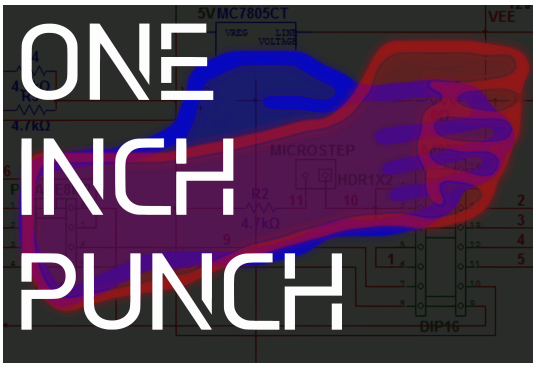
ROBOTS EVERYWHERE

WEBSITE: www.robots-everywhere.com
SUPPORT: support@robots-everywhere.com
OFFICE: +1-415-324-9827



ONE INCH PUNCH

Serial Multiplex Bus Controller

INTRODUCTION

The One Inch Punch (1IP) controller is a versatile integrated controller with a small physical footprint of one square inch. Similar to CANBus, it allows communication between sensors, motors, steppers, and other microcontrollers without the need of a central computer system. The small physical footprint allows it to be integrated into designs ranging from small robotics to large machine installations without concern for weight or miniaturization.

SPECIFICATIONS

Input Voltage: 6V to 24V

Serial Logic Input: 3.3V to +-12V, RS232 (inverted), TTL compatible

Quiescent Current: 35mA max at 16Mhz

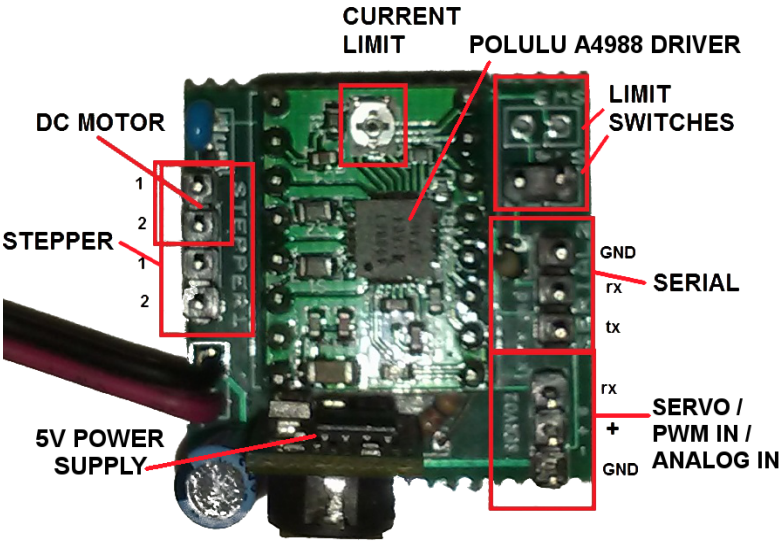
Operating Temperature (°C): -20° to 85°

5V Output Current: 1.5A

Switch Input Impedance: 50kΩ

Serial Input Impedance: 100kΩ

Ports: 1 servo/sensor/analog, 1 motor/stepper, 2 switch, 1 serial/switch



CONNECTING DEVICES

The 1IP controller is able to connect devices to 3 input headers. The motor/stepper control header, the servo control header, and the serial/sensor header.

SERVOS

Connecting servos to the 1ip controller is simple. Ensure the orientation of the servo plug matches the orientation of the header. Connect the servo. The pin towards the corner of the board is ground, as per the above diagram.

STEPPER MOTORS

A single stepper motor may be connected to the 4 pin stepper connector controlled by the a4988 stepper driver. The orientation of the stepper will determine its forward and reverse directions but is functional in either orientation.

Please refer to Polulu website and datasheet for more information on the stepper driver itself.

DC MOTORS, LEDs, OTHER POWERED COMPONENTS

A DC motor may be connected to the 4 pin stepper connector controlled by the a4988 stepper driver. Simply connect the motor to the lower two pins, farthest from the power connector. Connecting to any other pins will require a firmware change.

These pins may also be used to drive **12V** LEDs and other directly powered components. Connect them like a DC motor, and drive them with the same commands.

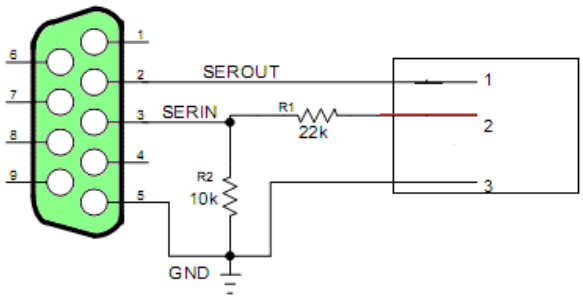
LIMIT SWITCHES

Limit switches can be connected by adding headers to the two pairs of two holes in the bottom corner of the board opposite the servo. See the diagram.

