines and commands received on the I2C port. The MM130 can operate as a single axis G Code processing system by acting on commands sent to it via the I2C or Step/Direction input lines. See the section on software operation for a full description of the complete command structure.

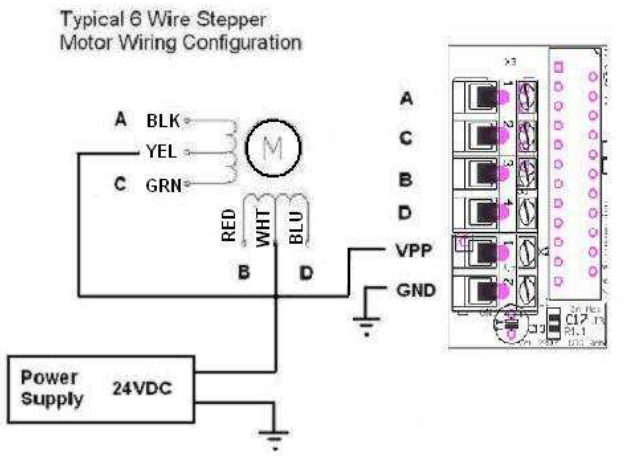
The control program in the MM130 processor can be re-Flashed using programming cable Part No. MCM-8 and the appropriate host software MK4Prog.exe.

The MM130 can be custom programmed in C using either a GNU C Compiler, AVR Studio V4.13 or higher with GNU C integrated with the IDE or a third party IDE such as ICCAVR from ImageCraft.

Stepper Motor Connection

Motor Drive voltage should be between 10-44VDC. A potentiometer sets average current and should be adjusted while the board is running. A heat sink provides radiant cooling so forced air cooling is not required in most applications. At extreme operating conditions forced air cooling should also be used. The diagram below shows stepper motor connections for the SM2006 and SM3006 stepper motors – wire color for other stepper motors may vary although most coil phase diagrams are similar. It is a good idea to wire a fuse between the motor power supply and the MM130. Try not to turn motor power on if the logic side of the MM130 is not turned on. If the motor will not turn then check your wiring.

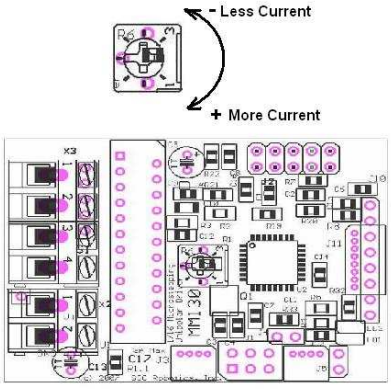
MM130 Motor Connection for SM2006/SM3006



Adjusting Motor Drive Current

The MM130 has a single potentiometer for setting motor drive current. Turning the potentiometer clockwise increases current while turning the potentiometer counterclockwise decreases current. By default the potentiometer is set in the middle of the range at the factory. The best setting is determined by running the motor at different speeds under full load and increasing current until the motor doesn't misstep. Note that this setting will change if the step mode changed. Use the built in ramp function (described in detail in the command section) to set the minimum and maximum step rate and the rate of change of step. It is possible to adjust the potentiometer in such a way that the control chip delivers two much current to the motor which may result in motor overheating so it is best not to turn the potentiometer fully clockwise. If the motor missteps at high current settings then your motor maybe undersized for your application and can not develop enough torque – in this case use a bigger motor.

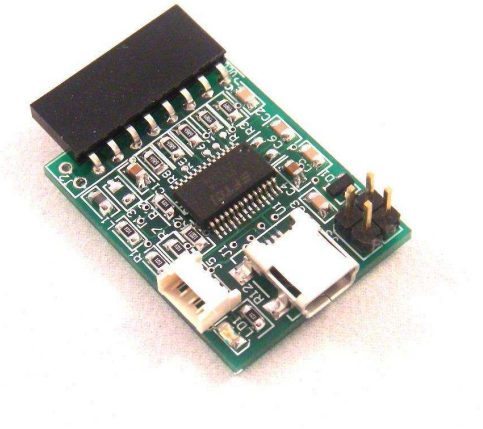
MM130 Motor Current Potentiometer Setting

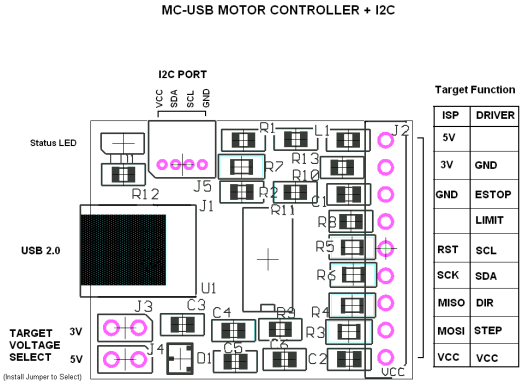


MM130 Controllers

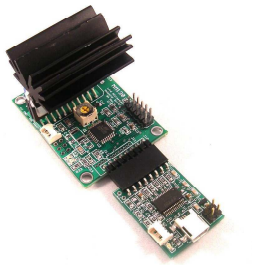
Although the MM130 can operate standalone it usually requires a controller to send step and direction input signals to it. SOC Robotics has several controllers compatible with the MM130 - MC-USB, MK1, MK4, MK14, MK54 and MK200. Each controller supports increasing levels of sophistication and control flexibility. The MK1 is a simple breakout board that allows three MM130's to be attached and controlled. The MK4 is a four axis breakout board with four auxiliary output ports, four limit switch inputs and one Estop input. The MK14 is a special version of the MK4 with a USB 2.0 interface. The MK54 is a high performance G Code processor with 10/100BaseT, CAN and USB 2.0. The MK200 is a ultra high performance G Code processor with onboard vision processing. For more information on our line of controllers and to help determine which one is best for your application go to our web site or contact the firm.

With the MC-USB stepper motors can be controlled via a host PC USB port.



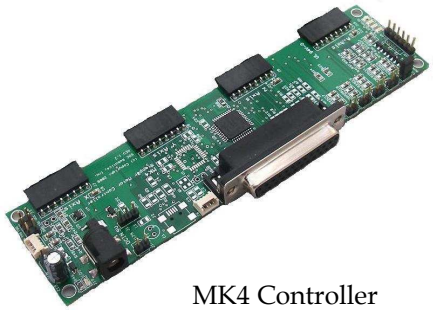


The MC-USB is a USB 2.0 based controller that generates the step/direction signals to control the MM130. An I2C communications bus allows a host PC to communicate with the MM130 in real time allowing step mode changes, limit switch detection and aux IO monitoring and control.



MM130 with MC-USB Controller

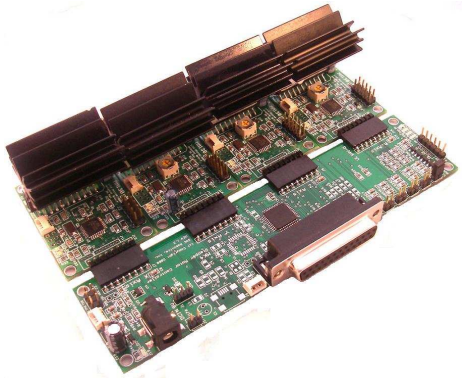
The MK4 is an optional parallel port controller with connectors for four MM130 drivers. The MK4 has four aux open collector outputs and five limit switch inputs all set/read via the parallel port. The MK4 is an excellent choice to attach up to four MM130's to a PC's parallel port and be controlled using Mach3.



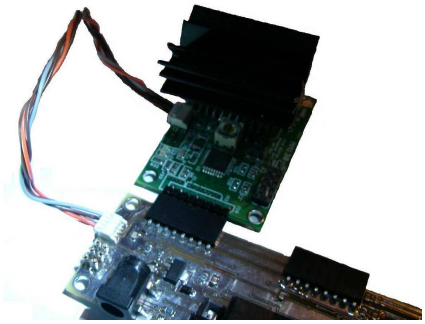
MK4 Controller

The MK14 is a USB 2.0 version of the MK4 with an onboard processor that accepts up to four MM130 drivers. The MK14 has four aux open collector outputs and five limit switch inputs all read via the parallel port. The onboard processor (AT91USB162) accepts step commands via USB and allows the host PC to communicate with the individual MM130 drivers and send high commands directly to each driver. In this case the parallel port interface can be bypassed and commands sent directly to each MM130 processor using a standard serial communications protocol.

