KT- 5193A Modbus Programmable Stepper Motor Controller

The Modbus Programmable Stepper Motor Controller is a four axis
multifunction programmable stepper motor controller which utilises the modbus protocol for communications.
The Modbus Programmable Stepper Motor Controller is designed for use as an alternative to an expensive PLC for simple process type applications. It can be used as a standalone controller to control up to 4 stepper motors and I/O or in conjunction with a PLC to control stepper motors
Using a PC and the supplied program “StepModb” a user can develop a program and then download it via a RS232 serial link, into the controllers non volatile (EEPROM) memory.
Alternatively the controller can be controlled via a PLC using the modbus protocol.

Up to four stepper motors can be used concurrently. Each Motor has its own direction and step outputs. If a motor is not being used the two outputs for the motor can be used as general outputs for control purposes. Four digital inputs are present and these can be used as limit switches for the stepper motors or as general inputs for program control.

Instructions available are:

Name	Description
Absolute move	Moves motors to an absolute position. Tripping the limit switch stops the motor moving and advances to the next instruction. Allowed values: -32767 to 32767
Relative move	Moves motors a number of steps relative to the current position. Tripping the limit switch stops the motor moving and advances to the next instruction. Allowed values: -32767 to 32767
Relative move then wait for Lim Sw	Moves motors relative to the current position and then waits for the limit switch to be tripped. Allowed values: -32767 to 32767
Wait Milliseconds	Waits a number of milliseconds x 0.2 (approx). Allowed values: 1 to 32767
Digital Output On/Off	Sets unused outputs on or off according to the value. Allowed values: see note for output on parameter
Max rate of stepping	Sets the maximum rate of stepping for each of the motors, in milliseconds x 0.2 (approx). Allowed values: 1 to 9999
Start rate	Sets the minimum rate of stepping for each of the motors, in milliseconds x 0.2 (approx). Allowed values: 0 to 9999
Acceleration increment	Sets the increment used when accelerating from the start rate to the maximum rate, in milliseconds x 0.2 (approx). Allowed values: 1 to 9999
Set Position	Set the current position of the motors to a new value. Allowed values: -32767 to 32767
Jump to instruction	Jump to any instruction Allowed values: 1 to 127
Wait for input	Wait for a limit switch to be high, rising edge or falling edge. Allowed values: see note for input parameter
Jump to instruction 1	Returns to the start of the program, this is the default instruction Allowed values: All parameters are ignored

Using the Modbus Programmable Stepper Motor Controller with this list of instructions, allows various process tasks to be completed with a number of stepper motors and digital I/O.

Holding Registers:

Instructions are loaded into the controllers memory to what are called Modbus Holding Registers. The registers 1 to 20 are control registers and the registers 21 to 276 are available for program space.

The holding registers are:

No.	Description	Read/Write
1	Parameter 1 of current instruction	R
2	Parameter 2 of current instruction	R
3	Parameter 3 of current instruction	R
4	Parameter 4 of current instruction	R
5	Function code of current instruction	R
6	Status of Motor 1	R
7	Status of Motor 2	R
8	Status of Motor 3	R
9	Status of Motor 4	R
10	Position of Motor 1	R
11	Position of Motor 2	R
12	Position of Motor 3	R
13	Position of Motor 4	R
14	Line Number of current instruction	R
15	Single Stepping Enable	R/W
16	Single Stepping Step	R/W
17	Single Stepping Ready	R
18	Stop	R/W
19	No of motors (1 to 4)	R/W
20	Control Register	R/W
21-276	Program space	R/W

1-4: These registers hold the parameters of the current instruction being executed

5: This is the type of instruction

6-9: The status of each motor is shown, each bit in the status registers represents the following

Bit	16	8	4	2	1
Desc.	Limit Switch A	Done	Limit Switch B	Direction	Moving

Limit Switch A is current status of limit switch 1 = ON
Done is 1 if this parameter has finished executing
Limit Switch B is 1 if the limit switch has gone from off to on in this instruction
Direction is 0 if forward and 1 if reverse
Moving is 1 if the motor is moving

10-13: These registers hold the current position of each of the motors

14: This is the number of the instruction currently being executed

15: Write 1 to this register to allow the controller to single step through instructions

16: Write 1 to this register to execute one instruction when in single stepping mode

17: This register is 1 when the program is ready to execute the next instruction when in single stepping mode

18: If 1 the controller will not perform any operation

19: The number of motors to use

20: This is the Control Register. Writing 0 to this register will make the program stop executing, a positive number will make the program execute that number of times and a negative number will make the program run indefinitely.