SERIAL CONNECTION

A serial connection can be made using 3 pins. To connect a standard DB9 serial connector, please follow the diagram on the right.

Please ensure the orientation of the serial connector. The pin next to the resistor is ground, as per the connection diagram.



SERIAL CONTROL

To control the 1IP over the serial port, via the default firmware, commands are sent over 3 pin serial. The serial port configuration is as follows:

Baud Rate: 19200

Parity: None

Data Bits: 8

Stop Bits: 1

The command format is as follows:

@@[Address][command][value]

@@ is always two @ characters.

[Address] is the defined address of the 1IP on the 1IP bus. When first flashed, or from the factory, the default address will be AA. This can be changed with the * command. It will always be 2 characters.

[command] is a single character defining the command being sent. Valid commands are !, >, <, {, }, [,], (,), %, $, and * and will be explained below.

[value] is the value passed to the command, such as the value of a servo pulse or number of ms to run a motor or LED.

All commands must be terminated with a not-null character, and **all commands must be sent with a 5ms delay between characters.** If the serial terminal you are using does not support inter character delay, it will not work!

COMMAND REFERENCE

COMMAND	DEFINITION	VALID VALUES	EXAMPLE
*	Change controller address.	XX0YY where XX is the first letter and YY is the second, from 1 (for A) to 26 (for Z).	@@AA*10010 to set address JJ.
%	Sets the stepper motor clutch to stay engaged on move	0 for disengaged, any other value for engaged.	@@AA%0
{	Move DC motor forward, stop if limit switch hit	Number of milliseconds (max 65535) to run motor	@@AA{12345
}	Move DC motor backward, stop if limit switch hit	Number of milliseconds (max 65535) to run motor	@@AA}12345
[	Move DC motor forward, do not stop	Number of milliseconds (max 65535) to run motor	@@AA[12345
]	Move DC motor backward, do not stop	Number of milliseconds (max 65535) to run motor	@@AA]12345
>	Move stepper forward this many steps quickly	Number of steps (max 65535) to run stepper	@@AA>12345
<	Move stepper backward this many steps quickly	Number of steps (max 65535) to run stepper	@@AA<12345
(	Move stepper forward, stop if hit limit switch	Number of steps (max 65535) to run stepper	@@AA(12345
)	Move stepper backward, stop if hit limit switch	Number of steps (max 65535) to run stepper	@@AA)12345
!	Set servo to position	100 to 200 for servo position (min to max)	@@AA!190
$	Set slow and fast stepper speeds.	XX0YY where XX is the slow speed and YY is the fast speed. Values from 0 to 99.	@@AA$05099

TROUBLESHOOTING / FAQ

Issue: Motor or Stepper Stalls or Fails to Start.

Root Cause: Undercurrent on Motor or Stepper

Solution: Turn the current limiting switch clockwise, re-test until motor runs as intended.

Issue: Servo Stalls when using Large Servos

Root Cause: Undercurrent on Servo

Solution: External power supply for servo. The One Inch Punch controller has a limited power supply and may encounter issues with large servos.

Issue: Motor or Stepper Runs Backwards

Root Cause: Connected in Reverse

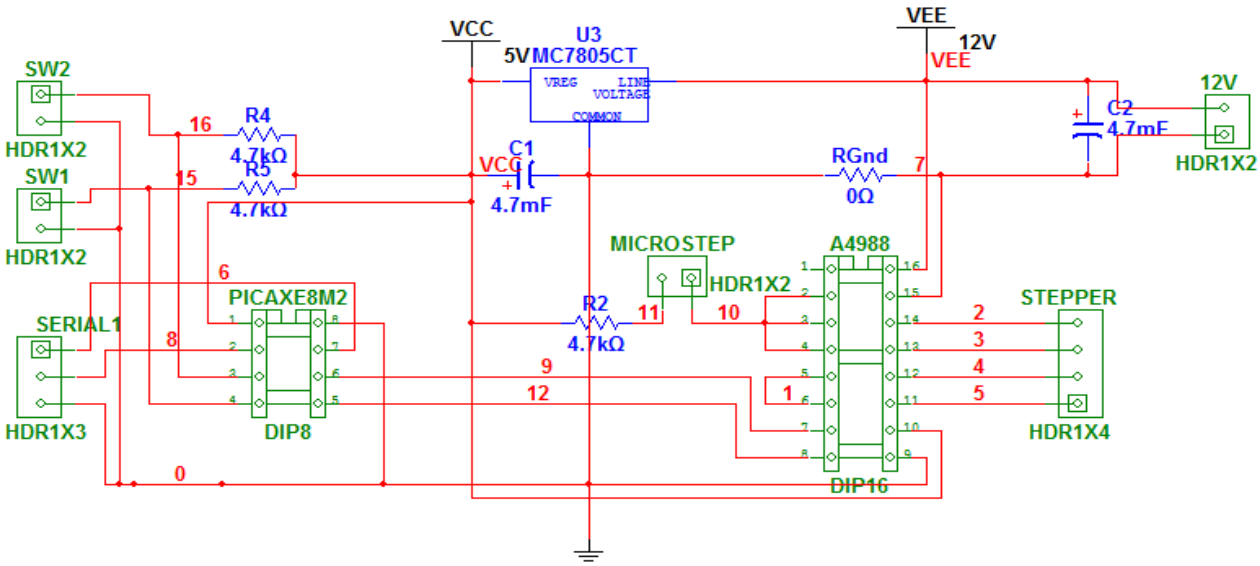
Solution: Reverse connection orientation of motor or stepper.

