

ET-M8194H Getting Start

(Version 1.0)



ICP DAS CO., LTD.

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 1997-2008 by ICPDAS Inc., LTD. All rights reserved worldwide.

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Technical Support

If you have problems about using the product, please contact ICPDAS Product Support.

Email: Service@icpdas.com

1	Introduction.....	4
2	Wiring	5
2.1	Electrical Wiring	5
2.2	Ethernet connection	5
3	ET_M8194H Toolkit Installation.....	7
4	Network Setting	11
4.1.1	Setting via RS232	11
4.1.2	Setting via Ethernet.....	13
5	LED Description	15
6	EzMove Utility.....	16
6.1	Utility Layout.....	17
6.2	Utility Features.....	18
6.3	Editing a macro program	19
6.3.1	Macro editing method 1	21
6.3.2	Macro editing method 2.....	21
6.4	Downloading macro programs.....	23
6.5	Executing macro programs	24
6.6	Displaying motion pathway	25
6.7	Sending single commands.....	28
6.8	Modbus message structure.....	28
6.8.1	Reading holding register	29
6.8.2	Writing multiple register	30
7	Libraries	32

1 Introduction

ET-M8194H is an Ethernet based 4-axis stepping/pulse-type motion controller and uses Modbus TCP/IP as a communication protocol between client and server. This intelligent motion controller also has a variety of built in motion control functions, such as 2/3- axis linear interpolation, 2-axis circular interpolation, T/S-curve acceleration/deceleration, various synchronous actions and automatic homing. While driving the motors the motion status and other I/O status can be monitored via Ethernet. The ET-M8194H is suitable for general-purpose motion applications.

ICPDAS provides a large variety of functions and examples to reduce the programming effort of the system developer, making it a highly cost-effective solution for motion builders.

EzMove is a utility assisting the user in writing macro programs and in getting familiar with the ET-M8194H and its motion commands. Furthermore it can be used for motion monitoring and tracking of the motion path.

2 Wiring

2.1 Electrical Wiring

Power requirement 24 ~ 30 V/DC

Attach the earth wire to the **IGND** screw terminal and the power cable (+**24 VDC**) to the +**VS** screw terminal as indicated in Figure 1.

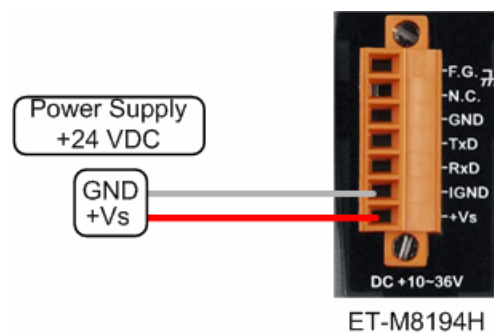


Figure 1: Power supply wiring

2.2 Ethernet connection

The ET-M8194H module can be accessed by the PC in two ways:

Method 1: Connecting to the Local Area Network (LAN)

Plug one end of the RJ-45 internet cable to one the Ethernet switch ports (Port1 or Port2) of the ET-M8194H module and the other end to a Switch/Hub of the local area network.

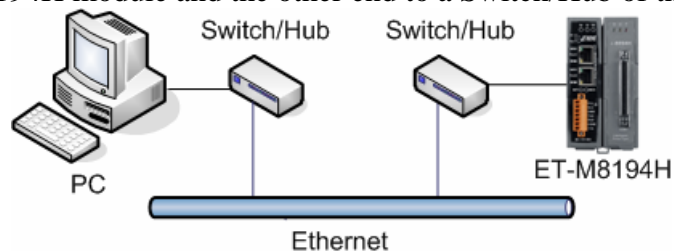


Figure 2: ET-M8194H in a local area network

Method 2: Directly connecting the ET-M8194H with a PC

The computer and ET-M8194H can be directly connected with a crossover cable. It hardwires the Ethernet transmitter on the computer to the receiver on the ET-M8194H device. As a result a crossover cable does not have delays caused by collisions; data can be sent in both directions simultaneously.

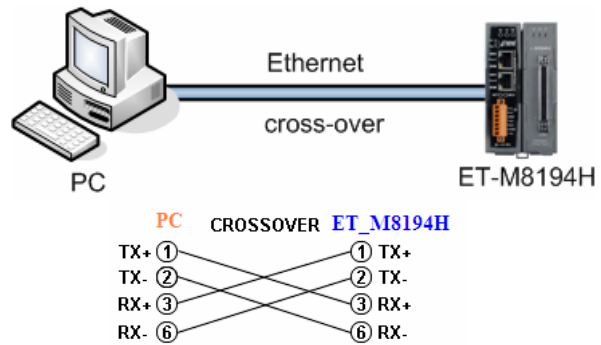
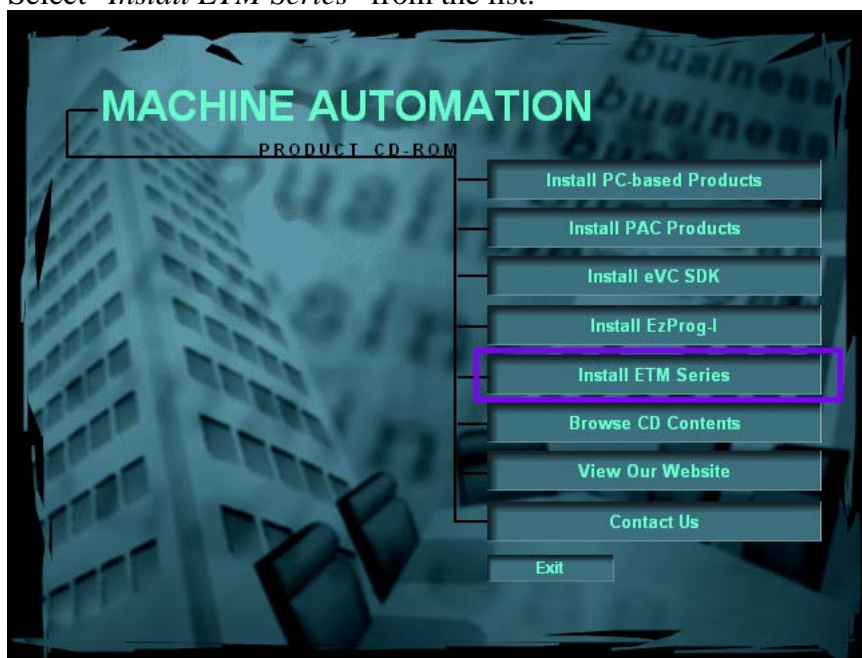


Figure 3: Crossover connection

3 ET-M8194H Toolkit Installation

Step 1: Insert the “*MACHINE AUTOMATION*” DVD in your DVD drive. The Setup program should start automatically. In case the wizard fails to start double click the “Launch” execution file on the DVD.

