



# ST-Stepper Motor Control Board User's Guide

## Table of Contents

### Chapter 1: Introduction

Features	1
Quick Start	2

### Chapter 2: Function Description

Stepper Motor Control	3
Stepper Motor Control IC	3
The Driver IC	3
Communication with PCs7	3
1TC232-A	4
Networking with an 110 Board	4
Other Features	4
A/D Converter	4
IRQ, PWM and 110 Ports	4
Software	4

### Chapter 3: Operation of the ST Board

Connections	5
Connections of Stepper Motors	5
Power Supply for the Boar&	5
Power Supply for the Stepper Motors	5
Communications with PCs	5
Controlling Stepper Motors	5
Initializing the Board-	5
Sending Commands to the CY500 ICs	6
Moving the Stepper Motors	7
Stopping the Stepper Motors	7
Triggering the Stepper Motors	<b>a</b>
Home Buttons	8
A/D Conversion	8
Reset	8
Initialization, Getting Conversion Results	S

### Chapter 4: Programming the ST Using the Provided DLL

Using the DLL withr Different Programming Languages	&
Controlling the Motors	9
Establishing the Connection Between a PC and ST	9
Initializing the ST	9
Configuring the Motors	9
Turning and Stopping the Motors	9
Motor Homing	9
Motor Command	10
Error Message	10
A/D Conversion	10
Network Funetions	10

## Appendix A: DLL Functions

# ST-Stepper Motor Control Board User's Guide

**DISCLAIMER:** *RMV ELECTRONICS INC. does not assume any liability arising from the application and/or use of the product/s described herein, nor does it convey any license. RMV ELECTRONICS INC. products are not authorized for use as components in medical, life support or military devices without written permission from RMV ELECTRONICS INC.*

*The material enclosed in this package may not be copied, reproduced or imitated in any way, shape or form without the written consent of RMV-ELECTRONICS INC. This limitation also applies to the firmware that the Integrated Circuits in this package might contain.*

**WARRANTY:** *RMV ELECTRONICS INC. will replace, free of charge, faulty components in this package with the exception of the Integrated Circuits it might contain, for a period of 6 months after the date of purchase*

## Chapter 1 INTRODUCTION

The ST Stepper Motor Control Board Series is designed for multi stepper motor control through a PC or a terminal. The boards come with drivers for 2 (ST2), 3 (ST3) or 4 (ST4) simultaneous steppers. Acceleration and deceleration modes are supported. Hardware stop pins are available for left and right limits or Home buttons. The boards also provide analog and digital inputs, interrupt inputs and a PWM signal output. Interface to-a PC can be done via RS232 or- via RS485 used in conjunction with the RMV 110-485 board. The latter can be configured in an up to 32 board network and-therefore up to-1 28 steppers can -be sequentially or simultaneously controlled from one PC serial port. An RMV RS232/RS485 converter makes the network transparent to an ordinary PC serial-port.

Boards come with a DLL to allow easy interfacing to Windows~ programming languages and software example.

This manual assume-s that you have already read the 1TC232-A manual. It is also recommended. that you read the 10-485 manual should you use the ST boards in a network.

## Features

1. Control of up to 4 stepper motors~ independently.
2. Motor driving power up to 2 Amp. 50V.
3. Acceleration and deceleration modes are supported.
4. Communication with PCs through RS-232 from 300 to 1152(X) Bands.
5. Multiple boards in a network may be controlled through RMV 110-485 board
6. 8 TTL compatible I/O lines.
7. 10 channel AD Converter with 8, 10 or 12 bits of resolution.
8. One 10 .10000 Hz. 0 .100 % duty cycle Pulse Width Modulation channel
9. High and Low interrupt lines.
10. Driver software (16 and 32 bit DLL) for Visual C (Microsoft or Borland), Delphi and Visual Basic is available.

The following is a brief description of how to set up the board. Please refer to the connections diagram of the stepper control board in the Appendix for the connection sites.

