MOTOR-2

Dual Stepper Motor Controller Board

Quick Start Guide

Document Revision: 1.02

Date: 11th June 2010

BiPOM Electronics, Inc.

www.bipom.com

Overview

Thank you for your purchase of MOTOR-2 dual stepper motor controller board.

MOTOR-2 board is a powerful and flexible motor controller boards that is based on dual Allegro Microsystems A3979 Micro stepping Motor Drivers. MOTOR-2 is designed to control two unipolar or bipolar stepper with an output drive capacity of up to 35 Volts DC and ±2.5 Amperes.

The on-board ATMEGA168 microcontroller with a pre-programmed firmware provides additional features:

- RS232 interface to connect to a PC COM port to control motors directly from a PC client software
- SPI /I2C serial interfaces to control the motors through Expansion connector
- 2 buttons, 2 analog inputs and 4 digital inputs to control the motors in manual mode

MOTOR-2 supports many different modes of operation:

- As a standalone microcontroller board
- As a standalone motor controller board
- As a motor controller peripheral board for BiPOM MINI-MAX Series microcontroller boards
- As a motor controller peripheral board for third-party microcontroller systems
- As a motor controller peripheral board for a PC

Configuration

MOTOR-2 has several configuration jumpers. Default jumpers are shown in Figure 1.
.

Note. Settings are common to both motors.

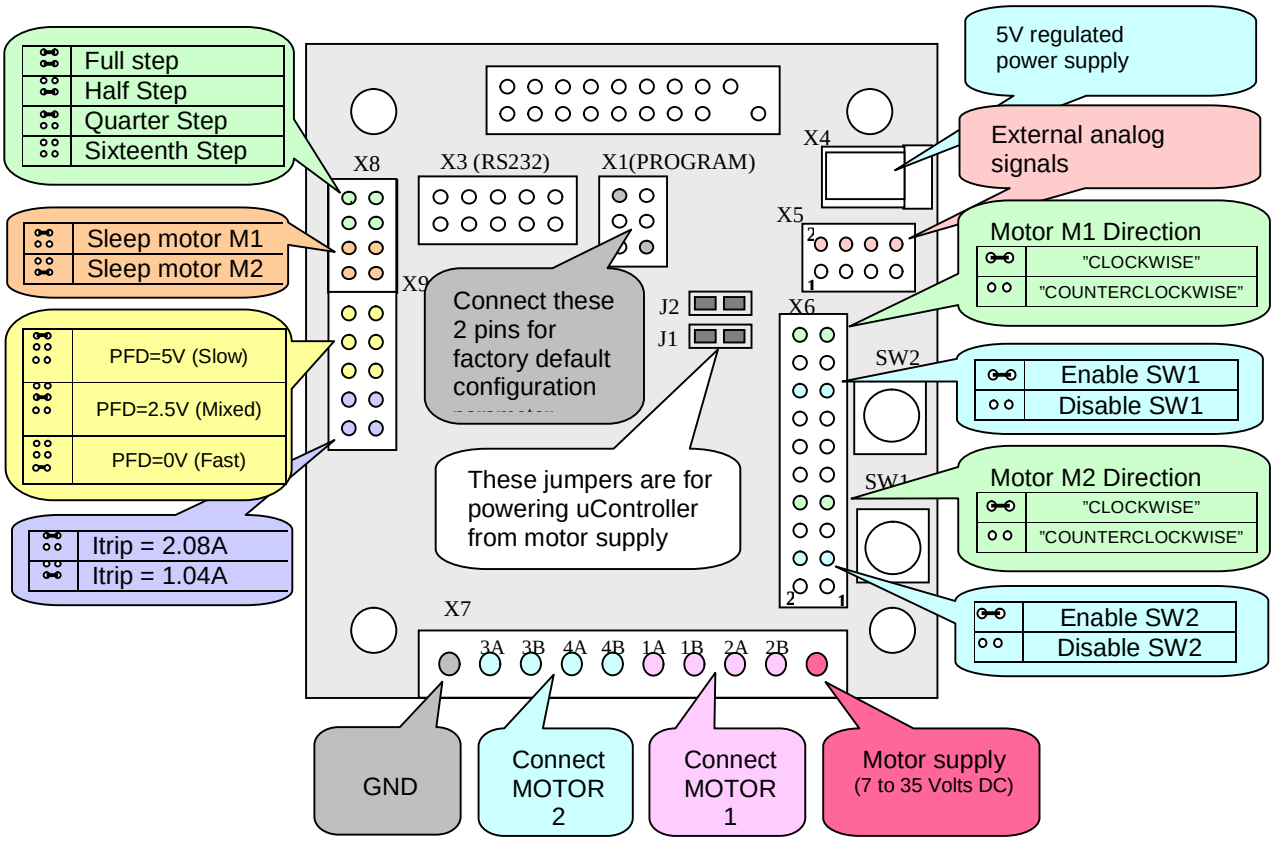


Figure 1: MOTOR-2 connections and settings

Power supply

Normally, MOTOR-2 is powered by two separate power supplies:

The 1st power supply (regulated 5VDC, 100mA) is needed to power the microcontroller section.

The 2nd power supply (regulated or non-regulated 12-36 VDC) is needed to power the motor section. Power of this supply depends on the current requirements of the stepper motors being used.

Microcontroller and motor sections are fully isolated from each other.

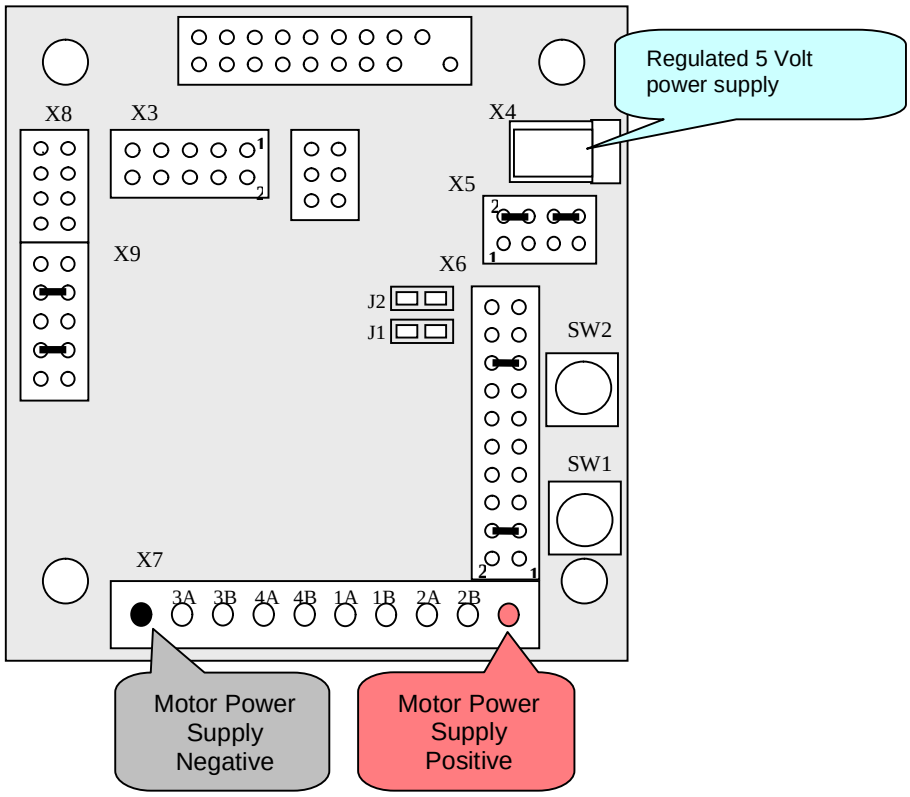
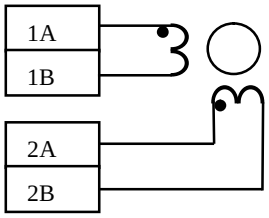
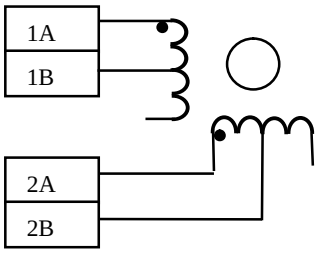


Figure 2: Motor power supply connections

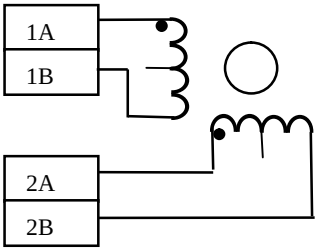
Motor Wiring



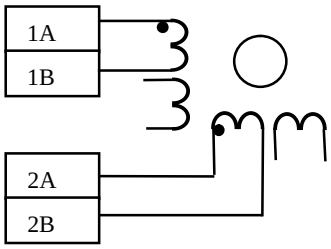
Connection of 4-wire motor



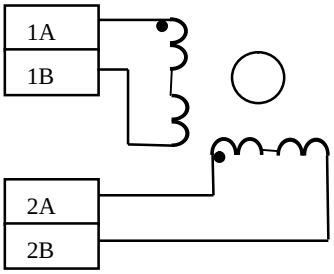
Connection of 6-wire motor



Series Connection of 6-wire motor



Connection of 8-wire motor



Series Connection of 8-wire motor

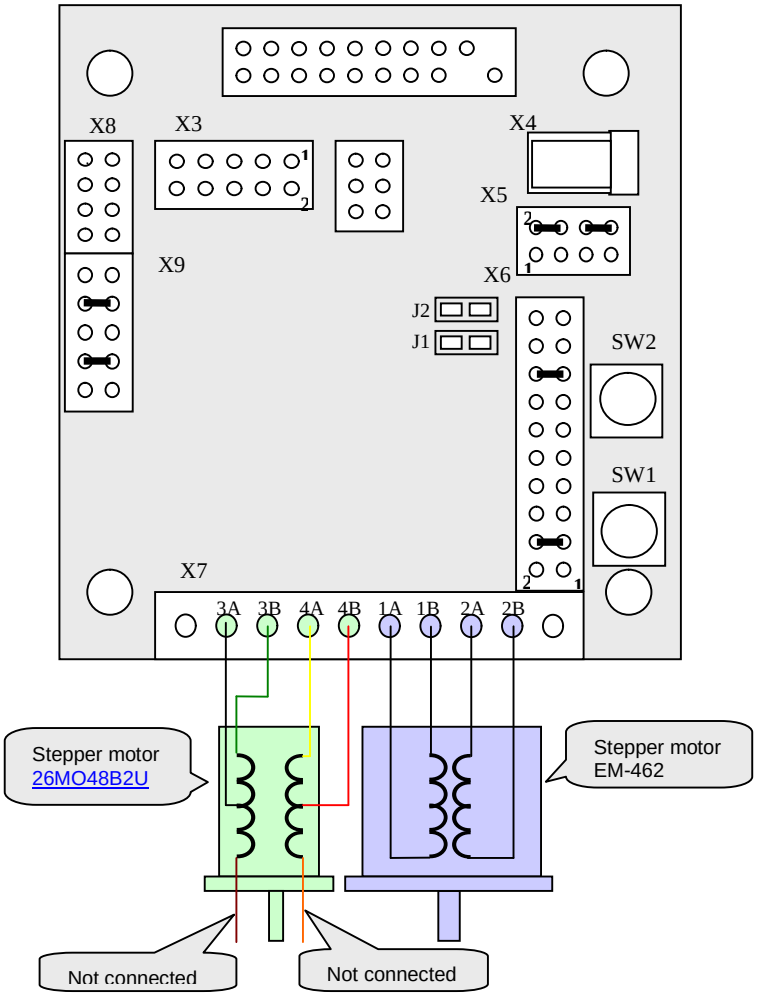


Figure 3: Connection of 4-wire and 6-wire stepper motors simultaneously

Push buttons

The board is shipped with default settings to allow “continuous rotation” mode in manual mode.

To get started:

- Connect one or two motors according to one of the wiring diagrams;
- Check all installed jumpers in accordance with Figure 3;
- Connect the two power supplies (5V and 12V).

Motor(s) will start turning as soon as both power supplies are applied. Press any push button (SW1, SW2) to change the motor speed.

External Analog Signals

The motors can also be controlled using two external 0 to 5 Volt inputs (for example, using potentiometers).

To control motors with external voltages, remove the two jumpers from X5 and connect 2 potentiometers as shown in Figure 4.

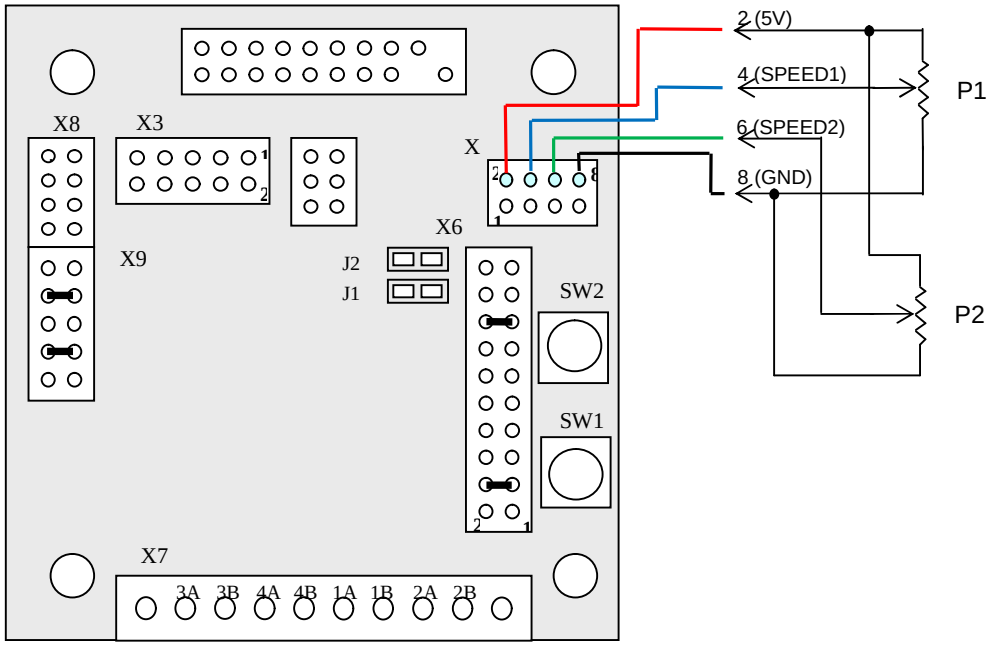


Figure 4: Connection of external potentiometers

Potentiometer value is not critical; it can be anywhere from 470 ohms to 10K.

If a potentiometer is in its middle position that corresponds to 2.3 to 2.7 Volts, the corresponding motor will stop.

If a potentiometer is moved to a position that corresponds to 0 to 2.2 Volts, the corresponding motor will rotate counterclockwise.

If a potentiometer is moved to a position that corresponds to 2.4 to 5.0 Volts, corresponding motor will rotate clockwise.

By playing with the potentiometers, the speed can be varied from 0 to 10 rotations per second.

PC Interface

MOTOR-2 communicates with a PC using a standard null-modem cable. One end of the cable is connected to an available COM port of the PC and the other end of the cable is connected to X3 10-pin dual-row header.

Table 1 shows the pin assignment of X3.

Signal	Pin	Pin	Signal
NC (Not Connected)	1	2	NC (Not Connected)
RXD	3	4	RTS
TXD	5	6	CTS
NC (Not Connected)	7	8	NC (Not Connected)
Ground (GND)	9	10	NC (Not Connected)

Table 1

As shipped, MOTOR-2 supports Modbus protocol for communicating with a host PC. Host PC sends Modbus command requests to MOTOR-2; MOTOR-2 executes an action and returns a reply to the host PC.

Modbus is supported by various third party software for management, configuration, and testing. One such software package is the free **MBReader** utility from KurySoft (www.kurysoft.com).

BiPOM distributes a MOTOR-2 support package that can be downloaded from:
http://www.bipom.com/files/mmcpb2/ mcpb2supportpackage.zip

The support package contains a **MBReader** configuration file ***ModbusAVR.mbc*** that preconfigured **MBReader** for use with MOTOR-2. Please see Figure 5:

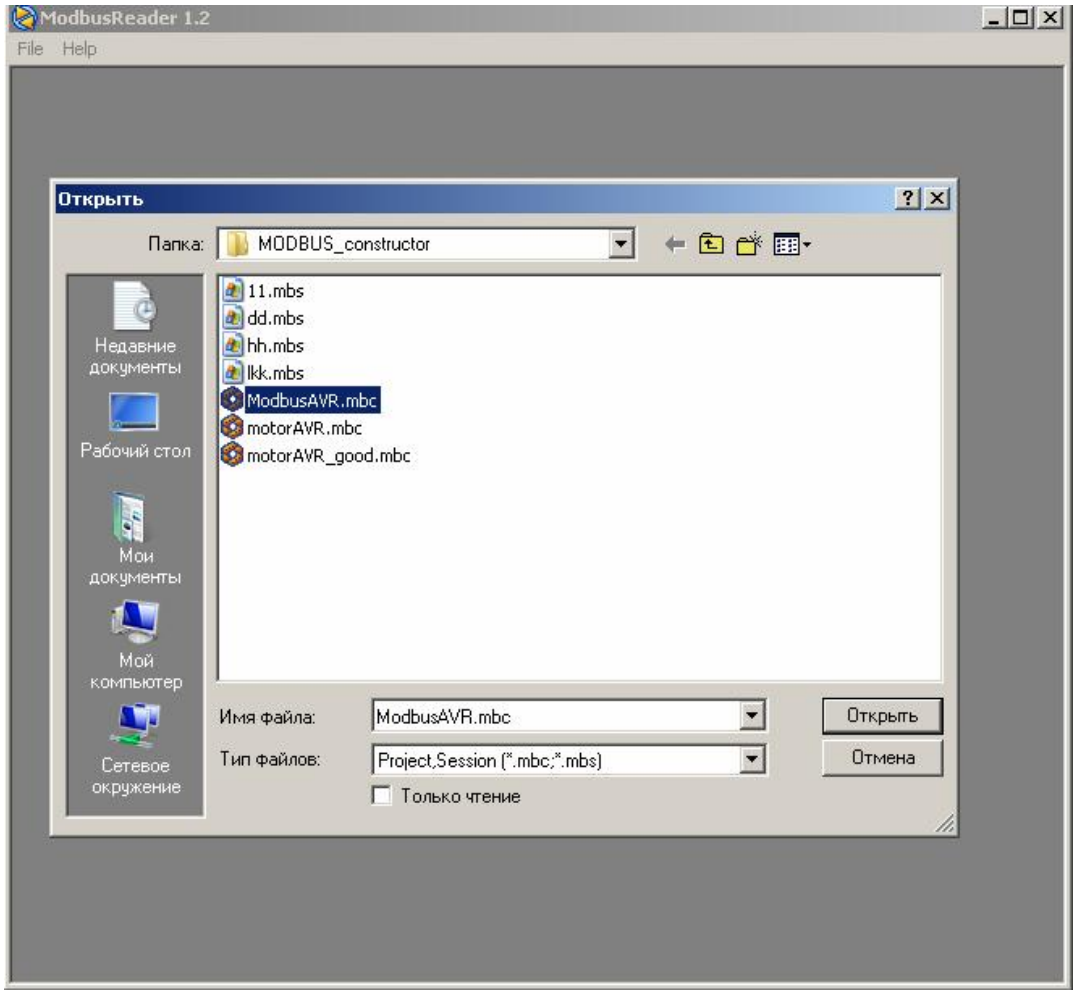


Figure 5: Opening ModbusAVR.mbc

In **MBReader**, select COM Parameters menu option and configure the COM port as follows:

Protocol	**0 (RTU)**
Baud Rate	**19200**
Stop Bits	**1**
Parity	**2 (EVEN)**

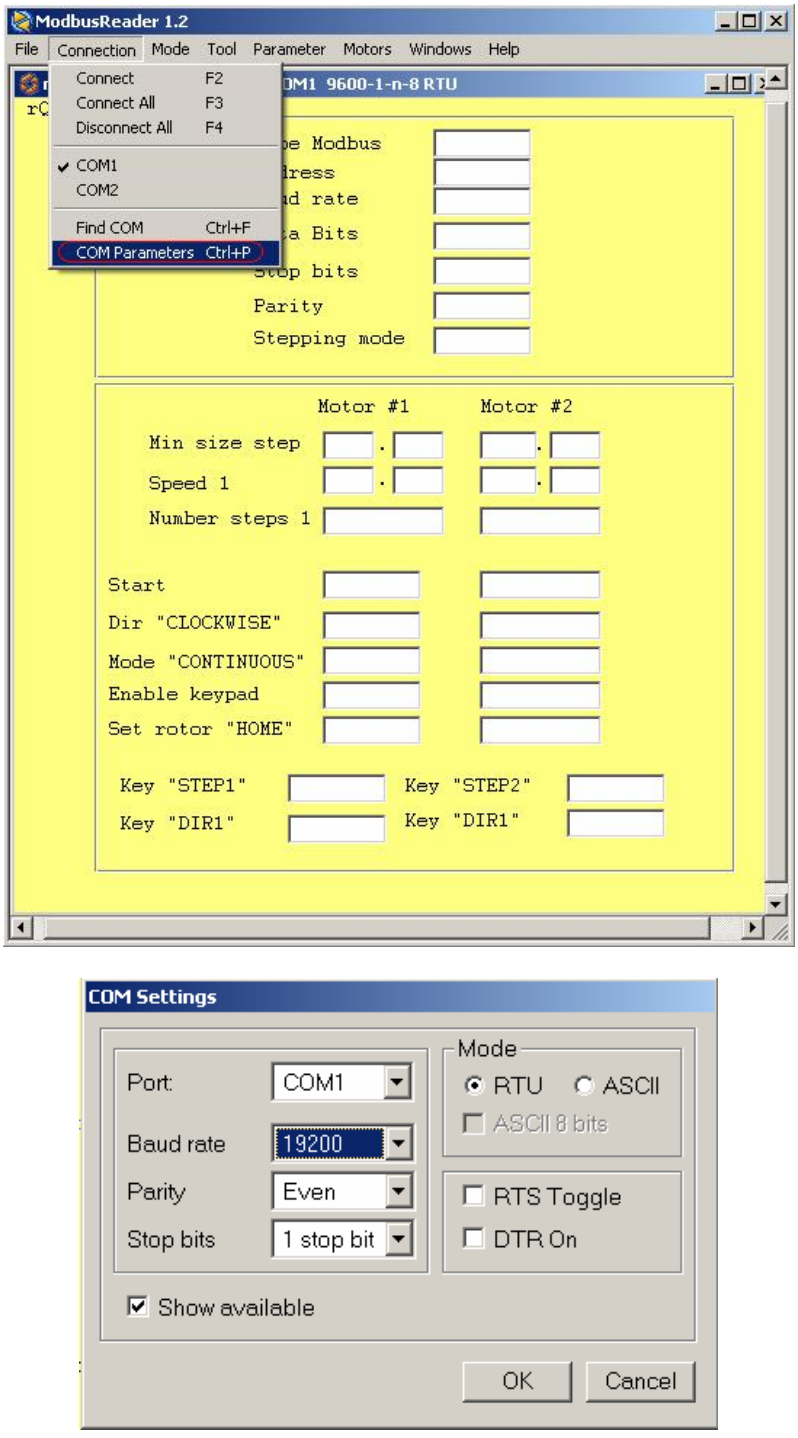


Figure 6: COM parameters

From the Connection menu, select the COM port that MOTOR-2 is connected. On many computers, there is only one COM port so this will typically be COM1.

Again, from the Connection menu, select Connect. **MBReader** will connect to MOTOR-2 through the selected COM port and periodically read and display the current parameters.

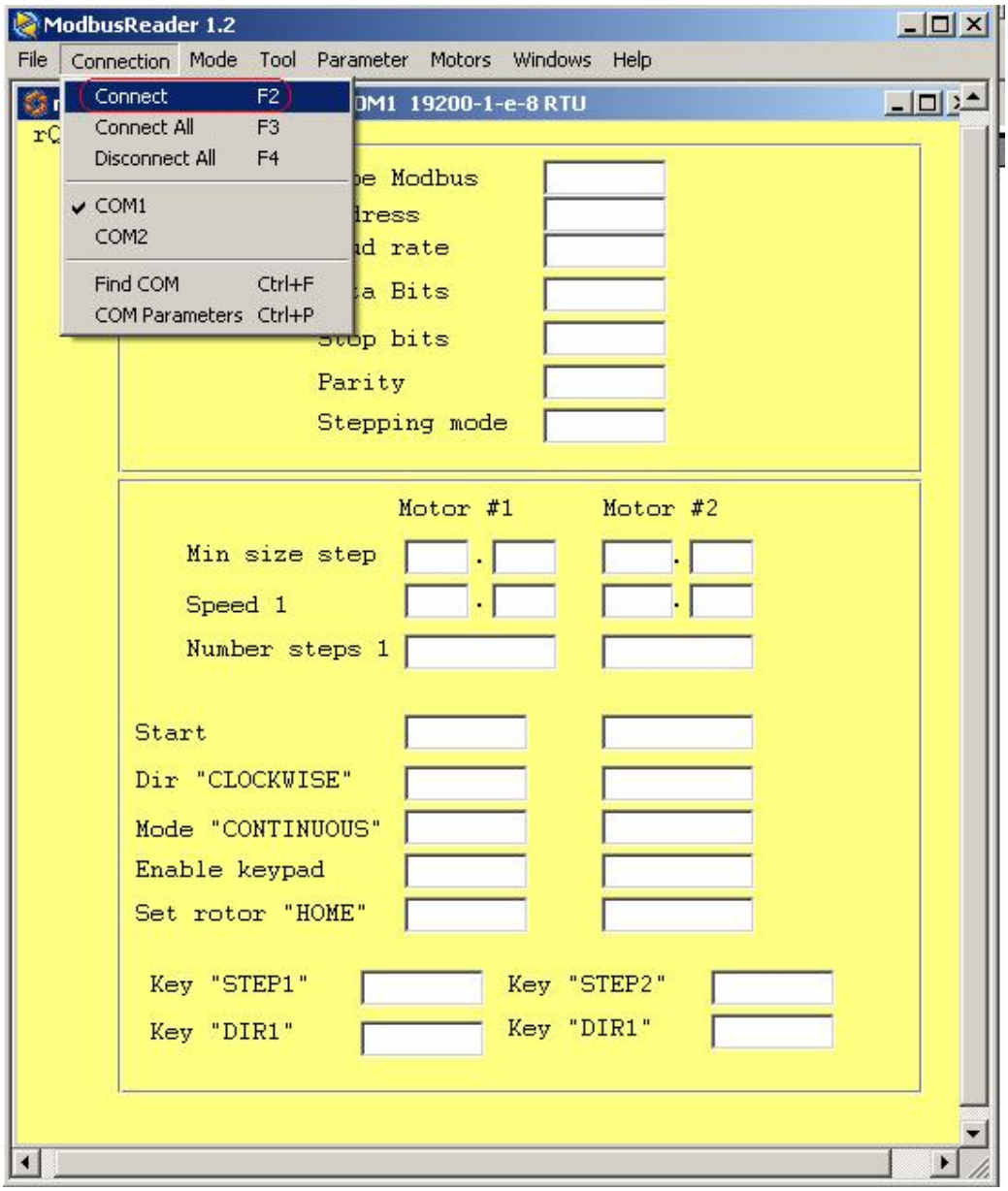


Figure 7: Connection start

Selections under the **Parameter** menu effect Modbus settings as shown in Figure 8:

Protocol type (MODBUS/RTU or MODBUS/ASCII), Modbus device address, baud rate, number of data bits, stop bits, parity, and the motor stepping mode.

If some of the parameters are modified by mistake, MOTOR-2 may lose communications with the host PC.

To restore default settings of the board it is necessary to:

1. **Turn off MOTOR-2**
2. **Install a jumper between 1st and 6th pins of X1 header ("MISO" and "GND" pins)**
3. **Turn on MOTOR-2**
4. **Wait for 3 seconds**
5. **Turn off MOTOR-2**
6. **Remove the jumper**

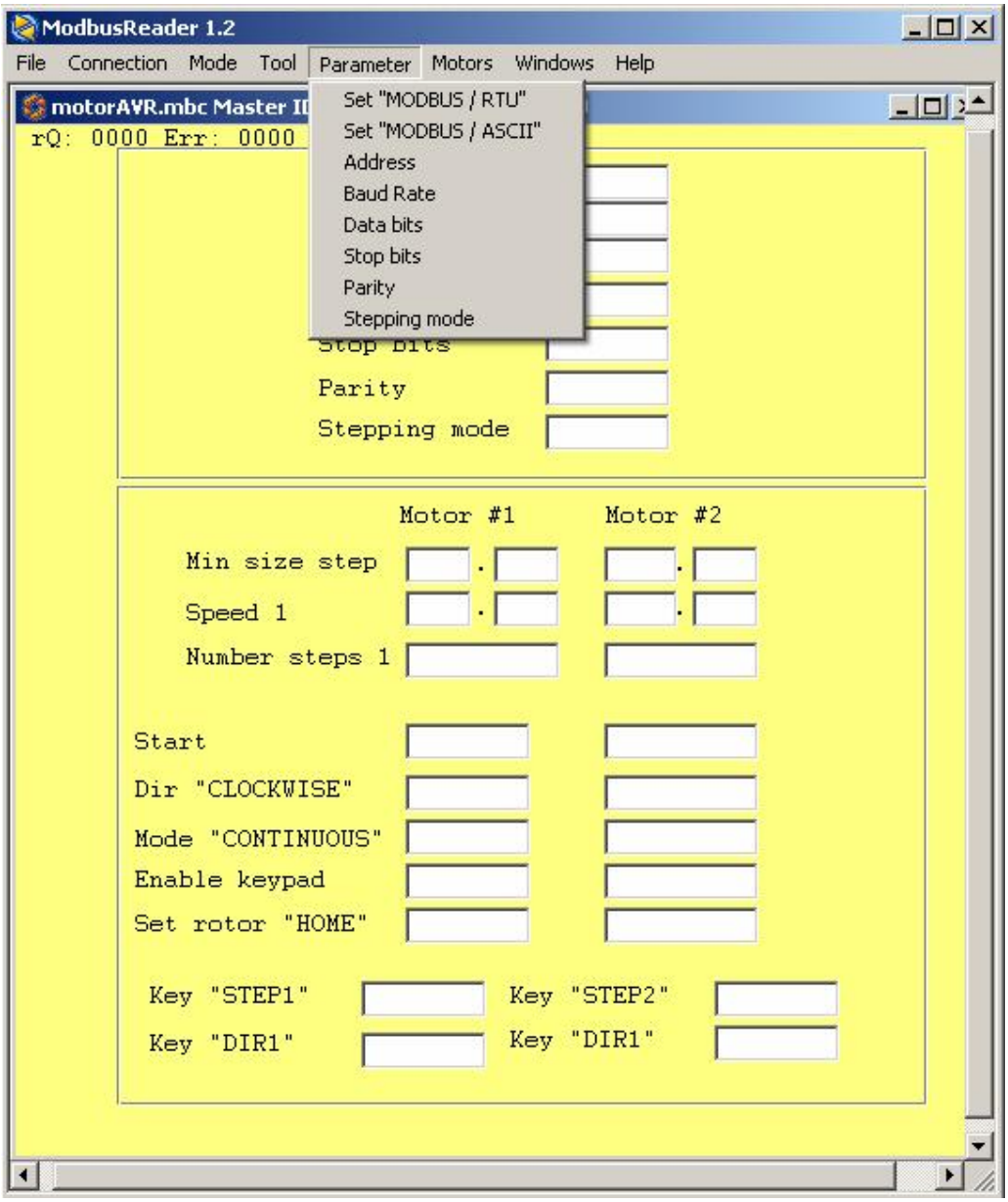


Figure 8: Parameters

Motors menu commands control the motors:

- Start and stop
- Set direction. To activate this feature it is necessary to execute Disable Keypad Motor #1, 2 command as the first step. By default, MOTOR-2 monitors the DIR1, DIR2 digital inputs to change directions.
- Select motor modes
- Disable/enable SW1,SW2 buttons (Disable/Enable Keypad Motor #1,2 command);
- Set the minimum step size for the motor
- Specify the necessary number of steps to operate in "step-by-step" mode;
- Set the motor speed
- Move motors to “HOME” position.

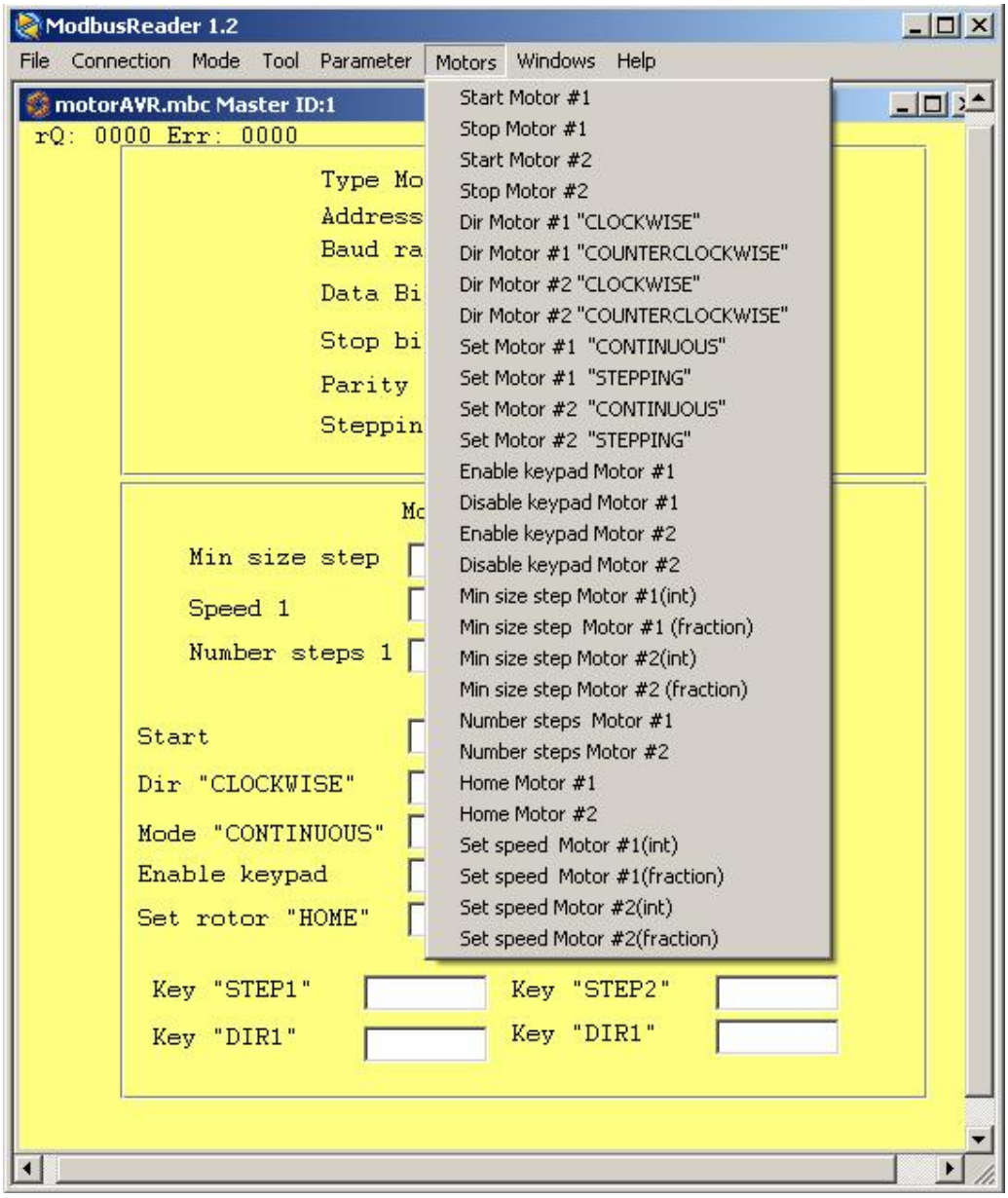


Figure 9: Motor Commands

Appendix A: MOTOR-2 Modbus Map

Table 2 shows the Modbus map of all sub-systems that are defined by COIL OUTPUT, INPUT, and HOLDING registers.

Coil output registers

Available Modbus functions:

- Function № 1 (0x01) Read Coils
- Function № 5 (0x05) Write Single Coil
- Function № 15 (0x0F) Write Multiple Coils

Modbus Logical address	Data Type	Description	Options	Default value
00001	bit	Motor #1 Control	0 - Stop Motor 1 - Start Motor	0
00002	bit	Motor #2 Control	0 - Stop Motor 1 - Start Motor	0
00003	bit	Motor #1 Direction	0 - Counterclockwise 1 - Clockwise	1
00004	bit	Motor #2 Direction	0 - Counterclockwise 1- Clockwise	1
00005	bit	Motor #1 Mode	0 - Step control 1 - Continuous	1
00006	bit	Motor #2 Mode	0 – Step control 1 – Continuous	1
00007	bit	Motor #1 Button	0 - Disable 1 - Enable	1
00008	bit	Motor #2 Button	0 - Disable 1 - Enable	1

Table 2

Examples of Modbus RTU protocol

Start Motor #1

Command:

01	05	00	00	FF	00	8C	3A
Device	Function	Address	Address	Value	Value	Checksum	Checksum

Reply:

01	05	00	00	FF	00	8C	3A
Device	Function	Address	Address	Value	Value	Checksum	Checksum

Stop Motor #2

Command:

01	05	00	01	00	00	9C	0A
Device	Function	Address	Address	Value	Value	Checksum	Checksum

Reply:

01	05	00	01	00	00	9C	0A
Device	Function	Address	Address	Value	Value	Checksum	Checksum

Check Motor #1 Operation

Command:

01	01	00	00	00	01	FD	CA
Device	Function	Address	Address	Count	Count	Checksum	Checksum

Reply:

01	01	01	01	90	48
Device	Function	Count	Value	Checksum	Checksum

Start Motor #1, Stop Motor #2 simultaneously

Command:

01	0F	00	00	00	03	01	03	CF	56
Device	Function	Address	Address	Quantity	Quantity	Count	Value	Checksum	Checksum

Reply:

01	0F	00	00	00	03	15	CA
Device	Function	Address	Address	Quantity	Quantity	Checksum	Checksum

Input registers

Available Modbus functions:

- Function № 4 (0x04) Read Input registers

Logical address ModBus	Data storage	Description
10001	word	Read “STEP1” Motor #1
10002	word	Read “STEP2” Motor #2
10003	word	Read “DIR1” Motor #1
10004	word	Read “DIR2” Motor #2

Table 3: Input registers

Examples of Modbus RTU protocol

Read all input register

Command:

01	*04*	*27*	*10*	*00*	*04*	*FA*	*B8*
Device	*Function*	*Address*	*Address*	*Quantity*	*Quantity*	*Checksum*	*Checksum*

Reply*:*

01	*04*	*08*	*00*	*00*	*00*
Device	*Function*	*Count*	*Value*	*Value*	*Value*

00	*00*	*00*	*00*	*00*	*24*	*0D*
Value	*Value*	*Value*	*Value*	*Value*	*Checksum*	*Checksum*

Holding registers

Available Modbus functions:

- Function № 3 (Read Holding Registers)
- Function № 6 (Write Single Register)
- Function № 16 (Write Multiple registers)

Table 4: Holding registers

Modbus Logical address	Address as Hex	Data type	Default value	Limit	Description
40001	9C41	word	0	0 or 1	Modbus Type : 0 - Modbus RTU 1 - Modbus ASCII
40002	9C42	word	1	0…247	Address device (EEPROM)
40003	9C43	word	19200	1200,2400,4800, 9600,19200, 38400, 57600	Baud rate (EEPROM)
40004	9C44	word	8	7,8	Data Bits : 7 for 7 data bits 8 for 8 data bits
40005	9C45	word	1	1 or 2	Stop Bit: 1 for 1 stop bit 2 for 2 stop bits
40006	9C46	word	1	0,1,2	Parity: 0 - Disabled 1 - Even Parity 2 - Odd Parity
40007	9C47	word	16	1,2,4,16	Step Mode: 1 - Full step 2 - Half step 4 - Quarter step 16 - Sixteenth step
40008	9C48	word	7	0…20	Motor #1 Step Size. Integer part (EEPROM)
40009	9C49	word	50	0…99	Motor #1 Step Size. Fractional part (EEPROM)
40010	9C4A	word	7	0…20	Motor #2 Step Size. Integer part (EEPROM)
40011	9C4B	word	50	0…99	Motor #2 Step Size. Fractional part (EEPROM)
40012	9C4C	word	1	0…1000	Motor #1 Speed. Integer part
40013	9C4D	word	0	0…99	Motor #1 Speed. Fractional part
40014	9C4E	word	1	0…1000	Motor #2 Speed. Integer part
40015	9C4F	word	0	0…99	Motor #2 Speed. Fractional part
40016	9C50	word	0	0	Reserved
40017	9C51	word	0	0	Reserved
40018	9C52	word	0	0	Reserved
40019	9C53	word	0	0	Reserved
40020	9C54	word	0	0…65535	Number of steps for Motor #1
40021	9C55	word	0	0…65535	Number of steps for Motor #2

Examples of Modbus RTU protocol

Set Motor #1 speed

Command:

01	*06*	*9C*	*4B*	*00*	*05*	*17*	*8F*
Device	*Function*	*Address*	*Address*	*Value*	*Value*	*Checksum*	*Checksum*

Reply*:*

01	*06*	*9C*	*4B*	*00*	*05*	*17*	*8F*
Device	*Function*	*Address*	*Address*	*Value*	*Value*	*Checksum*	*Checksum*

Set 128 steps for Motor #1

Command:

01	*06*	*9C*	*53*	*00*	*80*	*56*	*2B*
Device	*Function*	*Address*	*Address*	*Value*	*Value*	*Checksum*	*Checksum*

Reply*:*

01	*06*	*9C*	*53*	*00*	*80*	*56*	*2B*
Device	*Function*	*Address*	*Address*	*Value*	*Value*	*Checksum*	*Checksum*