ST400C-NT USER'S GUIDE

Table of Contents

DISCLAIMER: *RMV MOTION 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 MOTION INC. products are not authorized for use as components in medical, life support or military devices without written permission from RMV MOTION INC.*
The material endosed 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 MOTION 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 ST400B-NT Stepper Motor Control Board Series has been designed for multi stepper motor control through a PC or a terminal. Acceleration and deceleration modes are supported. A velocity-profiling feature is also available for complex motion parameters. Inputs for external control such as abort, limit and home allow easy interfacing with mechanical system. Analog and digital inputs give the user a variety of options for sensor reading and actuator control. Networking through the RS232 port gives the system the ability to control up to 16 stepper motors, by daisy-chaining up to 4 boards in a multi-drop configuration. A 32-bit DLL allows easy interfacing to Windows programming languages, and a demo software example is also included.

Features

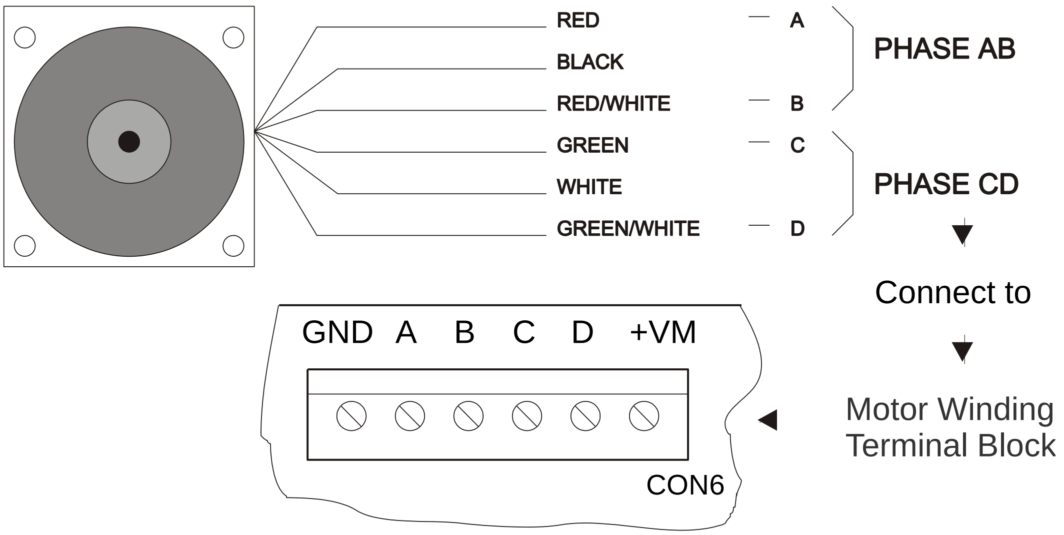
- Control up to 16 stepper motors independently (4 per board).
- Up to 8500 steps/sec,
- Biphasic, monophasic and halfstep modes.
- Current mode driver (chopper) for unipolar or bipolar motors up to 2 Amp / 40 V; operating current is set trough software commands.
- Drives 4,5,6 and 8 wire steppers.
- Automatic Power Saving timer.
- 16-million step position register can be read on the fly.
- First Rate, Slew Rate and Acceleration parameters.
- Velocity profiling mode for complex motion schemes using internal 128 byte FIFO.
- Abort, Home and Limit inputs for external control.
- High speed RS-232 port from 9600 to 115200 Bauds.
- Multiple boards in a network may be controlled through a single PC RS232 port.
- Up to 32 digital I/O lines for general purpose and SPI communication.
- 11 channel AD Converter with 8, 10 or 12 bits of resolution and 2.500 V precision reference.
- 32-bit DLL for Visual C/C++, Delphi and Visual Basic programming.

Quick Start

The following is a brief description of how to set up the board. Please refer to the board overview for a more clear idea of the locations of the parts mentioned herein.