- Step 1. Connect the provided serial cable with 9 pin male and 25 pin female connectors, to the DB9 connector on the board and the serial port of your PC.
- Step 2. Plug a board power supply (6.5 to 12V, 500mA recommended) into the powerjack (J 1) on the board. The red LED will light if there power to the board.
- Step 3. Insert a second power transformer into the motor power lines of the PM I screw terminal block. The rating of this transformer depends on the exact specifications for the stepper motors to be driven. Please refer to your stepper motor manual.
- Step 4. If more than one stepper motor is to be controlled, jumper wires must be connected to run power to each driver which requires power.
  - a. For motors which are less than 1A: If two stepper motors are to be controlled, a jumper wire is run from the positive terminal of PM 1 to the positive terminal of PM2. If a third stepper motor is required, an additional jumper wire is run from the positive terminal of PM2 to the positive terminal of PM3. To control a fourth stepper motor a third jumper wire is run from the positive terminal of PM3 to the positive terminal of PM4. Jumper wires do not have to be run between the negative terminals of the PMx terminals which are driving a stepper motor as they share a common ground on the board. The negative terminal only needs to be connected between the motor power supply and one of the PMx terminals being used.
  - b. For stepper motors over 1A: If your system uses steppers motors greater than 1 A. it is recommended you connect the power to the motors in the following configuration: For each stepper motor you wish to control, a wire is run from the positive terminal of each PMx terminal which is driving a stepper motor to a common point. Twist the free ends of these wires together and connect this to the positive terminal of the power supply. The same must be done for the negative terminals of each PMx terminal which- is driving a stepper motor. Naturally, the common point of the wires leading from the negative terminals is attached to the negative terminal of the power supply.
  - c. For stepper motors of differing power requirements: It is recommended that you connect the positive terminal of each PMx terminal which is driving stepper motor(s) with the same power requirements (i.e. same voltage) to the positive terminal of the suitable power supply as described in 'b' above. The negative terminals are also connected to the negative terminal of the same power supply.

The board is now fully connected, powered -and read-to run.

Insert the provided Stepper Motor Controller Board diskette and transfer the seW-extracting zip file ('Install.exe') onto your hard drive. Double click on the Install file to unzip the file. Double click on the 'Demo.exe' file to run the demonstration program. The default settings of Coin Port 2 and 9600 baud should work in most situations.

For a demonstration of the independent control of stepper motors, go to the 'Control' menu and select 'Motor Control Panel. Information on the different control functions may be found in the 'Help' menu.

## Chapter 2 FUNCTION DESCRIPTION

### Stepper Motor Control

#### Stepper Motor Control IC

The heart of the stepper motor control is the CY500 IC. This IC is a standard-5-Volt, 40 pin LSI device configured to control any 4-phase stepper motor. It will interface to any computer using asynchronous parallel TTL input-and-provides numerous TTL inputs and outputs for auxiliary control and interface.

The DBO-6 pins are connected to Port B of the 11C232-A chip for accepting commands. The DB7 pin is ground so that the CY500 works in ASCII mode. The CONTROL pin enables the L298-driver. The ABORT, TRIGGER and RESET pins-are controlled by the 11C232-A chip, which starts and stops the motors.

The ST4 board consists of four CY500 ICs (U8, U9~ U10 and U11). They are operated independently. Thus, the ST board can control a total of 4 stepper motors. To address the four CY500 ICs, two pins (P01, P02) of Port C of the 11C232-A are used, and are decoded by a 74LS139. The RESET, WR, and ABORT control pulse are latched in each 74LS139 when it is selected by a 74LS139. The two address pins (PC0~ PC1) are also connected to the selection pins of the 74LS139, which read back the BUSY/RDY pins of the CY500 ICs and read back the HOME BUTTON status.

The TRIGGER pins of CY500ICs are connected to the 74LS373 which latches the data from PB0~PB3 on the 11C232-A chip. When the TRIGGER pin is high, the motor will not turn, and when it is low the motor will turn.

The CY500 ICs are reset at power up, and can also be reset individually by a reset- signal through PC2 of the 11C232-A.

To stop a motor or to set the home position of a motor, a six-pin header strip and a circuit are used for each CY500 IC. The function is achieved by controlling the ABORT, and EXT START/STOP pins. The abort -signal comes from two sources. One is from PC4 of the 11C232-A. The other is from the header strip (JH1, JH2, JH3, or JH4) by connecting a push button or external logic signals.

#### The Driver IC - L298

All the stepper motors are driven by an L298 which is a high voltage, high current, dual-full-bridge driver. It can handle up to a 2 Amp current. The power supply for motors can be up to 50V. For more powerful stepper motors, a simple H bridge driver circuit can be added as shown in the figures at the end of this manual.