MK14 Controller

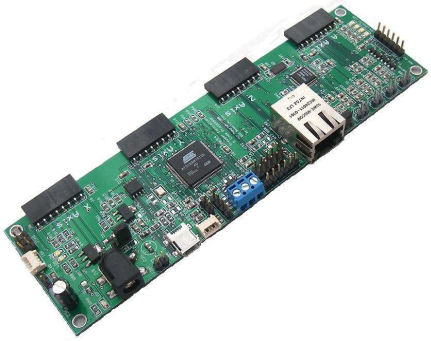


MK14 with 3 MM130's and MM133

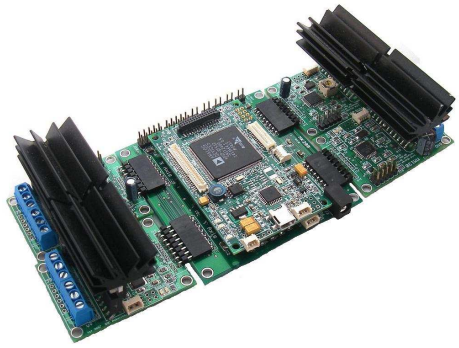


MM130 Programming Cable

The MK54 is a high end G Code processor with onboard ARM7 processor running 55MHz. The ARM7 processor executes G Code directly eliminating the need for a PC based G Code processor. The MK54 is capable of receiving commands via a USB 2.0 interface, 10/100BaseT Ethernet or CAN.



The MK200 is a high end DSP based G Code processing platform with built-in vision processing is capable of driving the MM130's at their maximum rate of 80,000 microsteps/second. The MK200 is intended for high end applications that require extremely fast stepper motor operation and/or real time vision processing.



The correct controller for your application depends on many parameters. If you require help in choosing a controller or need additional information please contact the company.

2.0 MM130 Detailed Description

2.1 Introduction

The MM130 is a compact single axis stepper motor microstepping driver with a dedicated onboard processor. The processor responds to step/direction signals on the primary inputs, receives and interprets commands received on the I2C interface and processing sensor data received on the auxiliary IO port.

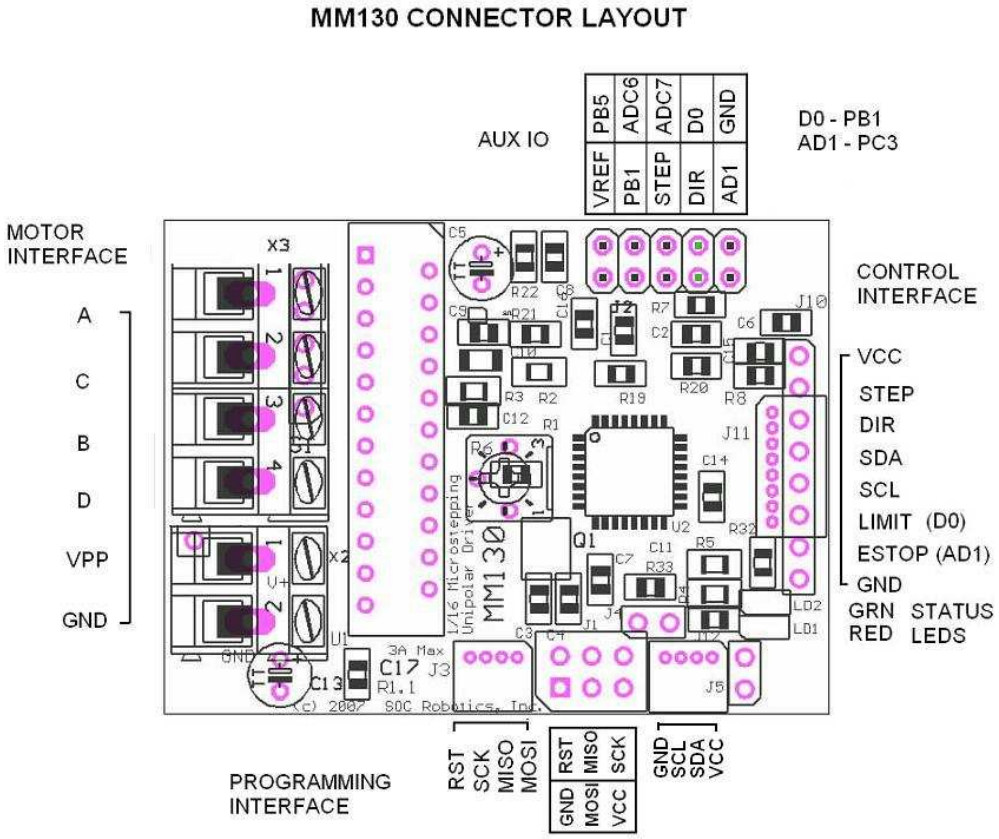


Figure 2-1. **Primary Components on top and bottom sides of PCB.**

2.2 Processor

The MM130 has an 8bit RISC AVR processor (Atmega168) running at 20MHz. Note that the Atmega168 requires 5VDC to run at 20MHz. The program running in the Atmega168 monitors the step/direction input lines and driver the SLA7078 microstepping driver chip. Commands supported by the processor firmware is described in the software operation section of this manual.

2.3 TWI Port

The MM130 has a TWI I2C port for communicating with smart peripherals or other I2C peripherals. The TWI port uses a 4 pin Molex connector with power and ground so the MM130 can power other peripherals or be powered itself via this connector. Default I2C address is 0x20.

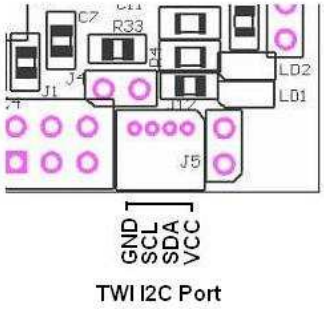


Figure 2-2. **MM130 I2C port pin assignments.**

2.4 Auxiliary IO Port

The MM130 has an auxiliary IO port.

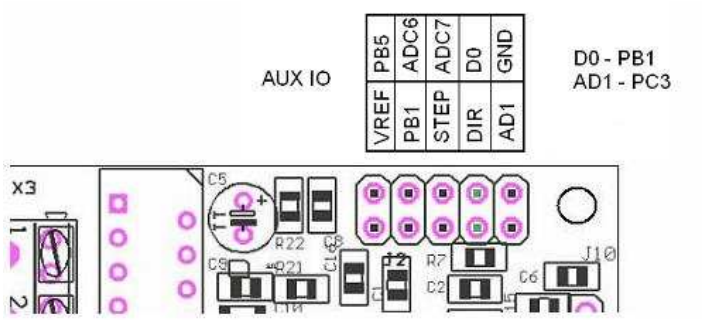


Figure 2-3. **MM130 Auxiliary IO port pin assignments.**

2.5 ISP Programming Port

The MM130 can be programmed using an ISP10 Parallel Port programming adapter and a CISP adapter. The ISP10 and CISP are not supplied with the board but can be purchased separately. Alternatively the MM130 is programmed using the MK4 and MK54 plus a programming cable.

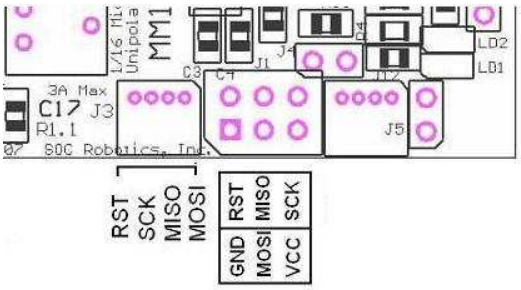


Figure 2-4. **CISP ISP Adapter and correct attachment to the MM130 .**

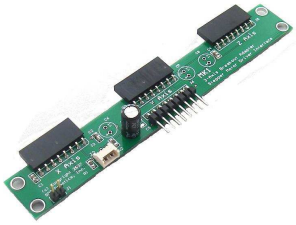
2.6 Molex Connectors

The Wasp uses two small Molex picoBlade 4 pin and 5 pin male connectors with 1.25mm pin spacing (4 pin Molex Part No. 53048-0410 - Digikey Part No. WM1744-ND - 5 pin Molex Part No. 51021-0500 – Digikey part no. WM1745-ND). These connectors mate with female Molex 4 and 5 pin housing connectors (4 pin housing Molex Part No. 51021-0400 - Digikey Part No. WM1722-ND – 5 pin housing Molex Part No. 51021-0500 - Digikey Part No. WM1723-ND).

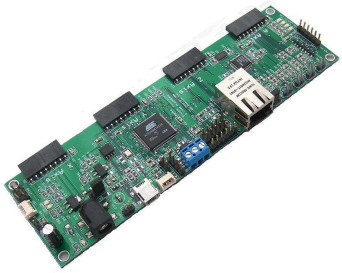
The housing connectors has two different crimp terminal types: 26-28AWG (Molex Part No. 50079-8000 - Digikey Part No. WM1722-ND – Crimp tool 63811-0300) and 28-32AWG (Molex Part No. 50058-8000 - Digikey Part No. WM1775-ND - Crimp tool 63811-0200).

2.7 Related Peripherals

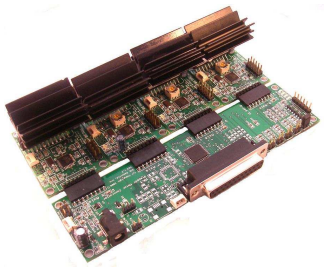
The MM130 is controlled directly by the USB10, MK1, MK4, MK4USB, MK54 and MK200. The MK4 attaches to a PC parallel port and provides an attachment point for up to four MM130's. The MK4 also support limit switch inputs and EStop. The MK4USB is a version of the MK4 that has a USB 2.0 interface to allow direct communication between each attached MM130 and a host PC for both motion control such as step/direction and configuration control such as changing step mode.



