



AQ11\_13525\_232  
User's Guide  
3010002

V 2.2

# AQ11\_13525\_232 RS232 Stepper Motor Controller

## User's Guide

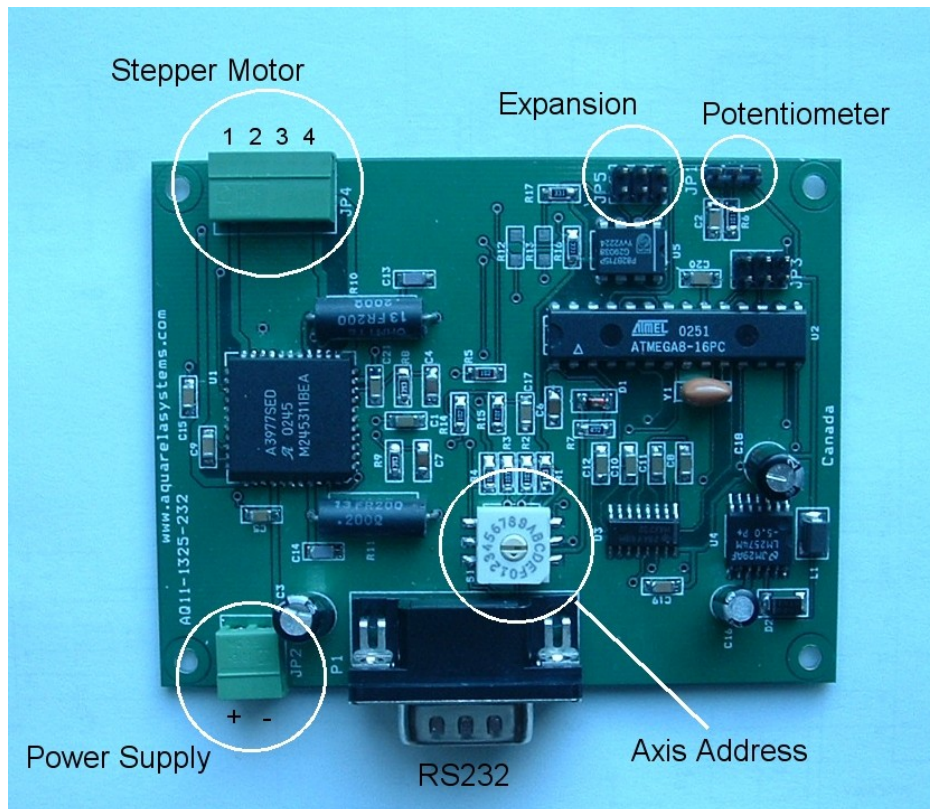
<i>Date</i>	<i>Rev</i>	<i>Change</i>
	1.0	First Release
Apr-19-2008	2.0	Add Undocumented Commands
Apr-20-2008	2.1	Clarification on several sections
Jan-25-2009	2.2	Added information on the break command

**1. Specifications:**

	Min	Typical	Max	Unit
Voltage	8		35	V
Current			2.5	A
Expansion Bus Total Length			50	Meter
Position Resolution	32-bit			
Potentiometer Input	0		5	V
Potentiometer Resolution	8-bit			
Thermal Shutdown Temperature		165		°C
Thermal Shutdown Hysteresis		15		°C
Adjustable Current Limit	0		2.5	A
Step Rate			12500	Step/s

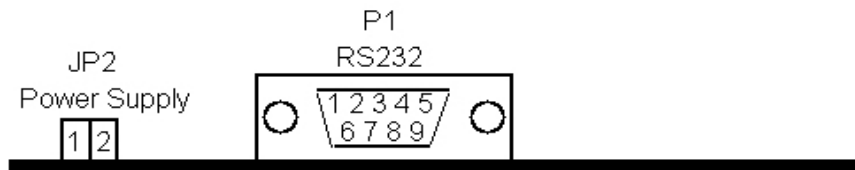
**Table 1: Specifications**

**2. Connections:**



**Figure 1: General View**

- a) Connect the power supply to JP2. You can use 8V to 35V DC. You can locate the power supply in the bottom left corner of figure 1. Note the positive wire must be connected to pin 1 (see figure 2 below)



**Figure 2: Power Supply and RS 232.**

- b) Connect the RS232 port to your computer. You will need a null modem DB9 female to DB9 female cable. The pin configuration is given by table 2:

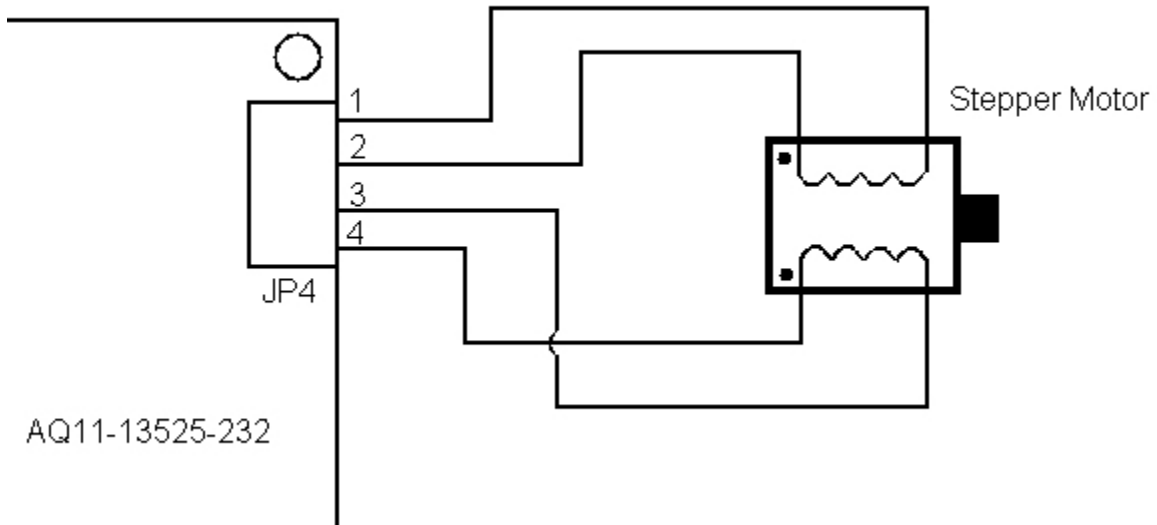
AQ11-13525-232		Computer	
Pin	Function	Pin	Function
2	TX	2	RX
3	RX	3	TX
5	GND	5	GND

**Table 2: RS232 Pin Configuration**

- c) Connect the stepper motor to JP4.



**Figure 3: Stepper Motor, Potentiometer and Expansion Connectors.**



**Figure 4: Motor Connection**

d) Connect potentiometer to JP1 if needed

Pin	Function
1	5V
2	Input
3	GND

### 3. Communications:

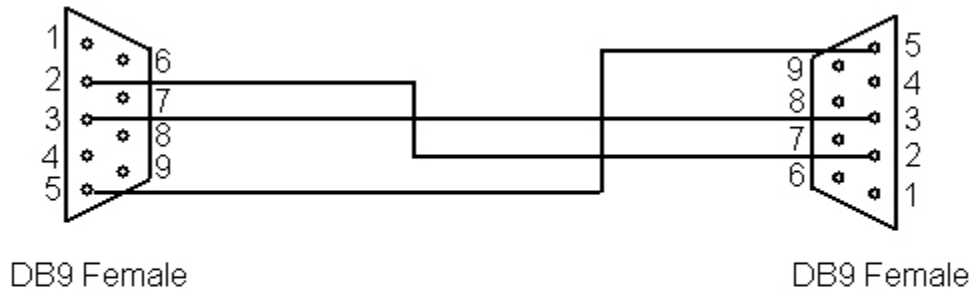
#### 3.1 RS232 Settings:

<b>Baud Rate</b>	9600
<b>Number of Bits</b>	8
<b>Parity</b>	None
<b>Stop Bits</b>	1

**Table 4: RS232 Settings**

#### 3.2 RS232 Cable:

Use a 9-pin female to 9-pin female RS232 cable without crossover. Please refer to Figure 5 for connection diagram.



**Figure 5: RS232 Cable Diagram**

#### 3.3 Commands:

Your AQ11-1325-232 supports two different protocols. One is ASCII based and is ideal for console operation. The other protocol is binary and is more efficient and easier to implement if you have computer control.

After power up the system is set to ASCII mode. To switch between ASCII and binary use the command “binp”.

ASCII Command	Binary Command	Action
stop	0x01	Stop the motor
movt	0x02	Move to position
jogf	0x03	Move forward
jogr	0x04	Move Reverse
	support@aquarelasystems.com	5 of 10

stpr	0x05	Change speed
movv	0x06	Move number of steps
movl	0x07	Move checking potentiometer
liml	0x0b	Set potentiometer lower limit
limu	0x0c	Set potentiometer upper limit
acst	0x0d	Set acceleration ramp first speed
acrt	0x0e	Set rate of acceleration
spos	0x0f	Set current position
rpos	0x10	Read current position
rpot	0x11	Read potentiometer position
smod	0x12	Set step mode
stri	0x13	Set trip current
rtri	0x14	Get trip current
binp	0x15	Change to binary/ASCII protocol
defa	0x16	Reload default values
brea	0x19	Brake
sbrk	0x1a	Set brake current

**Table 5: Commands**

**stop**

- Stop the motor or release brake.
- Argument: no

**movt**

- Move to position <Argument>
- Argument: Target position (positive or negative)

**jogf**

- Move forward
- Argument: none

**jogr**

- Move reverse
- Argument: none

**stpr**

- Set step rate
- Argument: Speed level 1 to 10

**mova**

- Move <Argument> steps
- Argument: Number of steps to move (positive or negative)

**movl**

- Move to position give by <Argument> but checking potentiometer limit
- Argument: Target position

**liml**

- Set potentiometer lower limit ( 0 to 1023) default 10.
- Argument: Potentiometer lower limit

**limu**

- Set potentiometer upper limit(0 to 1023) default 1010.
- Argument: Potentiometer upper limit

**acst**

- Set acceleration ramp first speed
- Argument: Speed level 1 to 10

**acrt**

- Set rate of acceleration
- Argument: Number of 0.1s to reach nominal speed

**spos**

- Set current position to <Argument>
- Argument: New nominal position

**rpos**


- Read Current Position
- Argument: none

**rpot**

- Read Potentiometer value
- Argument: none

**smod**

- Set step mode 1, 1/2, 1/4 or 1/8 step

	AQ11_13525_232 User's Guide 3010002	V 2.2
---	---	-------

- Argument: 1, 2, 4 or 8

**stri**

- Set trip current
- Argument: Percentage of nominal current (2.5A)

**rtri**

- Read trip current setting
- Argument: none

**sbrk**

- Set brake current
- Argument: Percentage of nominal current (2.5A)
- Default value: 10 (10%)
- Range: 0 to 100

**Note:** All movements with implicit stop at the end will end on brake mode. A small adjustable current is maintained to provide holding torque. Brake current increases motor temperature. To release the brake, use the “stop” command.

**Warning:** Under moderate to high brake current the motor will reach high temperatures very fast. Use with caution.

**3.4 Syntax**

The ASCII commands have the following syntax:

[axis] <command> [argument]

The axis is optional and if omitted the command will be executed by the card you are connected to. If axis is included it is 1 character from 0 to f

Examples:

Make card at address '5' jog forward:

AQ11> 5 jogf

Send stop command to the card you have RS232 connection:

```
AQ11> stop
```

Move axis 'a' to position 1200

```
AQ11> a movt 1200
```

### 3.5 Binary protocol

The binary protocol uses fixed size (8 bytes) messages. The card can be switched to binary protocol by typing binp at the command prompt.

The message format is as follows:

<b>Protocol</b>
<b>Address</b>
<b>Command</b>
<b>Value (MSB)</b>
<b>Value</b>
<b>Value</b>
<b>Value(LSB)</b>
<b>Checksum</b>

Protocol is always 0

Address is the address of the card receiving the command and is not optional like in the ASCII protocol.

Addresses are defined in the table below:

<i>S1 position</i>	<i>Address</i>
0	0x20
1	0x22
2	0x24

<i>S1 position</i>	<i>Address</i>
3	0x26
4	0x28
5	0x2a
6	0x2c
7	0x2e
8	0x30
9	0x32
a	0x34
b	0x36
c	0x38
d	0x3a
e	0x3c
f	0x3e

Command is as defined in Table 5

Value is a 32-bit (4 bytes) value MSB first

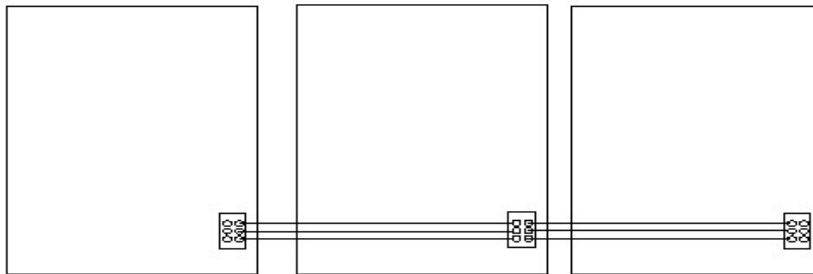
Checksum is the zero based checksum

Responses follow the same format with three extra Commands:

<i>Value</i>	<i>Command</i>
0xff	Invalid Command
0xfe	ACK
0xfd	NAK

#### 4. Multi axis operation

To run multiple cards from the same RS232 port the cards can be connected using the expansion connector JP5. This connector was designed to be daisy chained. The connections for three cards are seen below:



## 5. Firmware Upgrade:

The firmware can be upgraded in the field to fix bugs or add new features. Please contact [support@aquarelasystems.com](mailto:support@aquarelasystems.com) on how to perform the upgrade.