#### Communication with PCs

The ST can be controlled directly via a standard serial port or attached to our I/O-485 stations. The latter configuration allows the user to control up to 32 ST boards from the same serial port, with synchronic triggering of all motors- if required.

## **1TC232-A**

An 1TC232-A is used as the interface between the PC and the CY500 chips. Only Port B and Port C, and the serial port are used for this purpose. The communication between the ST board and the PC is implemented by connecting the RS232 port of PCs and the serial port of the board driven by the MAX232 IC. The baud rates supported are from 300 to 115200.

PBO.6 on the 1TC232-A is used for sending commands to CY500 in ASCII mode when PC7 is low. PBO.3 are also used to trigger the CY500 when PC7 is high. PB7 is used for reading the Home Flag which will be set to I when the moving device touches the Home Button.

PC0 and PC1 are used to address the four CY500 ICs. The combinations of PC0 and PC1 decode the four CY500 IC's addresses. PC2 resets the CY500 IC addressed by PC0 and PC1. PC3 sends a WR signal to the CY500 addressed by PC0 and PC1. PC4 is used for sending a stop signal to the CY500 addressed by PC0 and PC1. The status of the BUSY/RDY pins of the CY500 addressed by PC0 and PC1 is read back through PC5. PC6 resets the A/D converter. When PC7 is low, the 74LS139 is enabled and the address set by PC0 and PC1 is asserted; while the output of 74LS373 are not affected by its input. When PC7 is high, the 74LS139 are disabled, and the address is not asserted, while the trigger signals from PBO.3-PB.3 come through the 74LS373 to each CY500.

## **Networking with an 110485-Board**

The ST board can be connected to the 110-485 board so that it can be connected with an RS485 network. To do so, remove the 1TC232-A chip on the ST board, and connect the 40 pin header strip (JP6) on the ST to the 40 pin header strip (CON1) on the 110-485 board. The ST board is controlled by the UI (1TC232-A chip) on the 110-485 board. To operate the 110-485 board, please refer to the 110-485 board manual.

## **Other Features**

### **A/D Converter**

The ST board supports 10 analog to-digital conversion channels. The resolution of the conversion depends on the ADC used (MCI45041, 8 bits; MCI45051, 10 bits; TLC2543, 12 bits.) They work in a similar manner. The SPI port on the 1TC232-A is used to control the ADC. For more detail on using the ADC, please refer to the 1TC232-A manual.

### **IRQ, PWM and I/O Ports**

The ST board supports two Interrupt inputs (IRQL and IRQH), a Pulse Width Modulation (PWM) output, and 8 I/O compatible pins (Port A of the 1TC232-A). All these resources belong to the 1TC232-A.

## **Software**

A dynamic link library (DLL) and a demonstration program are provided for the ST board. The library provides all the necessary subroutines for controlling this board. The program can be run under Windows 3.1, Windows 95, or Windows NT, and can be called from Delphi, Visual Basic and C++.

The demonstration program provides examples: on controlling the board with the library; demonstrating the effects of all subroutines; and also provides a control panel to facilitate board control.

## Chapter 3 OPERATION OF THE ST BOARD

### Connections

#### Connections of Stepper Motors

There are four connectors (JIM 1, JM3, JM5, and JM7) on the S-T4 board, i.e. four 4-phase stepper motors can be attached to and controlled by the board. The stepper motors connected to JIM 1 are controlled by the first CY501 IC (U8); the one connected to JM3 is controlled by U9; the one connected to JM5 is controlled by U10; the one connected to JM7 is controlled by U11.

#### Power-Supply for the Board

A 7805 (+5 V) regulator provides the power to the board. A 6.5-9V, 500mA adapter is recommended. The power supply should be connected to the Stereo-jack (J 1) or the connector (PWRI) on the ST board.

#### Power Supply for the Stepper Motors

The power supplies for the stepper motors depend on the stepper motors used. Connector PM 1 is for the stepper motor connected to connector JM1. PM2 is for JM3. PM3 is for JM5. PM4 is for JM7. All the power supplies have the same ground.

### Communication with PCs

The DB-9 connector (P1) on the ST-board should be linked to the serial port of a computer or terminal. Communication settings for the ITC232-A and for the computer should be the same. These settings include Baud rate, data bits, stop bits, parity, and flow control. The power up configuration of the ST board is 9600-Baud rate, 8 data bits, 1 stop bit, no parity and no flow control. The Baud rate can be changed via software to any standard value between 300 and 115200 Baud.