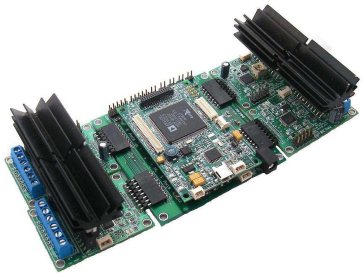
MK1



MK54



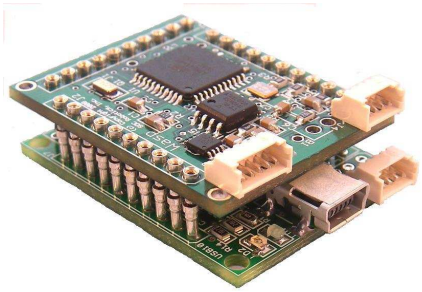
MK4 with 4 MM130's



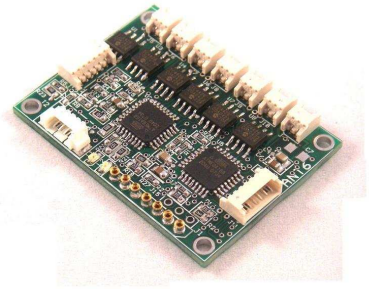
MK200 plus P0 DSP with four MM130's

MK1 with Wasp and USB10 G Code processor

The MM130 can communicate with other SOC Robotics embedded processors such as the Wasp, WaspARM, SAM48, AmberM, SmartLCD, Ant6, Cricket and a host of other embedded processors.

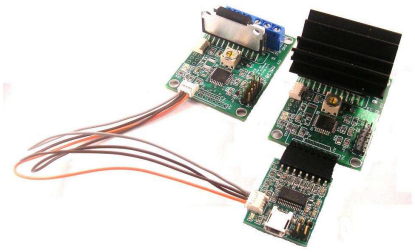


Wasp on USB10.

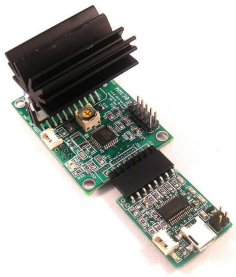


Ant6 6 Channel H-Bridge Controller

USB10 is a USB 2.0 device with an onboard AVR processor – the AT90USB162. The USB10 converts commands sent to it via the USB to step/direction commands that can drive the MM130.



Wasp mounted on a USB10 communications board.



Ant6 6 Channel H-Bridge Controller

2.8 Applications

The MM130 is a unipolar stepper motor controller that responds to step and direction signal inputs but is also able to communicate with other smart controllers via the I2C interface. In fact, large numbers of MM130's can be ganged together and be controlled via a single I2C master. This enables the creation of rich motion control systems beyond simple three or four axis system.

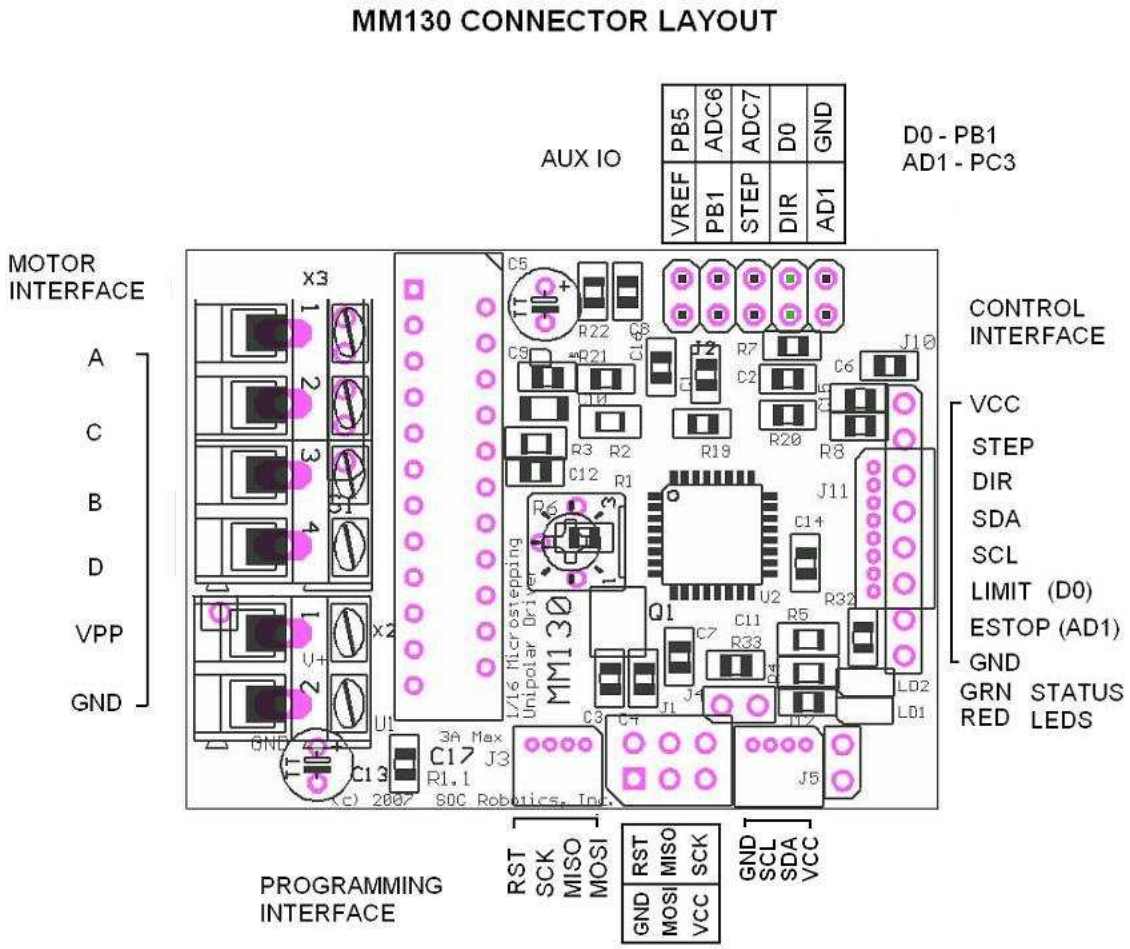
The MM130 can also accept sensor input on the Auxiliary IO and ISP programming lines provided an upgrade to a closed loop design in which sensor feedback is used to tune stepper motor operation.

The MM130 can be combined with other processor technology such as the Wasp or WaspARM with onboard 3-axis accelerometers to create sophisticated embedded motion control systems.

3.0 MM130 Hardware Expansion Port Summary

3.1 Introduction

The MM130 has three I/O ports: a motor control port, programming port and an auxiliary IO port as shown in the connector layout diagram below. Two alternative configurations of the board are available: Configuration 1 replaces the 0.1″ motor control port header with an 8 pin picoBlade Molex connector and configuration 2 replaces the motor 0.1″ motor control port header with a 4 pin picoBlade Molex connector for the I2C port.



3.2 Motor Control Port

Motor control port is routed to connector J10. Step and direction signals are fed to the board on this port along with main power (5V DC) and ground, I2C lines and limit switch input. The I2C lines allow a controller such as the MK4-USB, MK54 or MK200 to communicate with the MM130 and set configuration features such as step mode, limit switch recognition, step/direction polarity are also routed to the port so that an attach controller can communicate with the MM130. Limit switch inputs

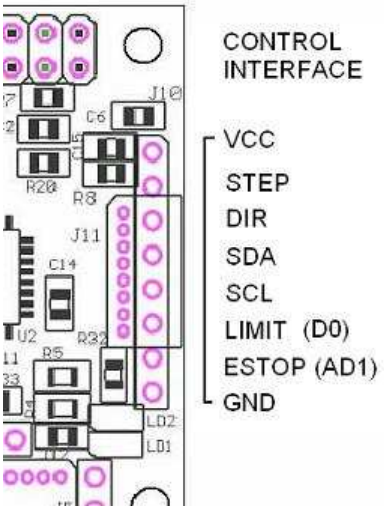


Figure 3-3. Step/Direction **Control Port Pin Assignment J10.**

3.3 Auxiliary IO Port

The auxiliary IO port provides analog input, digital IO and step/direction inputs. The digital IO can be used to connect external sensors such as rotary and linear position sensors or to allow the MM130 to control external devices such as relays. The analog inputs allow the MM130 to measure external analog sensors.