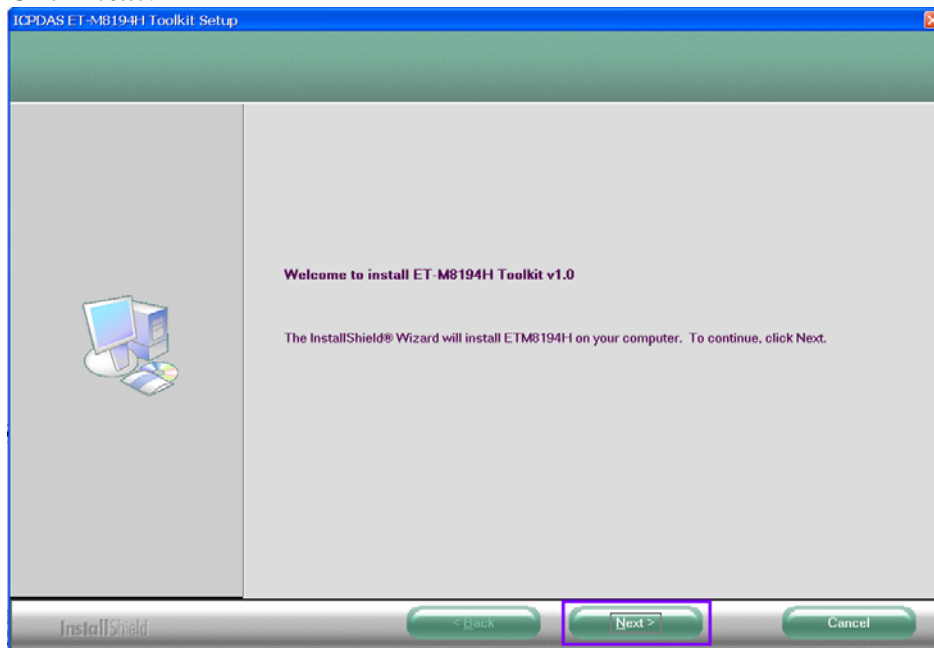
Step 2: Select “*Install ETM Series*” from the list.



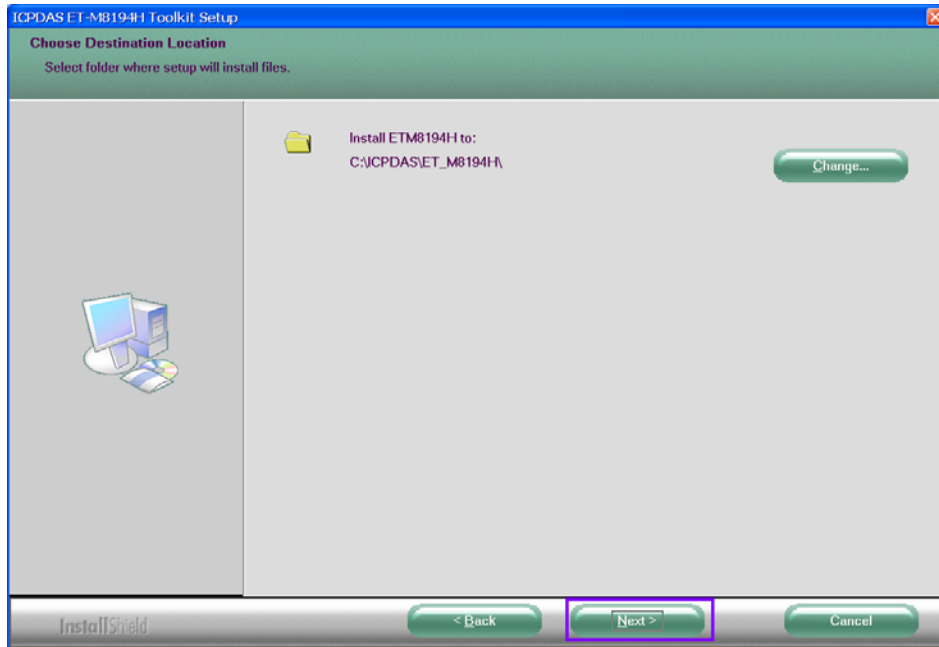
Step 3: Select “*Install ET-M8194H*”:



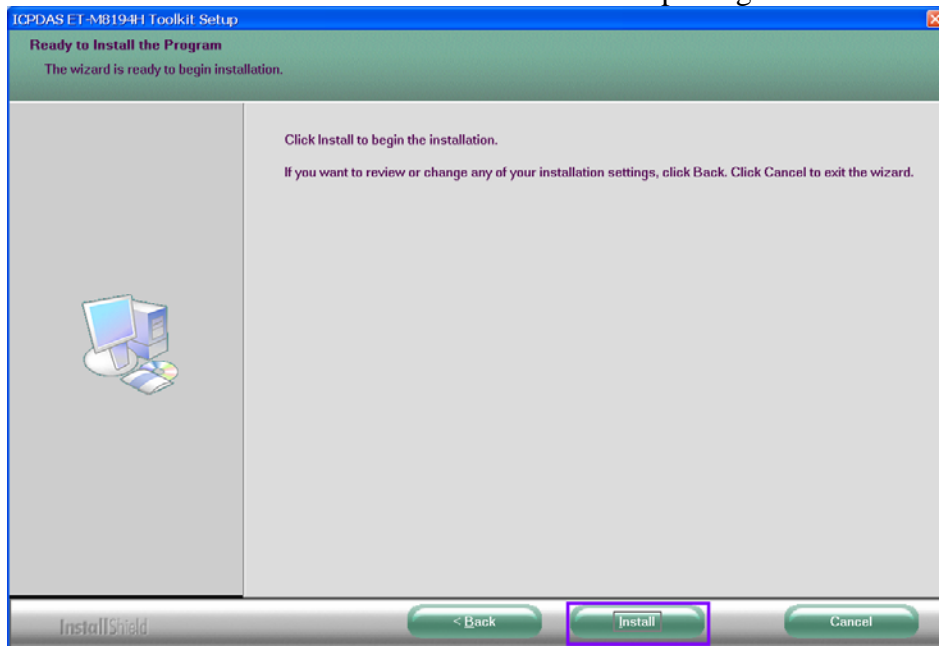
Step 4: Click *Next*.