1. Connect the provided serial cable to the board serial port RJ45 connector identified as INPUT (# 3 in the drawing). Using the RJ45 to DB9 female adapter connect the other end of the RJ45 cable, and then to an available serial port of your PC. Use a DB25-male to DB9-male adapter if applicable.
2. Connect a board power supply (7 to 15V, 500mA) to the board power terminal block (# 2) on the board. The board power LED (# 10) will glow if there is power on board.
3. In order to connect the motors, winding terminals need to be identified first, and connected to terminal blocks CON6 to CON9 as shown as follows:

6-WIRE STEPPER MOTOR



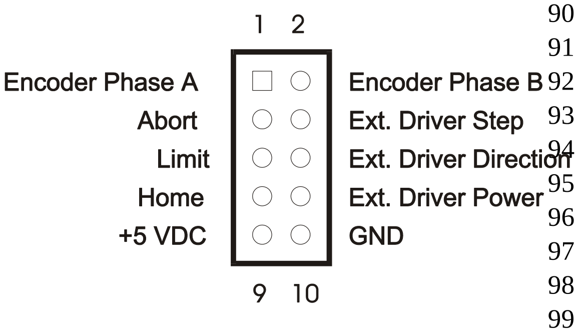
WINDINGS CONNECTION

The picture above shows the standard wire colors for a 6-wire stepper motor. Since this kind of motor is intended for use with unipolar drivers, they have a centered tap connection that is no needed when using a bipolar driver, like the one found on the ST400-NT. If your motor doesn't match the above depicted color configuration, and easy way of identifying phase wires is using an ohmmeter. Look for a pair of wires that have continuity and measure its resistance. You will find 2 sets of three wires, on each of them two wires will have a higher resistance value. Identify those and mark them as A and B (Phase AB). Repeat this operation for the second set of three wires and mark them as C and D (Phase CD). Once phases have been identify, you can connect the wires to the motor winding terminal block (#1 in the board overview) as marked on board: terminals A, B, C and D. The next issue is to provide a power supply for the motor.

4. In order to have a better performance on the current mode driver the voltage used for driving the motor should be at least twice as high the nominal voltage for the windings (you may find this data available somewhere on the motor's body). For instance, a 36 VDC / 16 Amps power supply should be enough for a 4 motor

application with each drawing 2 Amp per coil, and connecting all of them with a common power supply. The operating current can be set later on in software. Immediately after powering on the board, the operating current on the choppers is set to zero. Connect the positive terminal of the power supply to terminal **VM+** on the motor winding terminal block (#1) and the negative to **GND**. Observe polarity carefully.

5. Address setting DIP switches (#7) have been factory configured when the board was tested, and should require no changes. Should you for any reason alter the addressing, the provided DEMO program can guide you on how to set the DIP switches on board for a given address configuration.



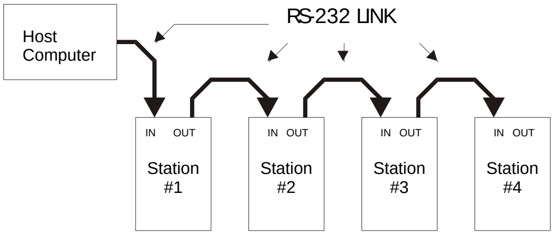
External Control Connector

6. Inputs for home, limit switch and abort are available on the External Control Connectors (# 4). Also the signals necessary for connecting to a shaft encoder, or an external power driver can be found on this connector as shown on the picture. Abort input (when low) causes the motor to stop at once and clears any remaining operations stored on the FIFO memory. Limit (when low) will stop the motion and set a flag according to the moving direction (CW or CCW). Home (when high) will also stop the motion and set a flag, provided that the SEEK_HOME mode has been previously enabled on the MOTORCONFIG register. Reading the MOTORSTATUS register can monitor the state of these three flags.
7. This completes the hardware settings. In order to test the motors, use the provided disk, follow the instructions enclosed on the README.TXT file and install the software as directed.
8. Once the software has been installed, you may find useful to have a look on the help files within the DEMO program. There you will find information concerning board programming and examples.