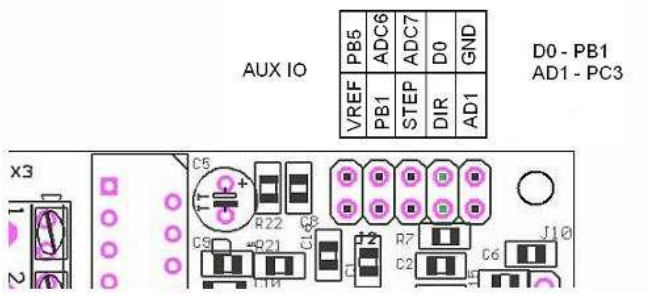


Figure 3-4. **Expansion Port Pin Assignment J2.**

3.4 I2C Expansion Port

The MM130 has an I2C communication port on connector J3 and J10. The I2C connector is not installed on the standard MM130 but is available by special order. The MM130 can be controller via the I2C port. See the section of Drive Commands for more information. Note that the I2C signals are also on the Motor Control Port connector. The MM130 is a slaved I2C device and can be controlled by other SOC Robotics Smart Peripherals such as joysticks, LCD display controllers data acquisition modules.

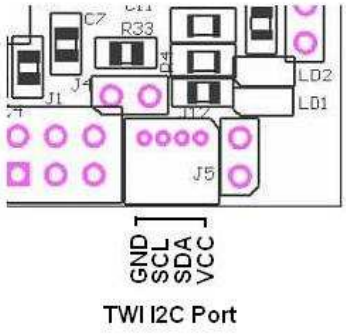


Figure 3-5. **TWI I2C Port with 4 Pin Molex picoBlade Connector.**

3.5 ISP Programming Port

The MM130 has two ISP Programming ports – a six pin header compatible with the Atmel 6 pin header ISP programming standard and a four pin picoBlade Molex connector compatible with the MK4, MK14, MK54 and MK200 controllers. The ISP programming port allows the onboard Atmega168 processor to be reprogrammed. See the Atmel ISP programming specification for detailed ATmega168 programming procedures.

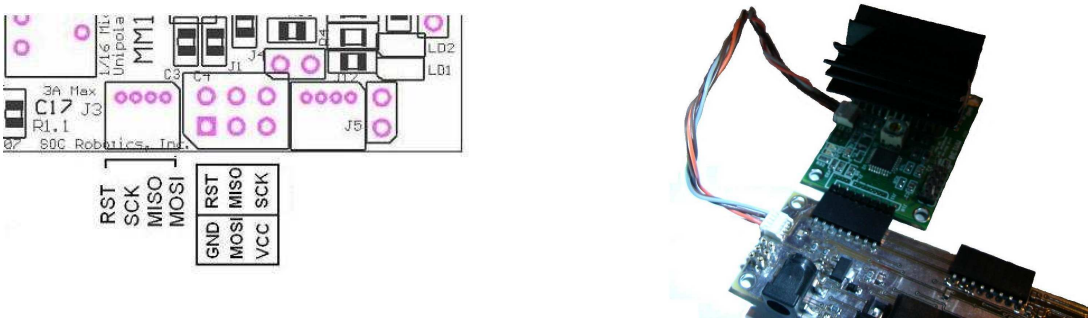


Figure 3-6. **ISP Programming Port and MCM-8 programming cable.**

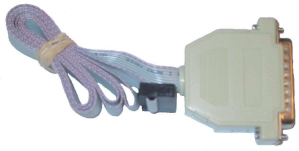


Figure 3-7. **ISP10 Parallel Port programming adapter.**

4.0 Software Operation

4.1 Theory of Operation

The MM130 is driven by step and direction signals applied to connector J10. The MM130 is shipped with half step mode enabled. The MM130 supports full, wave, half, quarter, one eight and one sixteenth step mode and can be changed by sending a command to the board via the I2C interface or using encoded commands on the step/direction signal lines. The direction signal must be set before the step pulse – the step pulse is a positive going raising edge that is held for at least 5useconds. The direction signal must not be changed during the step high level or the MM130 will enter command recognition mode.

The board power is applied through connector J10 and requires 5V regulated DC. The standard board runs at 20MHz.

The motor drive chip SLA7078PR is a unipolar microstepping driver with built in current sense resistors and logic to control drive current given a drive voltage. A potentiometer allows the user to set the operating point of the driver. The SLA7078 is designed to drive 2-3A motors but should be able to drive smaller motors without any trouble.

The SLA7078 has Motor Coil Short-Circuit Protection and Motor Coil Open Protection circuits that prevents damage to the device by turning the drive chip off under such conditions. Power must be cycled to re-enable the chip.

4.2 Auxiliary IO

The MM130 has a number of auxiliary digital and analog inputs plus digital outputs that allow connection of rotary or linear position sensors and limit switches. The processor can be field programmed with new functions to process signals from these sensors and limit switches. The current software version does not support external sensors.

4.3 Step Drive Connector

The MM130 controls an attached stepper motor by converting Step and Direction input signals on connector J10 to four Power Mosfets in the SLA7078PR chip. By default the step signal is an active high pulse lasting at least 5 useconds with optional noise reduction logic implemented in software. A high level on the Direction input causes a clockwise step. Default step mode is half step – step modes can be changed by sending commands encoded in the step/direction signal lines.

J10 also has I2C and limit switch inputs. Operation of I2C is described in the next section. Stepper motor operation can also be controlled through the I2C interface independent of signals on J10 and by interpreting signal encoded on the step/direction lines. Limit Switch inputs by default are not recognized but can be enabled by the user.

4.4 Driver Commands

The MM130 accepts commands in two different communication formats: commands encoded on the step/direction lines (step/direction line toggling) and commands sent through the I2C interface.

Command Features

A script based parameter configuration utility allows the user to set the configuration options of each axis on an individual basis. Configuration parameters can be stored in on-chip EEPROM for automatic power

up configuration control. A special built-in test/setup mode provides a simplified method for setting up the board. Configuration commands can be sent to each motor controller in real time using a special communications protocol based on the Step and Direction lines. A bi-directional communication utility provides a real time link with MM130 processor allowing dynamic change and update of system parameters on the fly. An API library is available to provide application developers access to the configuration features of the new software.

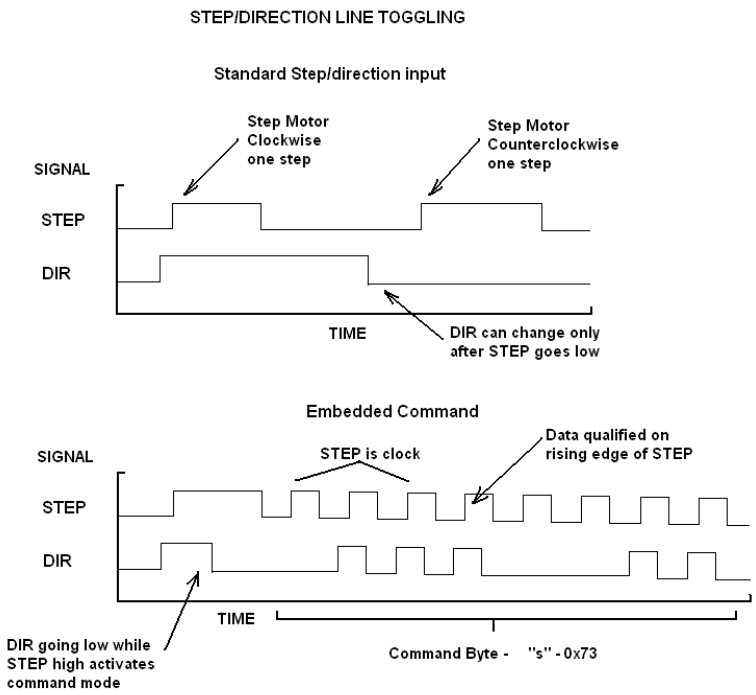
- New communication protocol for bidirectional data flow between the host and MM130 via the STEP/DIRECTION and Limit Switch lines and I2C
- The following parameters can be set by the user while the MM130 is operating:
    - Pass through mode enabled for noisy environments
    - Limit switch detection– enable, disable, smart enable, polarity settable
    - EStop switch detection– enable, disable, polarity settable
    - Step mode selection – wave, full, half, quarter, eight, sixteenth
    - Automatic motor shut off – enable, disable and time period settable
    - Motor power - enable/disable
    - Step rate, direction and number of steps to execute
    - Maximum step rate
    - Distance traveled for each step in inch or metric
    - Feed Rate – ipm or cmpm
    - Linear or rotary distance commands in inch or metric
    - G1 G code commands accepted directly
    - Acceleration curve settable
    - Single Step - Clockwise or counterclockwise
    - Step Movement Direction Polarity – Clockwise or counterclockwise
    - Step signal polarity setting - high pulse, low pulse
    - Direction polarity settable – high – clockwise, high –counterclockwise
    - Backlash set
    - Communication mode – enable/disable
    - Save configuration settings in EEPROM

- Configuration parameters can be saved in EEPROM
- Stored configuration parameters can be read from the motor EEPROM using MK4Prog.exe and used to create a text based configuration script file

Step/Direction Line Toggling

Step/direction line toggling uses the step and direction signal lines to embed commands in step sequences. If the step and direction inputs are both high and the direction line is brought low before the step line is brought low the driver interprets this to mean a one byte command is about to be sent to the driver. The step line is then toggled as a clock input while data is encoded on the direction line. The driver does not interpret this pattern as a series of steps but as a command byte. By sending a string of

bytes in this manner it is possible to send commands to the driver embedded within a stream of step signals. The exact format of the step/direction command byte is shown in the diagram below.