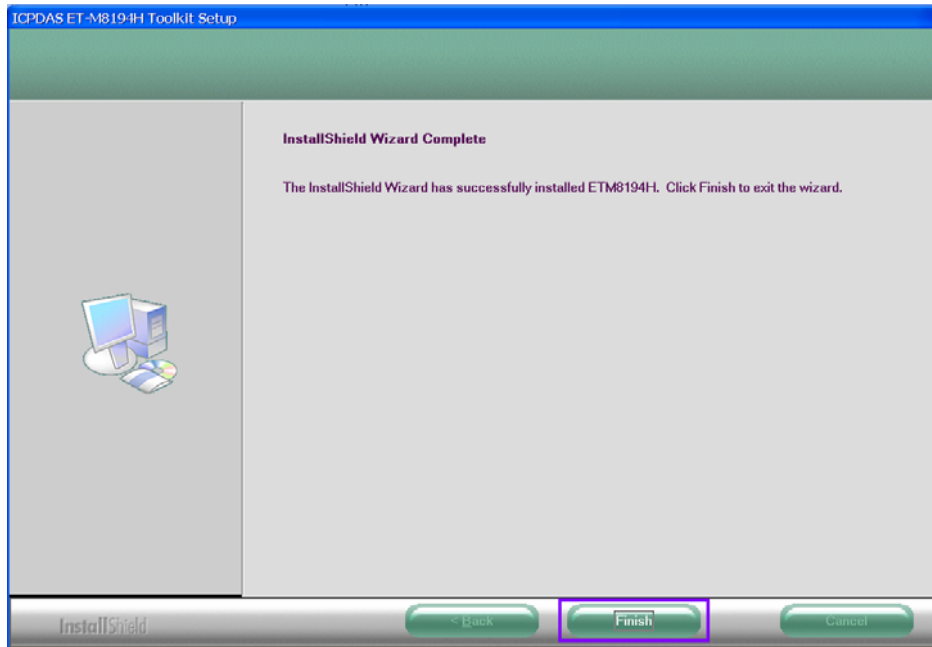
Step 5: Click *Next*. To guarantee consistency of the system it is recommended not to change the destination folder.



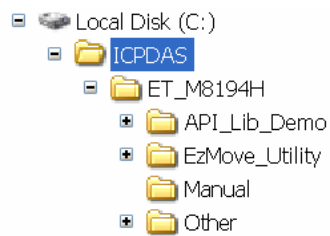
Step 6: Click ***“Install”*** to install the ET-M8194H software package.



Step 7: Click ***Finish***.



After the installation process has finished the required manuals, libraries, utility and examples are located in the `C:\ICPDAS\ET_M8194H` directory.



4 Network Setting

Before it is possible to communicate with the device over the network it is necessary to set the main network configurations. This can either be done via a RS232 connection or via Ethernet.

4.1.1 Setting via RS232

The following steps describe the procedure for setting and reading the network configuration from the ET-M8194H device via COM port.

- Step 1: Connect the ET-M8194H module to the COM port of your PC by following Figure 4. (If your PC does not have a COM port ICPDAS provides a USB to RS232 converter: **I-7560**).

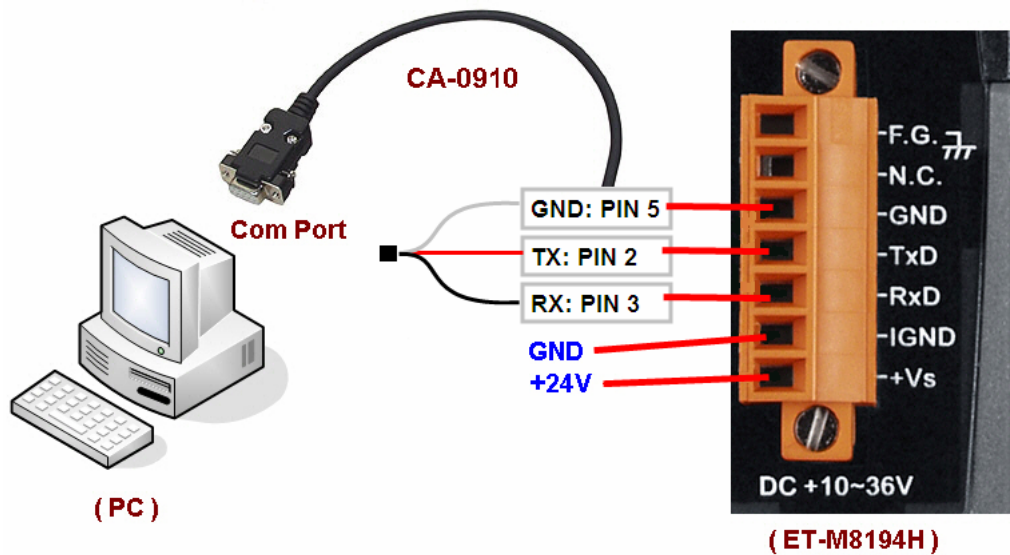
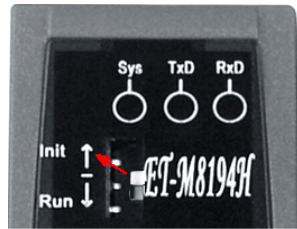


Figure 4: Connecting to COM port

- Step 2: Switch the ET-M8194H device **off**.
- Step 3: Set the dip switch to “**Init**” (Figure 5).