The jumper JP 1 and jumper JP2 allow swapping of the Tx and Rx lines to accommodate both straight and "null modem" cables.

### Controlling Stepper Motors

Read this chapter carefully if you are planning on developing code in DOS or for the MAC to access the ST directly. If you are working in Windows, use the DLL provided (see Chapter 4).

#### Initializing—the Board

- a. Configuring ITC232-A ports, unlock honre buttons

Port B of the ITC232-A should be configured as

pbc127 (0411 11-11),

i.e. the pins of PBU. 6 are configured as outputs and BP7 is configured as an input. Port C should be configured as

pcc223 (1101 11-11),

i.e. PC0, 4 and PC6, 7 are configured as outputs and PC5 is configured as an input.

This is the most important part of the initialization process. If the ITC232-A-ports are not properly

configured, the ST board would not work.

5

b. Reset the CY500 ICs, unlock home buttons

When the ST board is powered up. all the chips on the board will be reset, all the BUSY/READY pins of the CY500 ICs will be high, (i.e. in READY status), and the home button status read from the ZB pin (the second output pin) of the 74LS153 will be high which means the motor is not in the home position. To make sure the configuration is correct, we may need-to-configure this by sending the following commands through 1TC232-A:

```
pwb15 (0000 1141)
pwc1S6 (1001 1100)--;          this-will -set all the TRIGGER pins-to-high, so that the
                                motors will not move.

pwc8 (0000 1000)
pwc9 (0000 1001)
pwc10 (0000 1010)
pwc1 (0000 1011);             these will lower all-the RESET and ABORT pins.
[Delay about IO0ms];         this requires software be written by the user to
                                pertbrm this delay.

pwc24 (0001 1100)
pwc25 (0001 11-01)
pwc26 (0001 1110)
pwc27 (0001-1111)
```

As explained in section 2.1.1, the two- least significant-bits-of-the Port C of the 1TC232-A- are used for addressing the four CY500 ICs.

When one of the home-buttons is set at home position. the ABORT pin-of-the corresponding CY500 will be kept low until the PC4 of the 1TC232-A is low and the address of the CY500 is set

### **Sending Commands to the CY500 ICs**

Because the CY500 ICs are working-in ASCII mode, all-the- CY500 conunnnds sent through Port B of the 11(2232-A should be in ASCII string. For example. to set 100 steps. the command string is “N I0<return>” containing 5 bytes data, where <return> means the ASCII code 13’. To accept one data byte, a low strobe from the WR pin of the CY500 is needed. Thus, to send this string to- one of the four- CY504) ICs, the following code should be sent out-to--the 11(2232—A- (It is assumed in the example that the string is sent to the second CY500 (U9)).

```
pwcb000III0I ;          settheaddr-ess
prb              read Port B
if (PB7 is high) then (continue); When PB7- is high, the CY500 is ready.
Pwb78           ;          ‘78’ is the ASCII code of the letter N.
pwcb000IOIOI  ;          set the PC3-to low, i.e. set the WR pin low.
prb              read Port B
if (PB7 is lo~v) then (continue) else (wait);
pwcb000III0I  ;          settheWRpinhigh.
prb
if (PB7 is high) then (continue) else (wait);
pwb20           ;          ‘20’ is the ASCII code of the space.
pwcb000IOIOI  ;          set the PC3 to low, i.e. set the WR pin low.
prb              ;          read Port B
if (PB7 is low) then (continue) else (wait);
pwcb000III0I ;          settheWRpinhigh. prb
```

```
if (PB7 is high) then(continue)-else (-wait);
```

6

```
pwb31 ; '31' is the ASCII-code of 1.
pwc000 10101 ; set the PC3 to low, i.e. set the WR pin low.
prb ; read-Port B
if (PB7 is low) then (continue) else (wait);
pwc0001111ftl- ; set the WR pin-high. prb
if (PB7 is- high) then (continue) else (wait);
pwb30 ; '30' is the ASCII code of 0.
pwc00010i0-1- ; set the PC3-to-iow, i.e. set-The WR pin low.
prb ; read Port B
if (PB7 is low)-- then- (continue) else (-wait); pwc0001 1101 ; set the WR pin high. prb
if (PB7 is high) then (continue) else (wait);
pwb13 ; '13' is-the ASCII-code of carriage return..
pwc000 10101 ; set the PC3 to low, i.e. set the WR pin low.
prb ; read Port R
if (PB7 is low) then (continue) else (wait).
pwc0001110i- ; set the WR pin-high.
```

## Moving the Stepper Motors

To move the motor, several-steps should be followed.