I2C Commands

An alternative communication protocol is to send and receive commands via the I2C interface lines. I2C is a Philips two wire (SDA and SCL) communications protocol that is widely supported by both host processors and embedded processors. See related documentation for a detailed description of I2C. The I2C interface defaults to address 0x30 but can be changed by sending a broadcast set address command. Note that each MM130 on a single I2C chain must have a unique address.

Both communication protocols support a common command set. Commands are typically ASCII character sequences. There are three classes of commands – Setup, Drive and Global. Setup commands select various configuration options, Drive commands activate stepper motor operation while Global commands configure the communications address of the board and activate a special multiple MM130's stepper motor operational state.

4.5 MM130 Command Overview

The MM130 processor responds to commands sent to it via the Step/Direction signal lines. Using a simple encoding procedure it is now possible to communicate with the Motor Controller via the Step, Direction and Limit switch lines. All commands are lower case letters and numbers. It is possible to change Motor Controller operation and configuration on the fly while the controller is running. Configuration changes take effect immediately.

The MM130 supports the following commands. A detailed description of each command follows. A command consists of two or more ASCII characters. Some commands require entry of integer or floating point numbers. A floating point number entered as an integer is interpreted correctly. Integer or

floating point number must be terminated with a semicolon ";" character. Note that all commands are lower case characters. Commands are acted on immediately.

On power up the MM130 processor loads configuration settings from on chip EEPROM. If these settings are changed using the commands below note that the processor does not automatically store the new settings in EEPROM – the user must explicitly do this using the Save Settings command. If power is lost all changed settings will be lost.

A desktop program called `GStepPP.exe` is used to send commands to the MM130. `GStepPP.exe` sends commands to a specific driver by entering the "c" character following by an axis identifier (x,y,z,a) character followed by one or more of the command sequences below all entered on one line. The command is sent to the driver when the enter key is pressed. Command recognition mode must be enabled in the driver for commands to be recognized. An example of a command sequence is:

```
-cxs4
-cxcss
```

The first command instructs the x-axis driver to set the step mode to quarter step. The following command then stores this new setting in EEPROM so the next time the driver is powered up quarter step mode is activated. A few of the command functions can be halted by entering the character sequence `cxq`.

If the control software driving the MM130 does not support command mode then for increased reliability in noisy environments command recognition mode can be disabled by sending the command mode disable command `cxcci`.

Command Summary

All commands are printable ASCII characters.

s – Change step mode, 1-Full, !-FullF, 2-Half, @-HalfF, 4-quarter, 8-Eight, 6-Sixteenth `cxs4`
c – Enter Configuration Change mode:
  s – Save settings to EEPROM (needs two 's') cxcss
  t – Pass through mode, a-active, i-inactive
  r – Reset Motor drive chip
  h – Turn motor power off
  c – Command Recognition mode, a-active, i-inactive
  n – Noise reduction mode, a-active, i-inactive
    l – loop count for noise reduction – default 2, (int)
  f - Step/direction signals recognition enabled, y–yes, n-no
  i – I2C Mode o–on, f-off, a-address (int)
  e – EStop switch recognition, y–yes, n-no
  l – Limit switch recognition, y–yes, s–smart, n–no
  p – Step polarity, p-positive, n-negative TBD
  d – Direction, c–clockwise, w–counterclockwise
  a – Automatic motor power shut off mode, o-on, f-off
    t – Time after which the motor power is shut off, seconds (float) cxcat25.5;
  b – Set backlash (float) cxcb24;
  m – Step resolution parameters
    r – Resolution (float), cxcmr0.000125;
    n – Minimum pps (float), cxcmn20.0;
    x – Maximum pps (float), cxcmx2300;
    m – Ramp rate (float)
d – Drive motors (enter direction and step rate), e-exit (entered at any time)
  z – Zero absolute position
  i - Set feed rate (ipm), float cxdi10.0;
  c – Step clockwise
  w – Step counterclockwise
  s – Set slow step rate (pps), float
  f – Set fast step rate (pps), float
  m – Constant step cxdmw220000;1300;
    direction, stepmode, number of steps, step rate (pps)
  x – Ramp for n loops using default low high pps cxdx220;2000;4;
    stepmode, steps at each speed, constant step number, loops
  r – Ramp between low and high pps cxdr2300;2000;20;2000;
    stepmode, slow step rate, high step rate, step increment, constant steps
  d – Distance mode cxdd21.25;6.27;
    stepmode, new position, feed rate (ipm)
t – Enter Test mode
  t – Toggle back and forth – all step modes, hit any key to exit
  x – Ramp low to high – high to low – all step modes, hit any key to exit
  v – Ramp low to high – high to low – enter step mode, hit any key to exit
l – Led mode, 1-mode1, 2-mode2, 3-mode3

4.6 Detailed Command Description

Step Mode Command

Set the driver step mode. The MM130 is a microstepping unipolar stepper motor driver capable of full, half, quarter, eight and one sixteenth microstepping.

s – Change Step Mode, 1-Full, !-FullF, 2-Half, @-HalfF, 4-Quarter, 8-Eight, 6-Sixteenth
Set the driver step mode. Step mode is set to full, full high power, half, half high power, quarter, eight or one sixteenth microstepping. The following characters set the step mode:

1 – Full step mode
! - Full step high power mode
2 – Half step mode
@ - Half step high power mode
4 - Quarter step mode
8 - One eight step mode
6 - One sixteenth step mode

Examples: cxs1 cxs6

Note that changing step mode may not guarantee a smooth step transition. Step mode changes take effect immediately.

Drive Motor Commands

Set a motor driver state. The MM130 has a single axis G Code like feature that supports absolute position moves with backlash compensation and step resolution configuration setup.

d – Drive motors (enter direction and step rate), e-exit (entered at any time)
z – Zero absolute position
i - Set feed rate (ipm), float
c – Step clockwise
w – Step counterclockwise
s – Set slow step rate (pps), float
f – Set fast step rate (pps), float
m – Constant step `cxdmw220000;1300;`
direction, stepmode, number of steps, step rate (pps)
x – Ramp for n loops using default low high pps `cxdx220;2000;4;`
stepmode, steps at each speed, constant step number, loops
r – Ramp between low and high pps `cxdr2300;2000;20;2000;`
stepmode, slow step rate, high step rate, step increment, constant steps
d – Distance mode `cxdd21.25;6.27;`
stepmode, new position, feed rate (ipm)

d – Drive Motors (enter direction and step rate), e-exit (entered at any time)

Enter drive motor command function. The drive mode allows step commands to be executed directly by the on board processor. Relative and absolute step position functions are supported. Most of the commands can be terminated early by entering the exit command: `cxe`

z – Zero Absolute Position

Set the internal position counter to zero. This command resets the home position. The absolute position command converts a distance command to the required number of steps based on the backlash and step resolution commands.

i - Set Feed Rate (ipm)
Set the feed rate in inch per minute (ipm). This command sets the step rate based on the step resolution. The parameter is entered as an integer or floating point number.

Example: `cxdi12.25;` - Set feed rate to 12.25 inch per minute.

c – Step clockwise
Step clockwise one step using the existing step mode. Note that if the desired step direction is not clockwise use the Step Direction Polarity command to change the direction.

w – Step counterclockwise
Step counterclockwise one step using the existing step mode. Note that if the desired step direction is not counterclockwise use the Step Direction Polarity command to change the direction.

s – Set slow step rate (pps)
Set the slow step rate in pulses per second (pps). This command sets the slow speed step rate that is used by several other commands such as the ramp command. The parameter is entered as an integer or floating point number.

Example: `cxds1300;` - Set slow step rate to 1300 pps.

f – Set fast step rate (pps)
Set the fast step rate in pulses per second (pps). This command sets the slow speed step rate that is used by several other commands such as the ramp command. The parameter is entered as an integer or floating point number.

Example: `cxdf1300;` - Set fast step rate to 1300 pps.

m – Constant step
Step at a constant rate (pps) given an entered direction, step mode and number of steps. This command allows a set of step parameters to be entered. Variable length parameters such as step rate may be entered as an integer or floating point number. Direction is "c" or "w". Step mode is 1, !, 2, @, 4, 8 or 6.

Entered parameters: direction, stepmode, number of steps, step rate (pps)

Example: `cxdmw220000;1300;` - Step CCW, half step, 20000 steps at 1300 pps
`cxdmc41000;400;` - Step CW, quarter step, 1000 steps at 400 pps

x – Ramp for n loops using default low high pps
Ramp from a low step rate to a high step rate, step at the maximum step rate for the entered number of steps and then step from the high step rate to the slow step rate stepping the entered number of steps at each speed and repeat for the entered number of loops. The slow and fast step rate are entered using the "s" and "f" drive commands. Note variable length command parameters are terminated with a ";" character.