(Dip Switch -- Init)

Figure 5: Set the dip switch to “Init”.

Step 4: Switch the ET-M8149H device **on**.

Step 5: Start the EzMove_Utility.

Step 6: Open the “*Network Setting By COM Port*” dialog window:
Click *Setting* → *ET-M8194H* → *By COM Port* → *Network...*

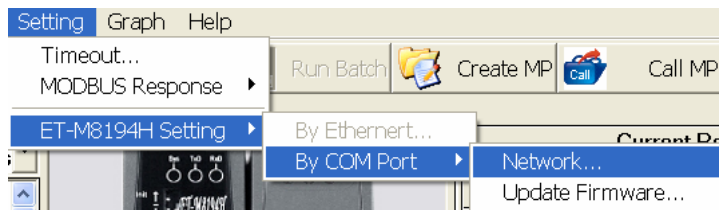
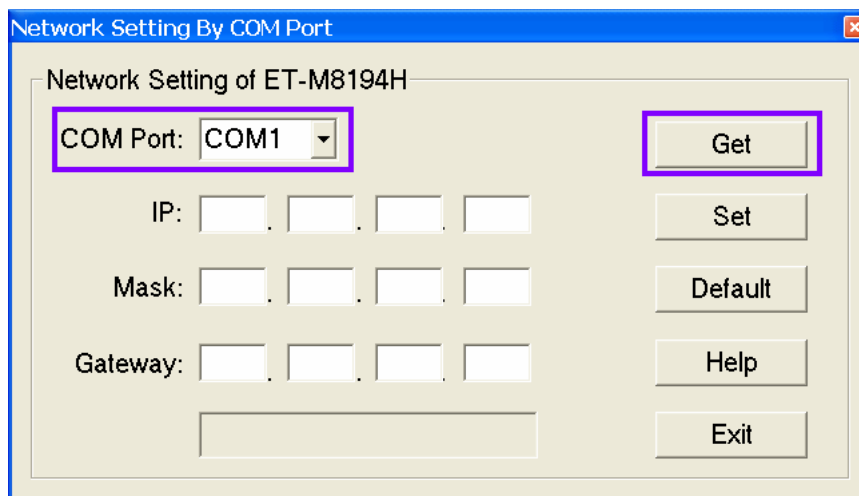


Figure 6: Open network configuration page

Step 7: Select the PC COM port to which the ET-M7184H is connect.



Read ET-M7184H network setting:

Step 8: Click the “*Get*” button to read the device network setting.

Set the ET-M7184H network:

Step 9: Enter IP address, mask and Gateway. Click the “**Set**” button to download the new setting to the device.

Step 10: Switch the ET-M8149H device **off**.

Step 11: Set the dip switch to “**Run**” (Figure 7).



(Dip Switch -- Run)

Figure 7: Set the dip switch to “Run”.

Step 12: Switch on the device. The device can now be accessed via the new network IP address.

4.1.2 Setting via Ethernet

The following steps describe the procedure for setting and reading the network configuration of the ET-M8194H device via Ethernet.

In order to configure the ET-M8194H over the Ethernet it is necessary to know its IP address. If the IP address is unknown consult chapter 4.1.1 for retrieving the IP address.

The ET-M8194H default factory network setting is as follows:

IP:	192.168.0.16
Mask:	255.255.255.0
Gateway:	192.168.0.254

Step 1: Start EzMove_Utility.

Step 2: Open the “**Connect**” dialog window:
Click “**Connect** → **Connect to ET-M8194H...**”.

Step 3: Enter the device IP address and click “**Connect**” (Figure 8).

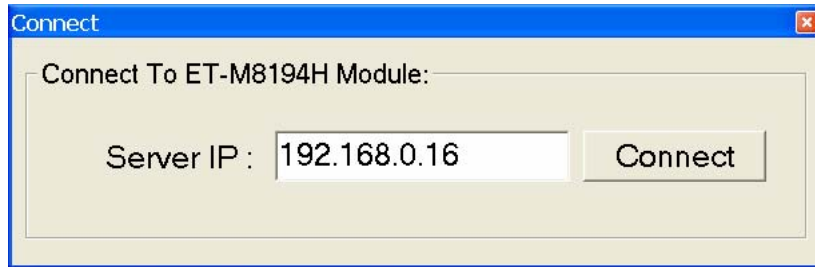


Figure 8: Connect to ET-M8194H

Step 4: Open the “*Setting by Ethernet*” dialog box.

Click *Setting*→*ET-M8194H Setting*→*By Ethernet...*

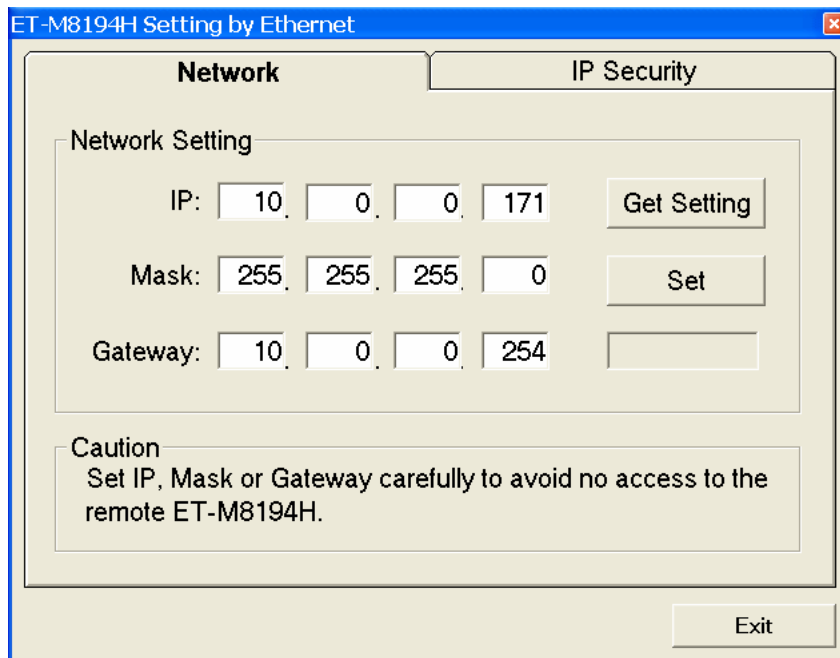


Figure 9: ET-M8194H network setting via Ethernet

Read ET-M7184H network setting:

Step 5: Click the “*Get Setting*” button to read the device network setting.