FUNCTION DESCRIPTION

Host Interface and Communication with PC's

Interfacing is accomplished by using a standard capacitive charge-pump RS232 IC, which generates the voltage sources necessary for driving the RS232 TX DATA signal. Two driver- receiver pairs handle the RS232 interface. The first one handles communication with the host (or remote board) while the remaining one allows multidrop operation.



Networking Operation

Up to four ST400B-NT can be network, making a 16 stepper motor system to be controlled by a single computer. The RS232 data flows from the host computer to the 1st board and from there it “daisy-chains” to the 2nd board. In this way up to 4 board can share the same RS232 line. On every station (an ST400B-NT board) the signal gets repeated and sent to the next station.

Stepper Motor Controller IC

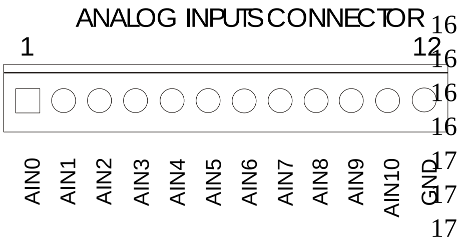
The heart of the stepper motor control is the RMV856 IC. This is a CMOS custom microcontroller that takes care of all the functions necessary to control the stepper motor, digital input-output and analog inputs. It has been designed using a network-oriented concept, which allows easy interaction and programming of several controllers at the same time. An embedded UART on this microcontroller allows asynchronous communication using any standard speed between 9600 and 115200 Bauds. The RMV856 can work together with a 4 phases standard driver (*phases A, B, C and D*), or generate the signals required for and external driver (*step, direction and power control*), depending on the configuration settings.

Power Driver Section

All the stepper motors are driven by an H-bridge IC, the L298. This is a high voltage, high current, dual-full-bridge driver. It can handle up to a 2 Amp current and is controlled to work as current mode chopper. Winding currents can be tightly controlled according to the value set by the on-board Digital to Analog Converter IC (TLC5620). The user has the ability of setting this current to any value between 50 mA and 2 Amps. When half step mode is selected, a torque compensation technique shapes the driving current to follow a pseudo-sinusoidal waveform. Current shaping greatly reduces resonance associated with full step driving, while improving torque characteristic of half step driving. Free wheel diodes are connected to the H-bridge legs, so that a very fast turn off time is achieved, allowing in this way high speed motor stepping.

Winding Current Setting and Power Saving

Operating winding current can be set to any value between 0.05 and 2.00 Amps, by using the corresponding software function on the DLL. When Power Saving is enabled, if the motor is idle for a period longer than 1 second, the winding currents will be set to half the programmed value. No sooner the motor restarts motion than the power control is takeover by the stepping procedure, and the current returns to the initially set value.

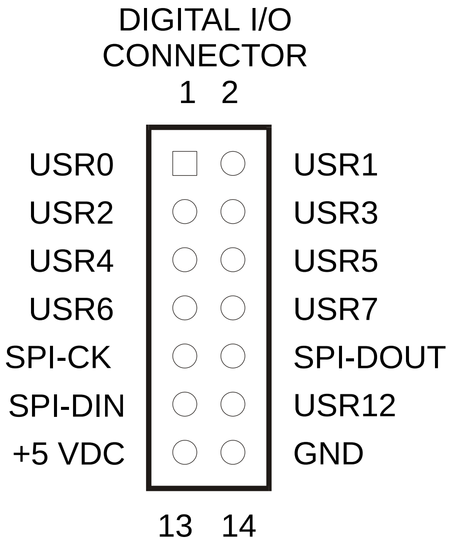


Analog to Digital Converter

Analog to digital conversion is also available through 12 circuits locking header (# 5). The input signal range must be within 0 to 2.5 V DC. Input impedance is 10 Kohms. A precision on board 2.5 V voltage reference guarantees reliable and precise readings. Three kind of ADC can be used with the board: 8, 10 and 12 bit resolution.