21-276: This is the program space

Memory Allocation:

Program space is allocated in the following manner.

Each command begins with the function code of the instruction in the first register, and then a number of parameters consecutively in the next registers. The number of parameters is always equal to the number of motors. This means that the start of each command is at registers

21,23,25,27…	For 1 motor
21,24,27,30…	For 2 motors
21,25,29,33…	For 3 motors
21,26,31,36…	For 4 motors

This allows for more commands to be stored in memory, if fewer motors are used.

The number of commands available in each case are:

No. of motors	Max no. of commands
1	120
2	80
3	60
4	50

The Function Codes and parameters of the instructions are

No.	Description	Param1	Param2	Param3	Param4
1	Absolute move	Target 1	Target 2	Target 3	Target 4
2	Relative move	Target 1	Target 2	Target 3	Target 4
3	Relative move then wait for Lim Sw	Target 1	Target 2	Target 3	Target 4
4	Wait Milliseconds	Time to wait	DC	DC	DC
5	Dig On/Off	Output Param	DC	DC	DC
6	Max rate of stepping	Max Rate 1	Max Rate 2	Max Rate 3	Max Rate 4
7	Start rate	Start rate 1	Start rate 2	Start rate 3	Start rate 4
8	Acceleration increment	Increment 1	Increment 2	Increment 3	Increment 4
9	Set Position	Position 1	Position 2	Position 3	Position 4
10	Jump to instruction	Instruction No.	DC	DC	DC
11	Wait for input	Input Param	DC	DC	DC
Anything Else	Jump to instruction 1	DC	DC	DC	DC

DC=Don't Care

Note: There must be a register space for the DC

The Output Parameter determines which digital outputs to turn on or off

Bit	32	16	8	4	2	1
Desc.	S2	D2	S3	D3	S4	D4

1=output on 0=output off

Only outputs which are not being used as motors can be used.

The Input Parameter determines which of the switches to wait for

Bit	2048	1024	512	256	128	64	32	16	8	4	2	1
Desc.	L1F	L1R	L1H	L2F	L2R	L2H	L3F	L3R	L3H	L4F	L4R	L4H

F=Falling R=Rising H=High

If multiple bits are set, any of the conditions being met will make the program advance to the next instruction.

Assembly & Testing the Kit:
All the components are mounted on the dual layer PCB. The list of parts is shown on page 7 of this manual.

Start by soldering the resistors and diodes and then the capacitors and crystal. Next add the IC sockets, voltage regulator and DIP switch. Finally add the terminals and 9 pin D connectors, noting the placement of the male and female connectors.

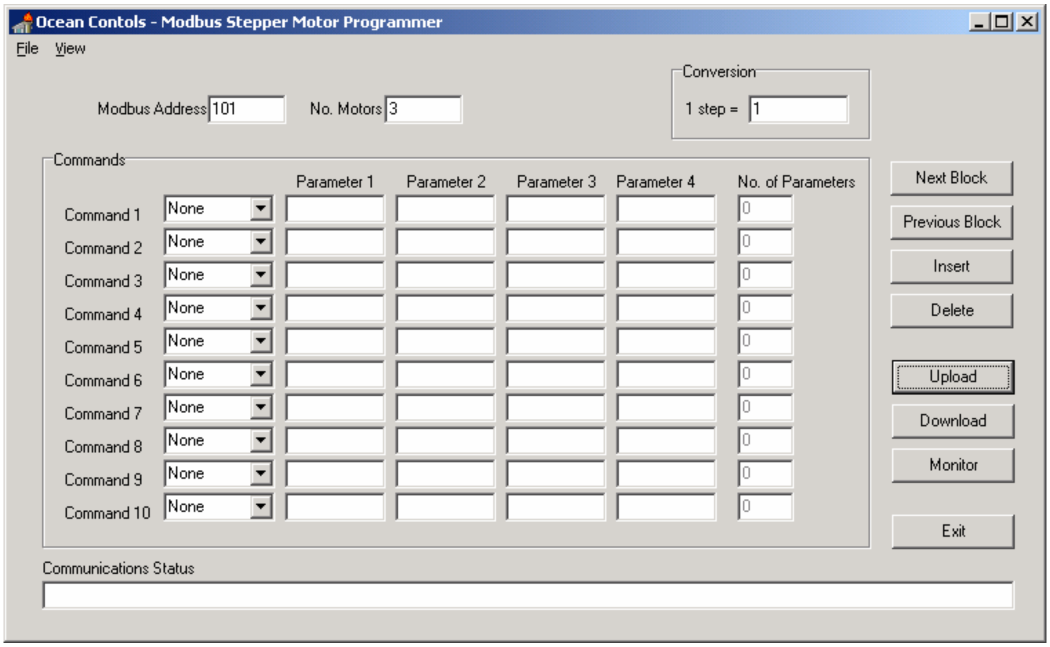
To test the controller, connect 12V positive to the Vs Terminal and negative to the COM terminal and measure the voltage across pins 10 and 20 of the socket for the 74HC245, ensuring that it is 5V. If this is ok remove the power and insert the microcontroller, max232 and solder in the 74HC245.