Entered parameters: stepmode, steps at each speed, constant step number, loops

Example: `cxdx220;2000;4;` - Ramp using half step, 20 steps at each ramp

increment, 2000 steps at max rate and loop 4 times

`cxdx410;20000;14;` - Ramp using quarter step, 10 steps at each ramp increment, 20000 steps at max rate and loop 14 times

r – Ramp Between Low and High pps

Ramp from a low step rate to a high step rate, step at the maximum step rate for the entered number of steps and then step from the high step rate to the slow step rate stepping the entered number of steps at each speed and repeat for the entered number of loops. The slow and fast step rate are entered using the "s" and "f" drive commands. Note variable length command parameters are terminated with a ";" character.

Entered parameters: stepmode, slow step rate, high step rate, step increment, constant steps

Example: `cxdr2300;2000;20;2000;` - Ramp using half step from 300 pps to 2000 pps incrementing 20 steps per speed change and step 2000 steps at the maximum rate.

`cxdr1200;700;10;1000;` - Ramp using full step from 200 pps to 700 pps incrementing 10 steps per speed change and step 1000 steps at the maximum rate.

d – Distance Mode

Move to an absolute position at the entered feed rate (ipm) given an entered step mode. The new position is entered as inches, step rate is inch per minute and step mode is 1, !, 2, @, 4, 8 or 6. At first power up the home is set to zero. Positive or negative absolute positions can be entered. Backlash compensation is applied automatically and step resolution defines the number of steps either clockwise or counterclockwise. The function automatically compensates for step mode changes so if a position command says move to absolute position 1.000 at full step and then step to 0.000 at half step the carriage will return the exact starting point.

Entered parameters: stepmode, new position, feed rate (ipm)

Example: `cxdd21.25;6.27; - Ramp using half step from 300 pps to`

Test Mode Commands

Enter test mode. Test mode commands drive the stepper motor through a series of direction toggle and ramp motions.

t – Enter Test mode
- t – Toggle back and forth – all step modes, hit any key to exit
- x – Ramp low to high – high to low – all step modes, hit any key to exit
- v – Ramp low to high – high to low – enter step mode, hit any key to exit

t – Enter Test Mode

Enter the test mode state and interpret any subsequent characters as test mode selections.

t – Toggle Back and Forth – All step modes, hit any key to exit
This command causes the drive to step the motor in the clockwise direction for a defined number of steps and then step in a counterclockwise direction the same number of steps at a pre-defined pps rate starting with full step mode and sequencing through all the step modes. Entering cxq caused the sequence to terminate.

x – Ramp Low to High/High to Low – All step modes, hit any key to exit
This command causes the driver to step the motor through a clockwise then counterclockwise motion while ramping the speed from a minimum pps rate to a maximum pps rate sequencing through all step modes from full to sixteenth step.

v – Ramp Low to High/High to Low – Hit any key to exit
This command causes the driver to step the motor through a clockwise then counterclockwise motion while ramping the speed from a minimum pps rate to a maximum pps rate at an entered step mode and loop count.

Entered Parameters: step mode, loop count

Example: `cxtv210` - Ramp in half step mode and repeat 10 times.

LED Display Mode Command

The user can set the display mode of the LEDs to one of three different display status modes. This function has not been implemented in the current software version. Mode 1 is active.

l – Led mode, 1-mode1, 2-mode2, 3-mode3
Set the led display mode. Command character is "l". Sets mode 1, 2 or 3.

Mode 1 (default)
Green led displays status of stepper motor power: on - power being supplied to motor, off – no power to motor. Red led displays step rate and blinks when driver is waiting for a step signal– led blinks for each step signal sent to the motor. The slow blinking Red led indicates the driver processor is alive and waiting for a step command.

Mode 2
Green led displays status of stepper motor power: on - power being supplied to motor, off – no power to motor. Red led display step direction – on – clockwise, off – counterclockwise.

Mode 3
The on or off condition of the Green and Red LEDs is set using commands. The next character selects the Red or Green Led and the state (on or off).

Entered Parameters: led, state (on-o, off-f)

Examples:
cxlro - Turn the Red Led on
cxlgf - Turn Green Led off.

Configuration Change Commands

The configuration change commands set the basic default operating modes of the **MM130**. The user can turn command mode recognition on or off (if command recognition mode is turned off commands are no longer recognized), save settings to EEPROM, enable a noise reduction mode, enable step/direction recognition, set EStop and limit switch recognition, change direction polarity, automatically turn holding torque on/off and time, set backlash and step resolution, minimum and maximum pps and ramp rate.

c – Enter Configuration Change mode:
- s – Save settings to EEPROM (needs two 's') cxcss
- t – Pass through mode, a-active, i-inactive
- r – Reset Motor drive chip
- h – Turn motor power off
- c – Command Recognition mode, a-active, i-inactive
- n – Noise reduction mode, a-active, i-inactive
  - l – loop count for noise reduction – default 2, (int)
- f - Step/direction enable, y–yes, n-no
- i – I2C Mode o–on, f-off, a-address (int)
- e – EStop switch recognition, y–yes, n-no
- l – Limit switch recognition, y–yes, s–smart, n–no
- p – Step polarity, p-positive, n-negative TBD
- d – Direction, c–clockwise, w–counterclockwise
- a – Automatic motor power shut off mode, o-on, f-off
  - t – Time after which the motor power is shut off, seconds (float) cxcat25.5;
- b – Set backlash (float) cxcb24;
- m – Step resolution parameters
  - r – Resolution (float), cxcmr0.000125;
  - n – Minimum pps (float), cxcmn20.0;
  - x – Maximum pps (float), cxcmx2300;
  - m – Ramp rate (float)

c – Enter Configuration Change Mode
Top level command. Any characters entered after the "c" character are interpreted as configuration change commands.

s – Save Settings to EEPROM (needs two 's') `CXCSS`
Save all configuration parameters in EEPROM. Once saved in EEPROM the next power cycle of the board loads the new settings at power up. Note that if any configuration parameters are changed they remain in effect as long as the power is applied. Configuration changes are not automatically stored – the user must use this command to save any configuration changes. Note that two "s" characters are required.

t – Pass through mode, a-active, i-inactive
Set Pass Through Mode to active or inactive. Pass through mode is a special operating state of the MM130 that disables command recognition on both the I2C and Step/Direction lines. This mode when active ensures the step and direction signal levels are recognized under all conditions. If the setup and hold times for command recognition on the step/direction can not be guaranted then the MM130 make accidently enter command mode. This may happen in noisy environments. Once Pass through mode is activated no commands are recognized – the only way to re-enable command mode is to re-flash the processor. Pass through mode is only activated at power-up so if this setting is changed from inactive to active it does not take effect until the board is repowered.

r – Reset Motor Drive Chip

Sets the motor drive chip to a known start state. Note that the motor rotor may be in an intermediate start position is any microstepping modes were selected.

h – Turn motor power off
This command turns motor power off. Motor power is reapplied on the next step input.

c – Command Recognition Mode, a-active, i-inactive
Command recognition mode is enabled or disabled by this command. Once command recognition mode has been disabled the only way to re-enable is to either re-powered to the board (if this setting was not saved) or re-program the processor. In noisy environments it may be necessary to disable command recognition mode so noise spikes on the step/direction lines are not mistakenly interpreted as step signals. By default command recognition mode is active. If a desktop application that is generating setup and holds times are not within specification then if may be necessary to turn Command Recognition mode off – note this applies to Commands encoded in the Step/Direction input lines only and not recognition of commands received on the I2C lines.

n – Noise Reduction Mode, a-active, i-inactive
l – loop count for noise reduction – default 2, (int)
In noisy environments a special noise reduction mode can be enabled. Noise reduction adds delay to the step pulse recognition logic. If step pulses are shorter than 5usec noise reduction mode should not be used. The loop count parameter sets the duration of the noise reduction period. By default noise reduction is disabled.

f - Step/Direction Enable, y–yes, n-no
Recognition of step/direction signals can be enabled or disabled. Step/direction signal recognition maybe disabled if step commands are encoded on the step/direction lines or I2C lines. By default Step/direction recognition is enabled.

i – I2C Mode o–on, f-off, a-address (int)
Commands received on the I2C signal lines can be enabled or disabled. By default I2C recognition is disabled. The default I2C address can also be set. By default I2C mode is disabled.

e – EStop Switch Recognition, y–yes, n-no
EStop switch detection can be enabled, disabled and active state polarity is settable. When disabled the state of the EStop switch has no effect on motor operation. If enabled the motor will stop immediately and remained stopped as long as the EStop switch is active. Motor operation will resume if the EStop switch active state is removed or EStop Switch detection is disabled. The polarity of the EStop Switch active state can be set high or low. The default is active low as the EStop switch input is pulled high with a 10K resistor. By default EStop recognition is disabled.