- enable the L298 (or power up the L298) by sending command “B” to the CY500.
- set the TRIGGER pin of the-CY500 to low, so that the CY500 is triggered: The code for-doing this, is as follow:

```
pwb0
pwc1001 1100
pwc0001 1100
```
- send CY500 commands like “N xx” “R xx”. “F xx”, “4”, “~S xx” to configure the speed. Number of steps, direction, acceleration, where “xx” means any number.
- send CY500 command of “G”. to start moving the stepper motor.

The sequence of the above steps is not important. Step b can be the last step. i.e. when the motor is ready to go, set the trigger.

## Stopping the Stepper Motors

There are several ways to stop a moving stepper motor:

- Auto stop: When the motor finishes the number of steps set by the command “N xx”. the motor will stop automatically.
- Abort: When the ABORT pin of The CY500 goes low, the motor will be stopped. There are two ways to set the ABORT bit low, one is by software by setting PC4 pin of the 11(2232-A) to low. Another way is by pressing-home button.- When the home button is set, the ABORT pin will-be set low and kept low until the PC4 pin is set low.
- Un-trigger: When the TRIGGER-pin of the CY500'-is high, the moving motor will be stopped.. however when the pin is low, motion will be resumed
- “Q” command: When the CY500 command “J” is used, the “Q” command will stop the motor.

## Triggering the Stepper Motors

As explained before-. to-send the trigger-signal-to the CY500s,-PC7 should be high, and the 4 least-significant pins of Port B are used as triggers. PBO is for the first CY500 (U8). PB I is for the second (U9), PB2 is for U10--and PB3 is-for U1 1. When PBO is low, the CY500 (U~ is

7

triggered ((the same for PB I. PB2 and PB-3)~ To- start two or more motors at the same time, you can un-trigger the motors first, then configure the motors so that they are ready to go and finally trigger them.

## **Home Buttons**

Every CY500 IC has a "Home button" circuit which sets the ABORT pin low (abortion) when~ the motor reaches home position, and a 6 pin header strip (JH1 for U8, 11-12 for U9. 1H3 for U 10 and JH4 for HI I) which connects with-up to-two push-buttons-or two external stop signal& Examples of the connection for the buttons can be found in the figures at the end.

## **AID Cotwersiqn**

### **Reset**

When the board is powered up. the A/D converter will be reset automatically. You can also-resot the ADC by setting PC6 to high for a period of 100 ins.

### **Initialization, Getting Conversion Results**

In the 11(2 232-A manual, the operation of initializing the ADC and getting conversion results from-the ADC is explained in-detail. Please-refer-to the manual for more information regarding this topic.

## **Chapter 4 PROGRAMMING THE ST VIA THE PROVIDED DLL**

The DLL for the ST board is designed to facilitate the work o~f-programuning the ST Beard-from Delphi. Visual Basic, and C++ in Windows.

The functions in this-DLL can be divided into- three major categories. There are port functions with the prefix 'Port' for serial communication. ITC functions with the prefix '11(2' for controlling the 11(2232-A chip and Motor functions-with the prefix--Motor' for step motor controlling.

### **Using the DLL with Different Programming Languages**

The DLL for the ST board is-written in-Delphi. however, it can be used equally well with Visual Basic and C. In the Demo program, the declarations for all the functions required by your program-are listed and they can be cut and-pasted-into-your own code. In Appendix-A-of this manual, all the functions are explained in detail.

## **Controlling the Motors**

## **Establishing the Connection Between a PC and the ST Board**

First, the serial port must be configured and opened. To configure the serial port, you call-either the *'PortConfigure'* or the *'PortConfigureForm'* functions. The difference between these two functions is that the latter provides a dialog form -with the DLL to set the required parameters-> *'PortConfigure'* does not open a configuration form; instead, the COM port number and the Baud rate are passed as function parameters~ These two~ functions-are optional before you-open the