Digital I/O Ports

32 TTL compatible Input - output lines are available for control of external devices or reading any kind of switches. Each line can sink or source up to 5 mA. Two high-speed synchronous interfaces are available, for interfacing with serial protocol devices such as memories, DAC's, LCD, etc.



Software Library and Examples

A dynamic link library (DLL) and a demonstration program are provided with the ST board. The library provides all the necessary subroutines for controlling this board. The program can be run under Windows 95 or Windows NT, and can be called from Delphi, Visual Basic and C++. The demonstration program provides examples like how to control the board with the library; demonstrating the effects of all subroutines and also provides four control panels (one per motor) making board control and easy task.

Chapter 3

OPERATION OF THE ST BOARD

Connections

Connections of Stepper Motors

8, 6, 5 or 4 lead stepper motors can be connected to the board using the terminal blocks Motor1 to Motor4 (#1). A separated power supply connection for each motor is provided on each of them. A common ground connection between all motors and the board power supply is arranged so that the ground terminal is the most negative point in all the connections. Unipolar motors are connected in bipolar mode, which leads to a better utilization of the windings and avoids unnecessary heating. This means that the centered taps must not be connected. The best thing to do is to keep them isolated by using a piece of shrinking tube and avoid any short circuit.

Power-Supply for the Board

The board power terminal block (#2) provides a connection for board power supply. Power supply voltage ranges from a minimum of 8 VDC to a maximum of 24 VDC. An inexpensive 12 VDC/500 mA wall transformer is a good option for satisfying that requirement.

Power Supply for the Stepper Motors

This power supply must meet stepper motors requirements in terms of torque, driving method and speed. Depending on whether the motors are going to be operated in constant current mode or not, the voltage must meet the requirement of the chopping circuit. That means that in order to achieve a high stepping rate, the motor's power supply voltage should be high enough for decreasing the turn on time on the windings. A typical voltage for most NEMA23 motors is 36 VDC. This power supply must also be able to provide the peak current at which the motor is rated. For instance, suppose the motor to be driven requires 3.3 V/2 Amps per winding to provide the rated static torque. The equivalent winding resistance is:

Equivalent Resistance = 3.3V/2 A = 1.15 Ohms

increasing the voltage from the nominal value of 3.3 VDC to 36 VDC will make the time constant to decreased about ten times, since the resistance value has been incremented by the same amount. This reduction in the electric time constant will allow the motor to reach a pull in rate very much higher than using a 3.3 VDC power (which is the supply voltage for providing the static torque current). That means that the acceleration rate and pullout torque at high stepping rate will also benefit from this situation.

Shaft Encoder Operation

A quadrature two channels shaft encoder can be connected to the External Control for reading back the shaft position. A 24-bit register is available for this purpose and it can be read at any time. Another use of the shaft encoder is the motor stall detection. When enabled, this feature will stop the motor if the controller detects that there is no position confirmation from the encoder when a step has been taken. If you are using a motor with a shaft encoder ready, be aware that in order to use the stall detection feature the number of steps per revolution on the motor must be equal to the encoder’s pulse count per revolution. Also keep in mind that the encoder phases should be connected in a way that when the motor turns CW the encoder position register is incremented. This can be done by properly connecting the encoder’s wires to Encoder Phase A and B inputs on the External Control Connector for that particular motor. If the result indicates a situation that opposes to the one above mentioned, the encoder’s wires should be switched. You can use the DEMO program in order to read the encoder counter register and verify the operation above described.

Programming the ST400B-NT

The distribution disk included in the box contains a DEMO program written in C++, a 32-Bit DLL and help file covering all the programming issues with examples for each function. Among others, there you will find information about the following topics:

- Serial Port Functions
- Motion Related Commands
- Motion Configuration Related functions
- Controller Addressing
- Digital to Analog
- Digital I/O
- FIFO and Velocity Profiling
- Error Messages