l – Limit Switch Recognition, y–yes, s–smart, n–no
Limit switch detection can be enabled, disabled or smart enabled and active state polarity is settable. When disabled the state of the limit switch has no effect on motor operation. If enabled the motor will stop immediately if the Limit Switch becomes active and the motor must be moved manually off the stop position (or enable limit mode can be disabled and the driver instructed to move). If Smart Enabled the motor will respond to move commands in the direction away from the stop. The polarity of the Limit Switch active state can be set high or low. The default is active low as the limit switch input is pulled high with a 10K resistor. By default Limit switch recognition is disabled.

p – Step Polarity, p-positive, n-negative TBD

Change the polarity of the step input signal. This function has not been implemented in the current version of the software. By default step polarity is an active going high pulse.

d – Direction Polarity, c–clockwise, w–counterclockwise
Change polarity of the direction input signal. Sets the direction the stepper motor turns when the direction signal is high. Note that depending on how the motor was wired clockwise and counterclockwise directions may be flipped. This command allows the user to flip the direction of rotation. By default the high level selects clockwise rotation.

a – Automatic Motor Power Shut Off Mode, o-on, f-off
t – Time after which the motor power is shut off, seconds (float)
After a step operation completes holding torque can be activated or deactived. If activated torque the time hold torque is applied can be set. The time duration is set in 0.1 second increments from a 1/10 of a second to 6000 seconds. Depending on the application if holding torque is applied for long time periods driver and motor overheating may result.

b – Set Backlash
If the backlash of the lead screw is known this command calibrates the absolute position command when direction changes. The backlash is the number of steps required to move the thrust nut in direction change operations.

Example: `cxcb24;` `- Set backlash to 24 steps`

m – Step Resolution Parameters
r – Resolution (float), `cxcmr0.000125;`
n – Minimum pps (float), `cxcmn20.0;`
x – Maximum pps (float), `cxcmx2300;`
m – Ramp rate (float)

The distanced traveled for a single step is used to calibrate the absolute position move command. Typically this parameter depends on the tpi of the lead screw and step mode. For example a 20tpi screw moves a carriage by 0.00025 inch in full step mode. The parameter is entered as a floating point number. The minimum pulse per second (pps) setting is used by the test and drive commands to set the slow step rate. The parameter is entered as a floating point number. The maximum pps setting is used by the test and drive commands to set the high step rate. The Ramp rate is a parameter that sets the acceleration for the test, drive and absolute position commands.

Linear or Rotary distance commands - inch or metric
Move an entered distance in inch or cm measurement at a defined feed rate. The number of steps is set by the Distance traveled by step parameter.

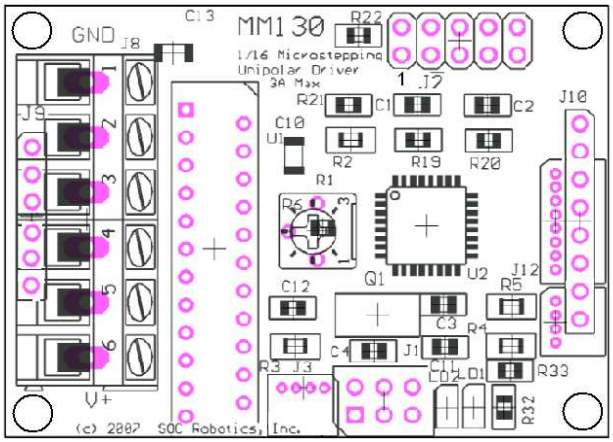
G Code commands accepted directly
Linear G Code commands can be sent to the controller directly and executed. For example a command like N12G1X1.035F3.5 will move the carriage 1.035 inch left at a feed rate of 3.5 inch per minute.

Acceleration Curve Settable
Define an acceleration curve by setting the time it takes to move from one step rate to another step rate. This is a linear acceleration curve.

5.0 Electrical and Mechanical Description

5.1 Component Layout

Components are mounted on the top side of the board. Not all components may be mounted. See the section on optional components for more information.



5.2 Electrical Specifications

Electrical

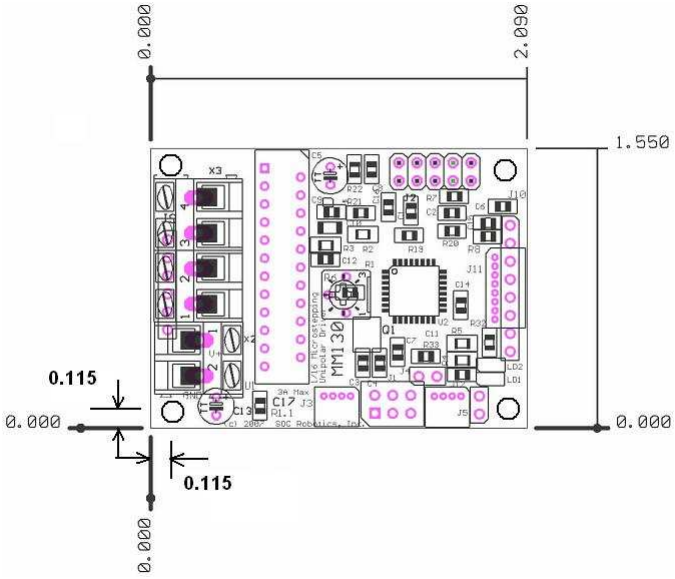
Input power: 5VDC @ 20ma

Mechanical

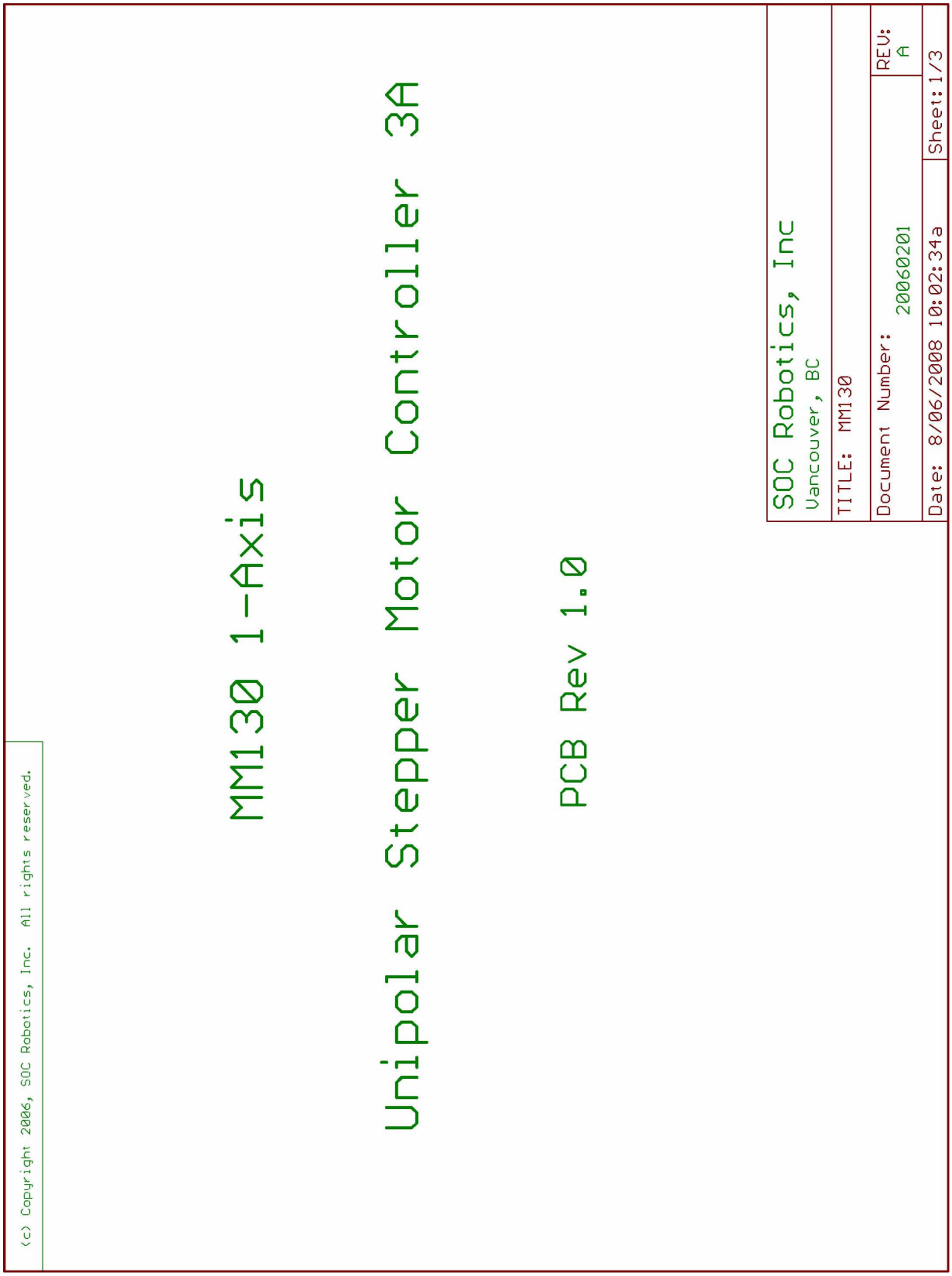
Dimensions: 2.12x1.58 in (one mounting hole)
Weight: 38 grams