Using the controller:
The controller requires 12Vdc to run. Apply 12V positive to the Vs Terminal and negative to the COM terminal.
A connection to a PC should be made using a standard RS-232 9-pin straight through cable.
Limit switches need to be normally open (NO) switches, which are connected between the limit switch input (L1-L4) and a COM terminal.
Motor driver step inputs are connected to the step outputs (S1-S4) of the controller and direction connections are connected to the dir outputs (D1-D4). A common connection will need to be made as well and this is done by connecting to a COM terminal.
If less than four motors are used the motors are connected to the lowest numbered outputs. The unused outputs will be available for digital control.
The modbus address of the controller is set by the DIL switches, as shown in the following table.

S1	S2	Address
0	0	101
0	1	102
1	0	103
1	1	104

This allows up to four controllers to be used on the one serial port.

The software provided the CD called StepModb.exe can be used to program the controller. The main screen of the software is shown in the figure below.



Modbus Address 101

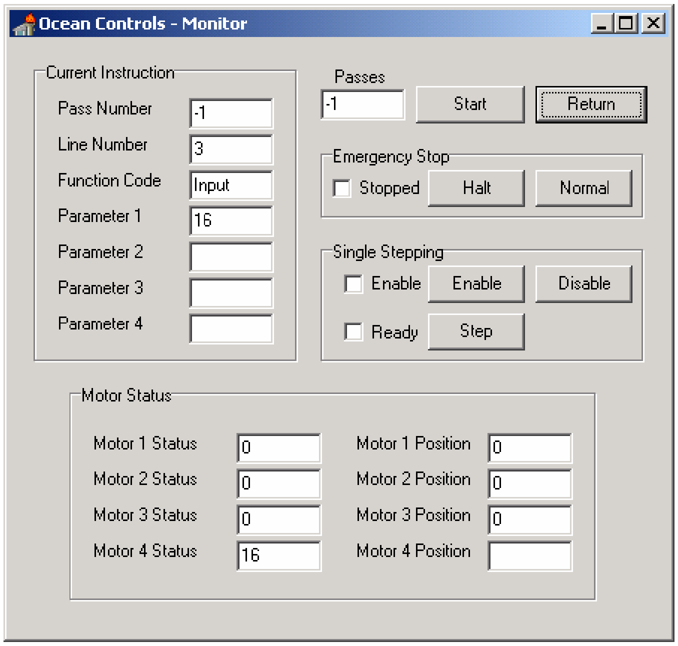
No. Motors 3

1 step = 1

		Parameter 1	Parameter 2	Parameter 3	Parameter 4	No. of Parameters
Command 1	None					0
Command 2	None					0
Command 3	None					0
Command 4	None					0
Command 5	None					0
Command 6	None					0
Command 7	None					0
Command 8	None					0
Command 9	None					0
Command 10	None					0

First set the Comm Port to match the Comm port to which the controller is attached. Do this by going to View-> Comm Port and then select the correct port number from the list.
Set the Modbus Address to match the address of the controller you wish to use, and then set the number of motors.
The conversion parameter can be used as a scaling factor.
The program can now be written for the controller.
To do this select an instruction from the drop down list and add the parameters you require.
More commands can be added by clicking on the Next Block button.
Commands can be inserted or removed by using the Insert and Delete buttons.
Programs can be stored on the PC by going to File-> Save.
To load a program saved earlier, go to File-> Load.
To view the program in the controller click Upload.
Once all the commands required have been entered, click Download to send the program to the controller.

Once the program has downloaded click Monitor to bring up the Monitoring Window, shown below.
Set the number of times you wish the program to run in the Passes textbox. Use -1 if you wish the program to run indefinitely. Click Start to run the program.
The Halt button in the Emergency Stop group will stop the controller mid instruction and the Normal button will return the controller to normal operation.



Pass Number	-1
Line Number	3
Function Code	Input
Parameter 1	16
Parameter 2	
Parameter 3	
Parameter 4	
Passes	-1

Motor 1 Status	0	Motor 1 Position	0
Motor 2 Status	0	Motor 2 Position	0
Motor 3 Status	0	Motor 3 Position	0
Motor 4 Status	16	Motor 4 Position	

The Monitoring Window shows various parameters on the status of the controller including the current motor positions and status, as well as information on the instruction currently being executed.
In the Single Stepping group, click Enable to put the controller into single stepping mode. When the Ready box is checked the controller has completed its first command and is ready to start the next one. Click Step to execute the next command. The Ready checkbox will become unchecked until the controller has finished executing the command, at which point the Ready checkbox will be checked again and the process can be repeated.

Example Process Programs:

1. Injection Molding Process
In this example a single stepper motor drives a syringe which slowly pushes material into a mould, and then quickly returns to its home position, where an indicator is lit and the controller waits for an operator to push a button to restart the process.
Set the number of motors to 1.
Connect the stepper motor to the S1, D1 and COM terminals.
Connect an ‘Operate’ Switch between the L4 and COM terminals.
Connect a ‘Go’ Pushbutton between the L3 and COM terminals.
Connect a ’Ready’ indicator to the D4 terminal (via a Relay).

Command No.	Instruction	Parameter	Description
1	Wait	100	Wait 100 ms so that command 2 can work properly
2	Wait Input	2	Wait for the Operate switch to go from Off to On (In the event of power failure, this ensures that the machinery does not operate until the operator has ensured the machinery is in the start position and the switch is switched Off and then On again)
3	Set Posn	0	Sets the current position of the motor to be the home position
4	Start Rate	20	Set the starting rate of the stepper motor to 20
5	Accel Incr	1	Set the acceleration increment to 1
6	Wait Input	1	Wait for the Operate switch to be in the On position
7	Dig On/Off	1	Turn on the Ready indicator
8	Wait Input	16	Wait for the Go button to go from Off to On
9	Dig On/Off	0	Turn off the Ready indicator
10	Rate	10	Set the speed of the motor to 10 (slow)
11	Abs Move	1000	Move slowly forwards 1000 steps (Injection)
12	Rate	1	Set the speed to 1ms (fast)
13	Abs Move	0	Go back to the home position quickly
14	Jump To	5	Jump to command 5

Enter these commands and click Download.
Put -1 in the No. of Passes box and click Start to make the controller run indefinitely.

PLC Control:
To use the Modbus Stepper Motor Controller with a PLC follow these steps:

1. Write the number of motors to holding register 19
2. Write 0 to register 20
3. Disconnect and reconnect power to the stepper motor controller
4. Write the instructions (Function code and Parameters) to be executed, to the registers beginning at 21 (see earlier section on allocation of program space)
5. Write 0 to the register following the final register of the instruction which was written
6. Write 1 to register 20 to begin execution
7. Wait for register 20 to return to 0
8. Repeat steps 4 to 8 for each group of instructions

Writing anything other than 0 to register 18 will halt program execution immediately, normal operation will only return when register 18 is 0. This register is stored in memory and is retained if power is lost.

Parts List:

2 14-pin DIP sockets (U1A, U1B)
1 18-pin DIP socket (U3)
1 20-pin DIP socket (U2)
1 2-way DIP switch (SW1)
1 D9 Female right angle connector (K2)
1 D9 Male right angle connector (K3)
6 3-way terminal blocks (T1, T3-T4)
1 2-way terminal block (T2)
1 20MHz crystal (X1)

Semiconductors:
1 Atmel ATMega168 programmed microcontroller (U1)
1 74HC245 Octal Buffer (U2)
1 MAX232 RS232 to TTL Level Shifter (U3)
1 7805 5V Voltage Regulator (VR1)
1 1N4004 silicon diode (D1)
1 1N4148 silicon diode (D2)

Resistors:
1 10K SIL pull up network (S1)
4 10K (R1-R4)
1 18K (R5) maybe substituted with a 20K or 22K

Capacitors:
2 22pF ceramic (C1, C2)
4 0.1uF monolythic (C3-C6)
4 1uF electrolytic (C7-C10)

November 2006

The top speed of the KT-5193 Modbus programmable stepper motor controller has been increased to approximately 4.7KHz. All parameters previously measured in milliseconds are now approximately 0.2 x Value in milliseconds. This gives the higher top speed and more discrete speeds can be used.