Issue: Motor or Stepper Does Not Run

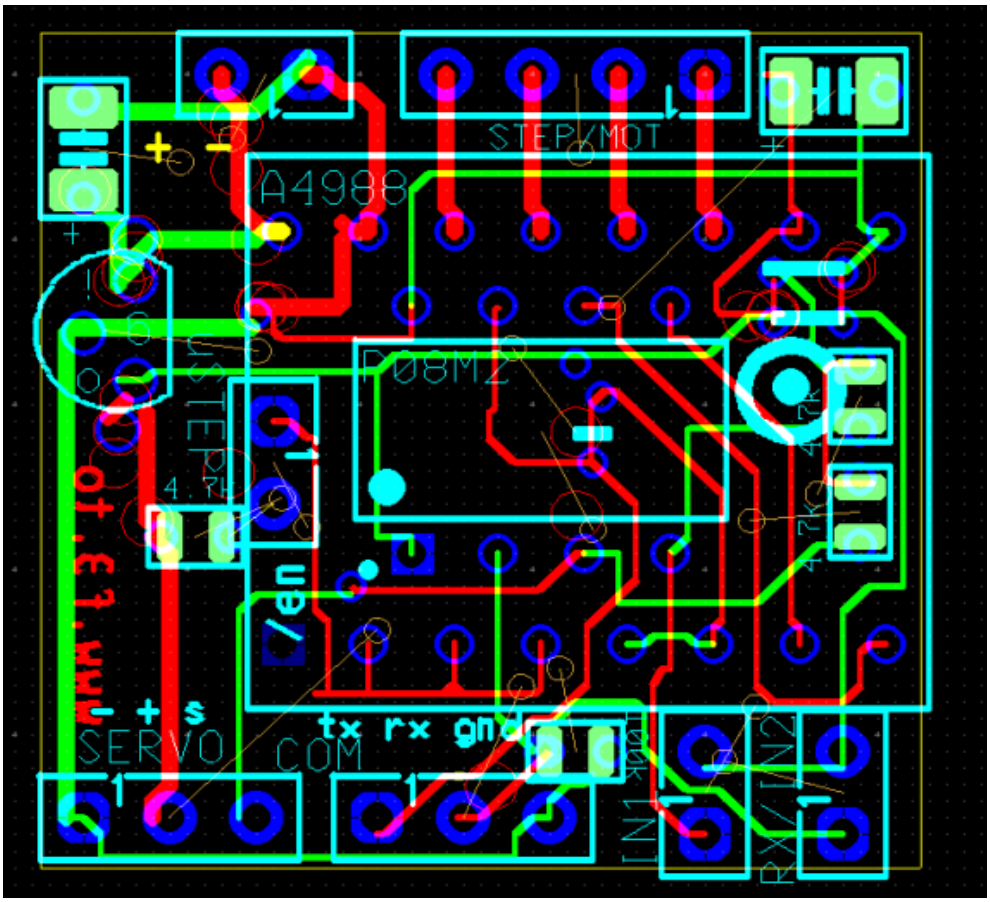
Root Cause: A4988 Backwards

Solution: Check orientation of A4988 Stepper Driver in socket. See photograph and connection diagram. Current limit screw should be near edge of board, and when socketing, lined up with the (+) marker on the PCB underneath the A4988.

SCHEMATIC



LAYOUT REFERENCE



OPEN SOURCE LICENSE

WARRANTY

This product is covered by a **lifetime replacement warranty.** We stand behind our products, which are made in the United States of America.

This warranty covers only express defects of manufacture, and wear and tear until the planned obsolescence of the product as deemed suitable by the manufacturer.

This warranty **does not** cover intentional or accidental damage caused by use or misuse, including overvolting, or destruction in the honorable arena of combat robotics.