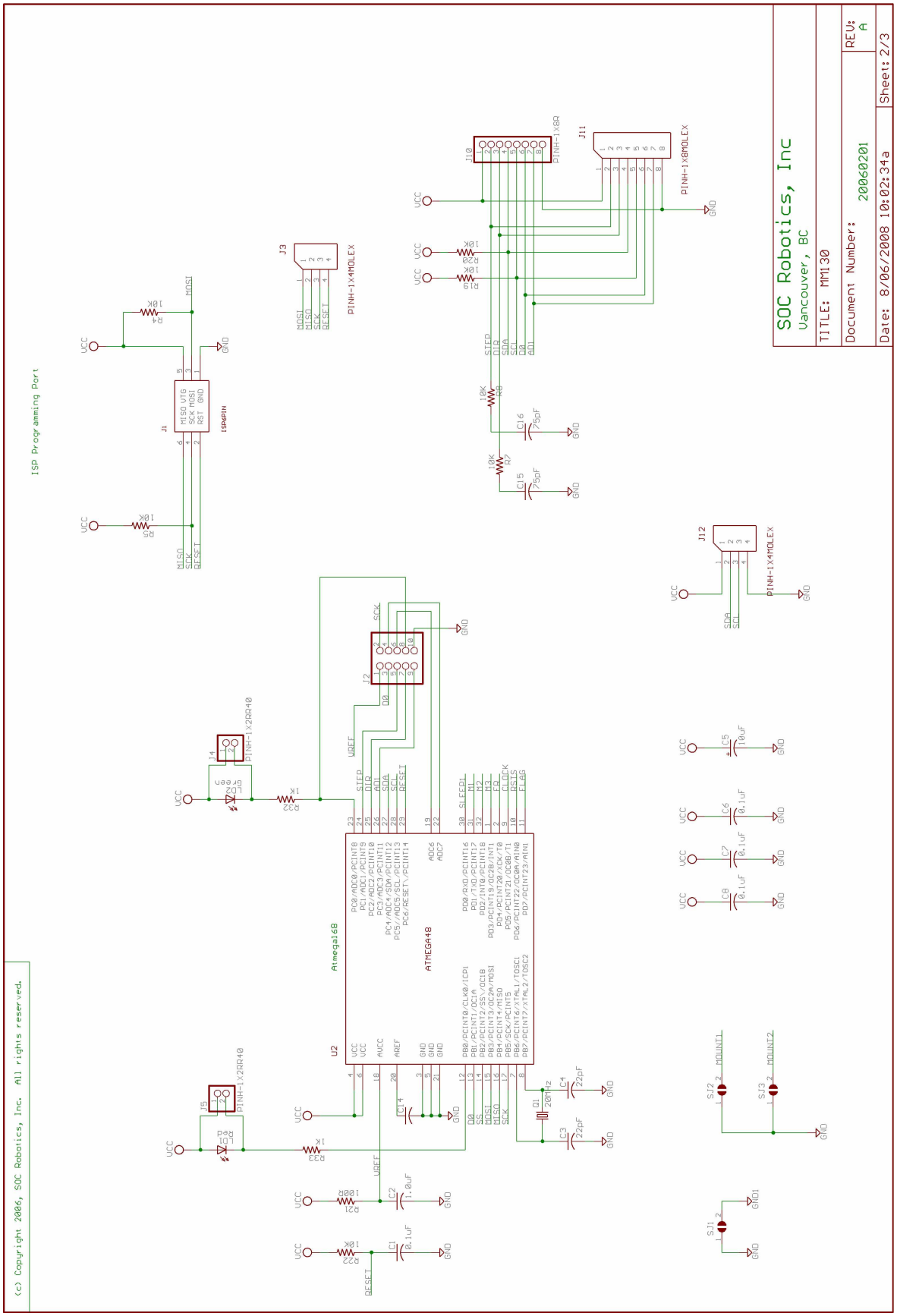
5.3 Mechanical Dimensions

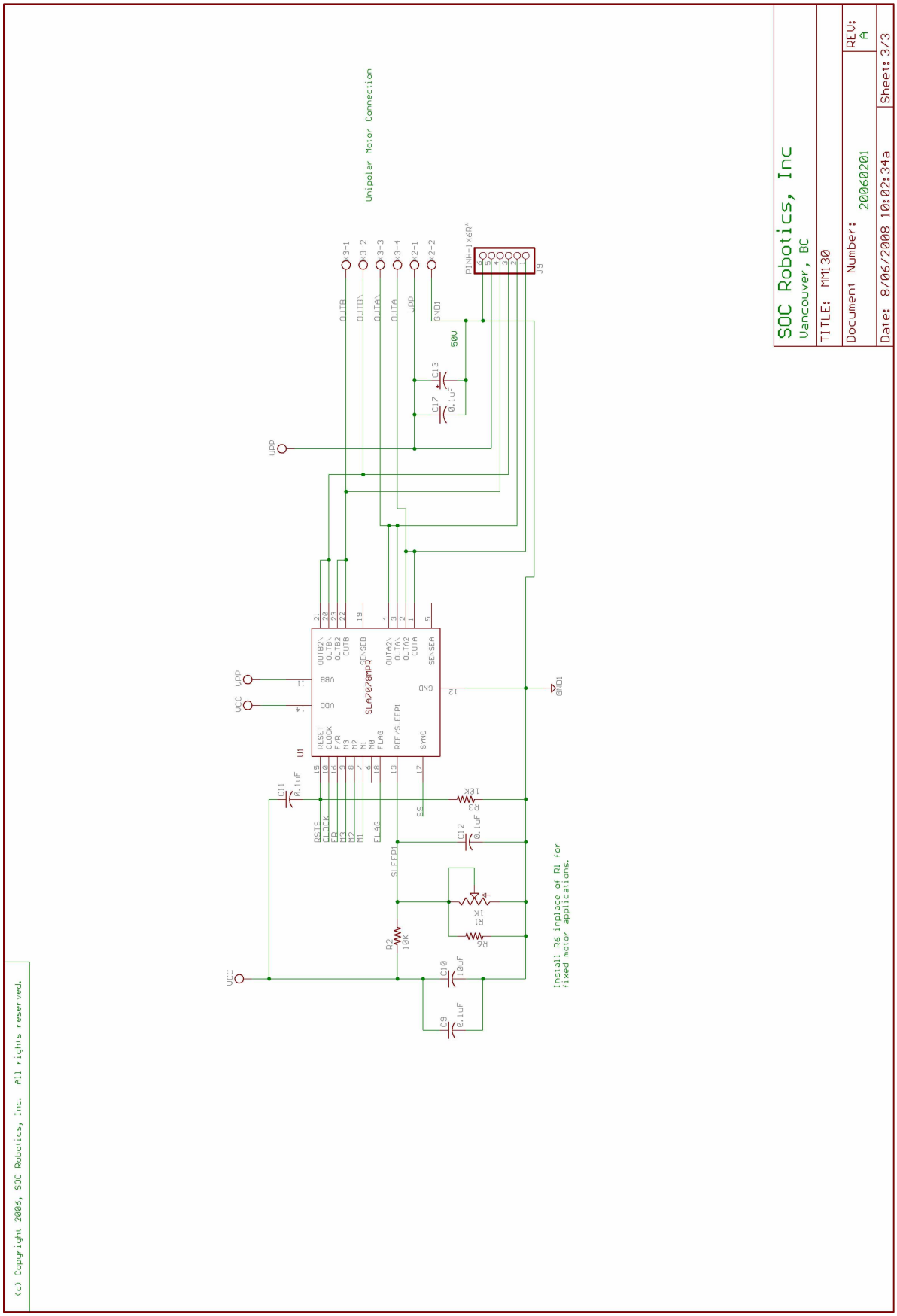
Board dimensions are stated in inches.



6.0 MM130 Circuit Schematics




SOC Robotics, Inc
Vancouver, BC
TITLE: MM130
Document Number: 20060201
REV: A
Date: 8/06/2008 10:02:34a
Sheet: 2/3
ISP Programming Port
ATMEGA48
Atmega168
PINH-1X4MOLEX
PINH-1X8MOLEX
PINH-1X2RA40
(c) Copyright 2006, SOC Robotics, Inc. All rights reserved.


SOC Robotics, Inc
Vancouver, BC
TITLE: MM130
Document Number: 20060201
REV: A
Date: 8/06/2008 10:02:34a
Sheet: 3/3
Unipolar Motor Connection
PINH-1X6R
J9
SLA7078MPR
U1
Install R6 inplace of R1 for fixed motor applications.
(c) Copyright 2006, SOC Robotics, Inc. All rights reserved.

Notes: