

MicroTech

MT Motor Controller

Version 1.30

www.mcu.hk

Warning:

Incorrect power connection to any electronic and electrical equipment may seriously damage them or even cause a fire hazard or explosion. Users must take care to identify the correct pins and supply an acceptable voltage to operate them safely.

Attention:

The Motor Controller does NOT has a voltage regulator built into it, user has to supply a proper 5V DC to power the control circuit!

3rd May 2007

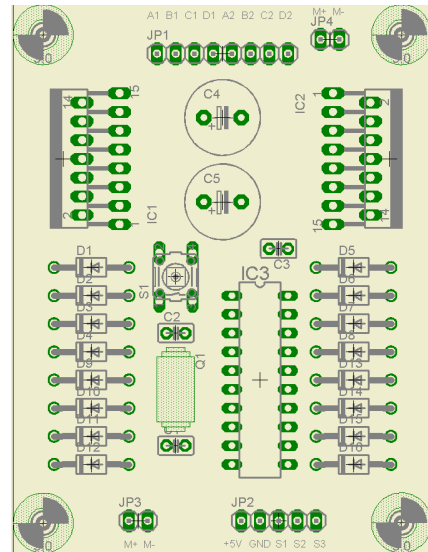
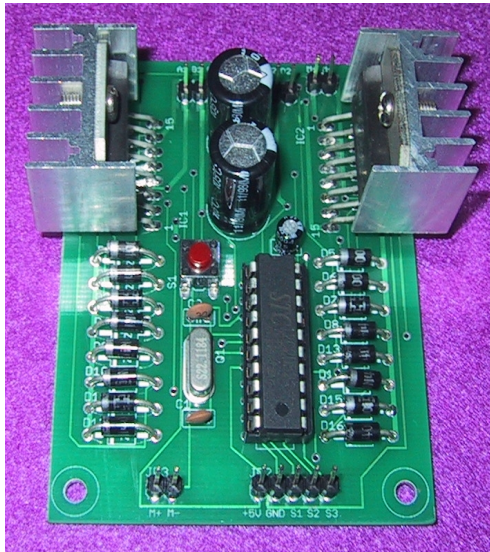
Introduction

The MT Motor Controller is capable of controlling up to **four** DC motors or **two** stepper motors with single stepping pace. The motor controller uses a 3-wire protocol developed by MicroTech to communicate with the controlling circuitry. The onboard MCU is an enhanced version of 2051-compatible microcontroller, which interprets the incoming commands to control the running speed and direction of the attached motors. A document describing the MT protocol can be downloaded from the website.

The motors are controlled and driven by a pair of L298N motor control ICs. A datasheet for this chip can be found on the Internet.

The primary design of this controller board is for robotic applications, but can be easily adapted for other motor control applications as well. Motors work in pairs. Therefore, if two stepper motors are used, they are separately connected to the outputs of the two L298N ICs. However, if only two sets of DC motors are being used then they both have to be connected to the same L298N to work as a pair.

Pinouts and Board Layout



Pin Description

Pin	Description
S1-3	Connect to MT protocol pins
A1-B1	1 st DC motor, 1 st coil of 1 st bipolar motor, 1 st & 2 nd coils of 1 st unipolar motor
C1-D1	2 nd DC motor, 2 nd coil of 1 st bipolar motor, 3 rd & 4 th coils of 1 st unipolar motor
A2-B2	3 rd DC motor, 1 st coil of 2 nd bipolar motor, 1 st & 2 nd coils of 2 nd unipolar motor
C2-D2	4 th DC motor, 2 nd coil of 2 nd bipolar motor, 3 rd & 4 th coils of 2 nd unipolar motor
M-	Ground
M+	10-35V power supply to the motors
GND	Ground
+5V	5V power supply to the board controller IC

Control Command format

The function, **MotorCmd()**, written in TinyC (in "MT\App\mctrl.prg"), illustrates the protocol and command parameters required to communicate with the motor controller.

MotorCmd() takes the following format:

MotorCmd(*Command type, Controller/ On/Off of StepMode, Motor/Number Of Step, Action type/Value*)

The meanings of the last 3 parameters are determined by the command type. Please refer to the example program for a demonstration of how to set the appropriate parameter values for different motor types and control actions.

The following table provide a summary of the various **MotorCmd()** parameters.

Parameter	Description
TYPE	Motor type command which sets the type of the 2 paired motors
SPEED	Speed type command which sets the 2 paired motors running speeds
DIR	Spinning direction type command which sets the running direction of a motor
SWITCH	Switch type command which turns ON/OFF for a motor
STEP	Switch Stepping Mode ON/OFF for both stepper motors
CTRL0	Specify the 1 st L298N motor controller
CTRL1	Specify the 2 nd L298N motor controller
DCM	Specify motors are DC motors
UNI	Specify motors are unipolar stepper motors
BIP	Specify motors are bipolar stepper motors
POS	Specify motors are common-positive (VCC) for unipolar stepper motors
NEG	Specify motors are common-negative (Ground) for unipolar stepper motors
MOTOR0	Specify the 1 st motor on the L298N motor controller
MOTOR1	Specify the 2 nd motor on the L298N motor controller
CW	Sets the spinning direction of motor = clockwise
ATCW	Sets the spinning direction of motor = anti-clockwise
ON	Switch motor ON
OFF	Switch motor OFF

Note:

When the motor controller and the control circuitry are both switched on, it is important to allow a delay of **ONE second** for the motor controller to initialize itself properly before sending commands to it. Also, appropriate delay timings between commands are necessary for the execution of a command and running duration of motors.

When the controller is configured into **Stepping Mode**, user has to repeatedly issue stepping commands in order to move the motors. The moving direction and speed have to be previously set before issuing the commands. Since the **MotorCmd()** function sends 4 bytes to the motor controller each time, the total number of steps to move is split into 2 bytes halves. This can easily be done by using the TinyC's '>>' operator.

Format:

MotorCmd(STEP, ON, steps high-order byte, steps low-order byte);

e.g.

```
MotorCmd(STEP, ON, 0, 1);           // switch Stepping Mode ON, single step
MotorCmd(STEP, ON, 1000>>8, 1000); // switch Stepping Mode ON, 1000 steps
MotorCmd(STEP, OFF, 0, 0);          // switch Stepping Mode Off
```

Schematics