Set the ET-M7184H network:

Step 6: Enter IP address, mask and Gateway. Click the “*Set*” button to download the new setting to the device.

The new setting will immediately take effect.

5 LED Description



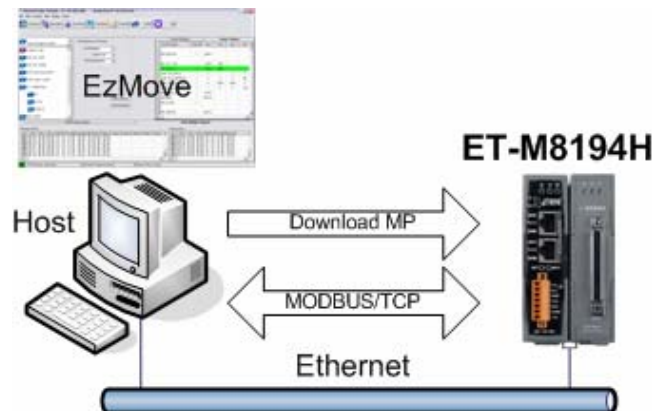
LED	Status	Description
Sys	on	Device is switched on and the firmware is running
	flashing	Device is switched on but the firmware is not running
	off	Device is switched off
Tx	flashing	Data is transmitted by the ET-M8194H via RS232
	off	No data is send by the ET-M8194H via RS232
Rx	flashing	The device is receiving data via RS232
	off	No data is being received
NET	on	Device is connected to Ethernet
	flashing	Data is being transmitted via Ethernet
	off	Device is not connected to the Ethernet
MOD	on	Module i-8094H is plugged into the ET-M8194H device
	flashing	A module different than the i-8094H is plugged into the ET-M8194H device
	off	No module is plugged into the ET-M8194H device

Table 1: LED description

The LED on the i-8094H module is described in the Quick –Start manual for the i-8094H.

6 EzMove Utility

The EzMove utility is written for the Windows operation system with a screen resolution of **1024x768** and a DPI setting of **96 dpi**.



The communication protocol between the master (PC) and the ET-M8194H slave device is Modbus TCP/IP. It is important to notice that according to the Modbus protocol only the master can initiate transactions and the slaves only respond to a request of the master. A slave never initiates a transaction and always responds to or confirms a master request. The user can exchange data with slave in two ways:

- Call the motion APIs of the ET_M8194H_API library. These APIs automatically convert the motion commands into the correct Modbus TCP/IP data format. The programmer therefore does not need to have any knowledge of the protocol and can focus on programming the motion control. The ET_M8194H_API library can only be used on a Window OS platform.
- If the master is running on a non-Windows based OS the software developer has to first convert the motion command with its parameter into the correct Modbus message as described in the manual *ET_M8194H_Manual* before sending it to the slave device.

Purpose of EzMove:

- Getting acquainted with the motion control commands
- Understanding the Modbus data format for the motion control commands
- Downloading and testing of single motion commands or a batch of motion commands.
- Assist the customer to develop its own motion application program.

6.1 Utility Layout

The EzMove utility has basically six interfaces:

- 1) Motion control command list
Motion command tree view lists all the motion commands supported by the ET-M8194H.
- 2) Parameter input interface
The editor can be used for writing
 - Batch programs. A Batch program starts immediately after downloading
 - Macros for simple motion and interrupt service routines. Macros are stored in a nonvolatile memory.
 - Single motion command instruction.
- 4) Modbus data field display
The window displays the actual Modbus command send for each dispatched motion command. The ET-M8194H only accepts Modbus commands and therefore the utility has to convert the motion command into a Modbus data format before it is sent to the ET-M8194H device.
- 5) Current logical and encoder position display
- 6) Graphical display of the motion in a Cartesian coordinate system

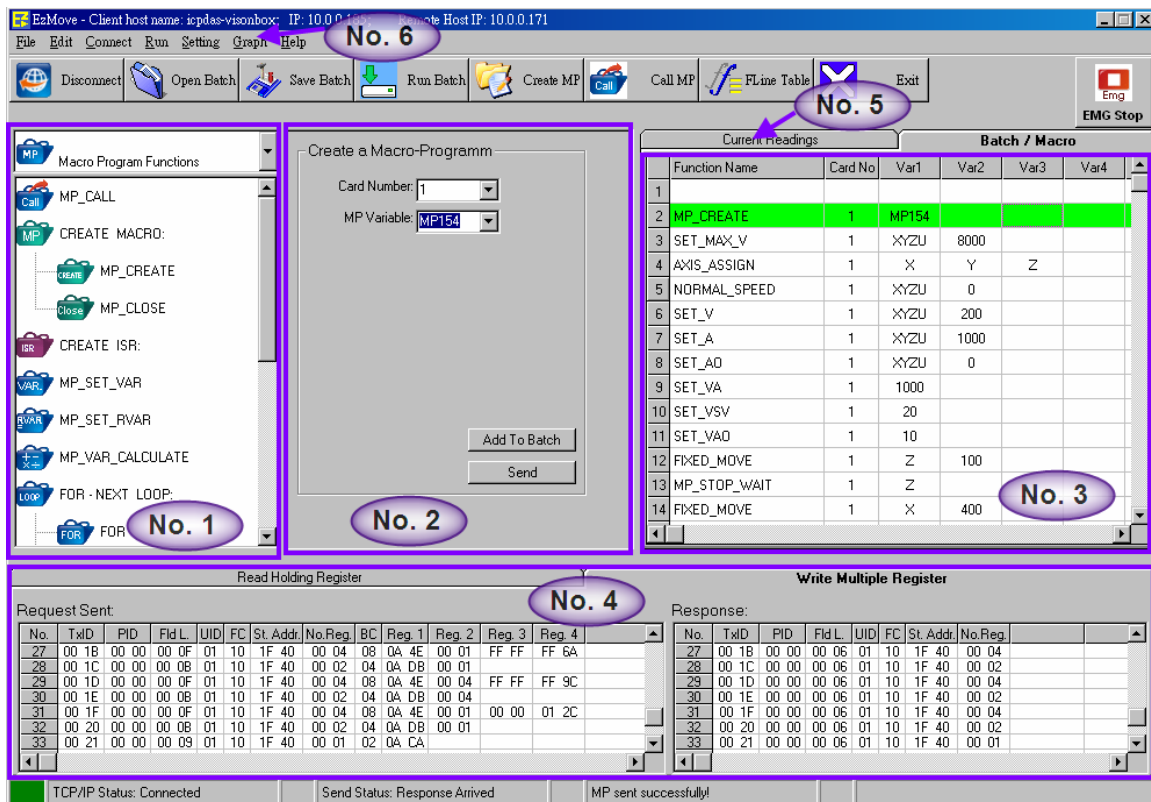


Figure 10: EzMove utility

6.2 Utility Features

- Remotely controlling the motion controller
- Testing and debugging
- Calling commands directly
- Sending several commands at once (batch)
- Creating, saving and changing Batch files
- Writing macro programs
- Displaying the request and response Modbus data format
- Visualizing the motion
- Getting acquainted with the different commands and their properties

Difference between a macro program and a batch file

Macro/ISR Program	Batch file (RTC)
<ul style="list-style-type: none"> • A macro program always start with a MP_CREATE and ends with a MP_CLOSE command. The motion instruction entered between these two commands are part of the same macro. The same applies to the interrupt macros (ISRs) except they start with ISR_CREATE and end with ISR_CLOSE. • Saved on a nonvolatile FRAM • Each macro program has a unique name. • A macro program can call another macro program. A macro can call an unlimited number of macros but in case of a nested macro call the number of nested layers is limited to 6 (Figure 11). 	<ul style="list-style-type: none"> • Consist of sequence of motion instructions. No start and no end command. • Is written to a volatile buffer (Dual Port RAM). Can store a maximum off 30 commands at a time. • Commands will be immediately executed in the sequence they arrive at the buffer. After a command has been executed it will be removed from the buffer. • Batch editor allows the user to quickly write a set of motion commands

Example: Nested MP Call

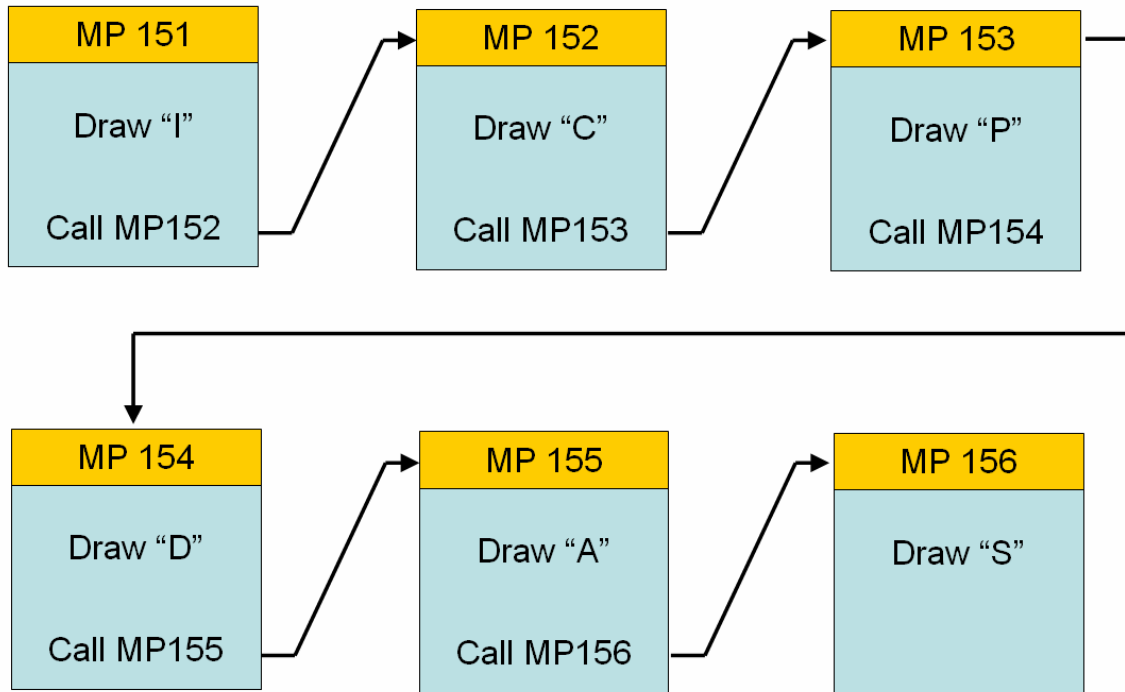


Figure 11: A 6 layer nested macro call

6.3 Editing a macro program

EzMove utility assists the user in programming, debugging and testing simple motion control macros. It can only be used together with the ET-M-8194H motion control module.

The motion macro commands have to be selected from the command list. This prevents the user from editing wrong command names or editing wrong parameters. With EzMove you can create two different macro types:

- Macro program (MP): This macro contains the actual motion control command sequence
- Interrupt service routine macros (ISR): Is being called when an interrupt of the motion card occurs.

These macro files have to be downloaded the ET-M-8194H device. Macros are stored in a nonvolatile memory.

The ET-M-8194H device supports up to 157 macro forms (MP1~MP157) with different sizes (stacks). The size indicates how many command lines a macro form supports. Therefore a macro form has to be selected according to the number of motion commands to be used in a macro. The macro forms are divided according to their sizes into five categories (8/16/32/64/128/512 stacks). A macro form with a stack size of 32 has space for 32 instructions. The user can add up to 157 macro forms to one macro file.

A macro program always start with a **MP_CREATE** and ends with a **MP_CLOSE** command.

Note:

- Always make sure to call **MP_CLOSE** at the end of a macro otherwise all the commands following the macro will be automatically included to the macro till the memory reserved for the macro is full.
- Make sure that not more commands are assigned to macro than it can hold. For example the macro MP1 reserves memory for eight command lines. That means the macro can hold between the **MP_CREATE** and **MP_CLOSE** up to eight commands. If more then eight instructions are downloaded the remaining commands are ignored and get lost.
- Remember that some commands require more space in memory than reserved for one command line. For example the **ARC_CW** and **ARC_CCW** each require memory space for two command lines. This reduces the number of commands a macro can hold. Consult the *ET_M8194H_Manual* or click **FLine Table → Function list for multiple FLines** to look up the number of command lines a command requires.

In addition every parameter input interface (see no 2 in Figure 10) indicate how many lines a command requires:

MP_FLx1 – one line
MP_FLx2 – two lines
MP_FLx3 – three lines
MP_FLx4 – four lines

A macro called by the interrupt service routine is created in the same way as a normal motion control macro described above except that a interrupt macro program always start with a **ISR_CREATE** and ends with a **ISR_CLOSE** command. The 8094H module is designed to store up to 20 macros (ISR1 ~ ISR20). The macros are categorized according to the number of commands they can hold (8/16/32/64 commands).

Procedure to enable an interrupt:

1. call the function
`ENABLE_INT(1)`
2. Assign the interrupt macro to a interrupt factor
`INTFACTOR_ENABLE(1, BYTE axis, BYTE nINT, BYTE ISRNo)`

The interrupt macro will be called when the assign interrupt occurs.

The following section describes how to

- program a macro for drawing a rectangular track in a X-Y plane,
- download the macro program to the nonvolatile memory of the ET-7184H device and
- execute the macro program.

6.3.1 Macro editing method 1

A Command first has to be selected from the command list, and then its parameters have to be set and thereafter it has to be added to the Macro/Batch editor.

Step 1: The motion commands are arranged into 11 categories to facilitate the selection.
Select “**Macro Program Functions**” category.

Step 2: Click the **MP_CREATE** command. Every Macro has to start with the command **MP_CREATE**.

Step 3: Select **MP93** as macro form. This form provides lines for up to 32 commands.

Step 4: Click the “Add to Batch” button to add the new command to the **Macro/Batch** editor

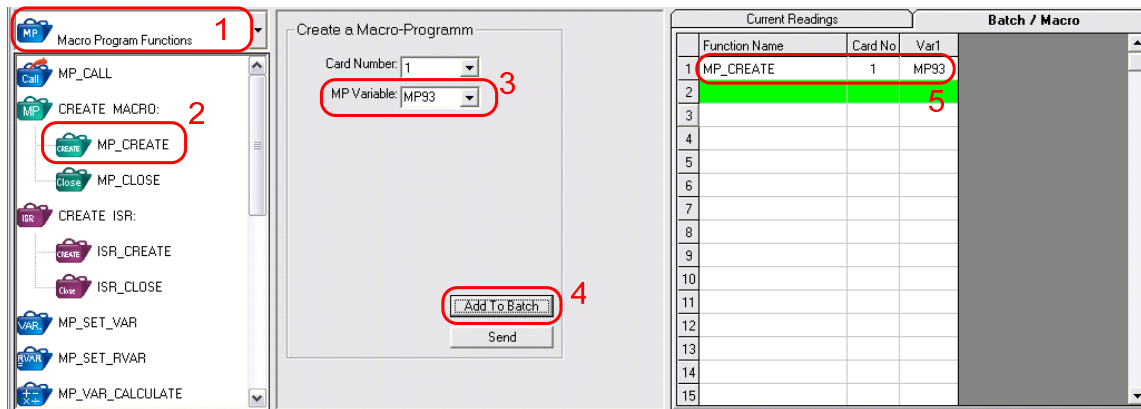


Figure 12: Editing commands (method 1)

6.3.2 Macro editing method 2

Command and their parameter can be directly selected from a dropdown list in the Macro/Batch editor.

Step 1: Click with the mouse in the first column of Macro/Batch editor the line directly beneath the MP_CREATE of command. A dropdown list with all the commands supported by the ET-7184H device pops up. Select **SET_MAX_V**.

Step 2: Now click the column of the first variable (Var1). Select **XYZU**. This means the velocity resolution which you are going to set applies to all axes.

Step 3: Click the column of the second variable (Var2). Enter the value **8000**.

Step 4: Click the first column of the next line to select the next command.

Current Readings		Batch / Macro		
	Function Name	Card No	Var1	Var2
1	MP_CREATE	1	MP93	
2	SET_MAX_V	1	XYZU	8000
3	SET_MAX_V			
4	SET_NHOME			
5	SET_OUT			
6	SET_PRESET			
7	SET_PULSE			
8	SET_PULSE_MODE			
9	SET_SLMT			
10	SET_SV			

Figure 13: Editing commands (method 2)

Step 5: Continue entering the following macro either using method 1 or 2.

	Function Name	Card No	Var1	Var2
1	MP_CREATE	1	MP93	
2	SET_MAX_V	1	XYZU	8000
3	NORMAL_SPEED	1	XYZU	0
4	SET_V	1	XYZ	200
5	SET_A	1	XYZ	1000
6	SET_SV	1	XYZ	20
7	SET_AO	1	XYZ	0
8	SET_LP	1	XYZU	0
9	FIXED_MOVE	1	Z	100
10	MP_STOP_WAIT	1	Z	
11	MP_TIMER	1	2000	
12	FIXED_MOVE	1	XY	100
13	MP_STOP_WAIT	1	XY	
14	FIXED_MOVE	1	Z	-100
15	MP_STOP_WAIT	1	Z	
16	FIXED_MOVE	1	Y	800
17	MP_STOP_WAIT	1	Y	
18	FIXED_MOVE	1	X	800
19	MP_STOP_WAIT	1	X	
20	FIXED_MOVE	1	Y	-800
21	MP_STOP_WAIT	1	Y	
22	FIXED_MOVE	1	X	-800
23	MP_STOP_WAIT	1	X	
24	MP_CLOSE	1		

This macro has already been created by ICPDAS and has been saved as *MP93_Square.dat* to the following directory:

C:\ICPDAS\ET_M8194H\EzMove_Utility\Demo

Open this file with the EzMove utility by clicking the “Open Batch” button

Step 6: Save the macro to file: Click “Save Batch” and enter a name for the file. The file extension is **.dat**.

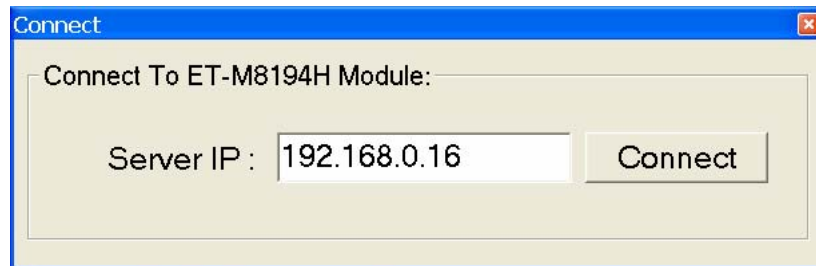
6.4 Downloading macro programs

Step 1: Power the ET-M8194H device on.

Make sure the dip switch is set to “Run” before you power on the device.

Step 2: Connect the EzMove utility to the ET-M8194H device.

Open the “Connect” dialog window (click “**Connect** → **Connect to ET-M8194H...**”) and enter the device IP address and click “**Connect**”. The status bar displays a green icon when the utility is connected to the ET-M8194H device.



Step 3: Click the “Run Batch” button in the toolbar.



The macro “**MP93**” is now downloaded to a nonvolatile memory.

Batch/Macro editor can contain one or more macros and also batch commands. All commands listed in the Batch/Macro editor are downloaded sequential from top to bottom to a volatile buffer (Dual Port RAM). Macro and interrupt macros are then read from the buffer and saved to a nonvolatile memory. The remaining commands are immediately executed one by one.

Note: To have a clear structure and to make debugging easier it is suggested to edit and download only one macro at a time. Save each macro created with the EzMove to a different file and download each macro separately.

6.5 Executing macro programs

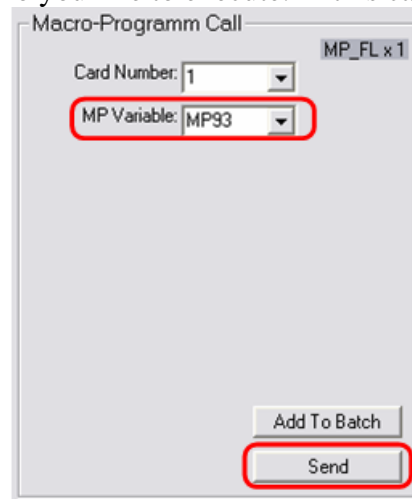
The utility provides two method of calling a macro:

Method 1: Calling a single macro

Step 1: Click the “Call MP” button in the toolbar.



Step 2: Select the macro you like to execute. In this case it is macro “MP93”



Step 3: Click “Send” to run the macro. The macro will now be executed.

Method 2: Calling multiple macros

Step 1: Click the “Call MP” button in the toolbar.



Step 2: Select a macro you like to execute then click “Add to Batch”. In this case the macro is not being called immediately but added to the Batch/Macro editor. Now select the next macro to the editor. In Figure 14 seven macro calls are added to the editor.

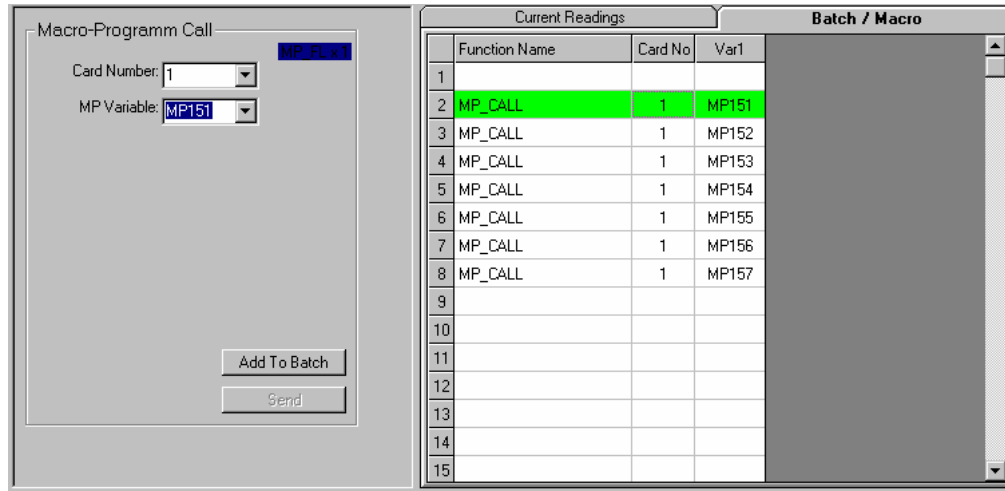
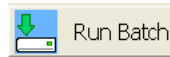


Figure 14: Multiple macro call

Step 3: Click the “Run Batch” button in the toolbar.



The macros will now be executed one by one in the same order as they are listed batch/macro editor.

6.6 Displaying motion pathway

EzMove allows the user to display the logical or encoder path while the ET-M8194H is executing motion commands.

Step 1: Click “**Graph**” in the menu bar. A graph window with a three axis coordinate system pops up.

Step 2: Enter the range for each axis (units are pulses) then click “**Set Axis Range**”.

Step 3: Select whether to display the logical or encoder position.

Step 4: Hide the window (click “Hide Window”).

Coordinate-Axes Range

0 < X <

0 < Y <

0 < Z <

☒ Logic Position

☐ Encoder Position

Step 5: Click the “**Current Readings**” tab. Enter 100 milliseconds for the polling time interval and click “**Enable**”. The EzMove will poll every 100 milliseconds the logical and encoder position.

Current Readings Batch / Macro

Logical Position

x-axis:

y-axis:

z-axis:

u-axis:

Encoder Position

Velocity

☐ Get Velocity

Get Current State Values

Enable / Disable Current State Timer

Time Interval: [ms]

Card No.:

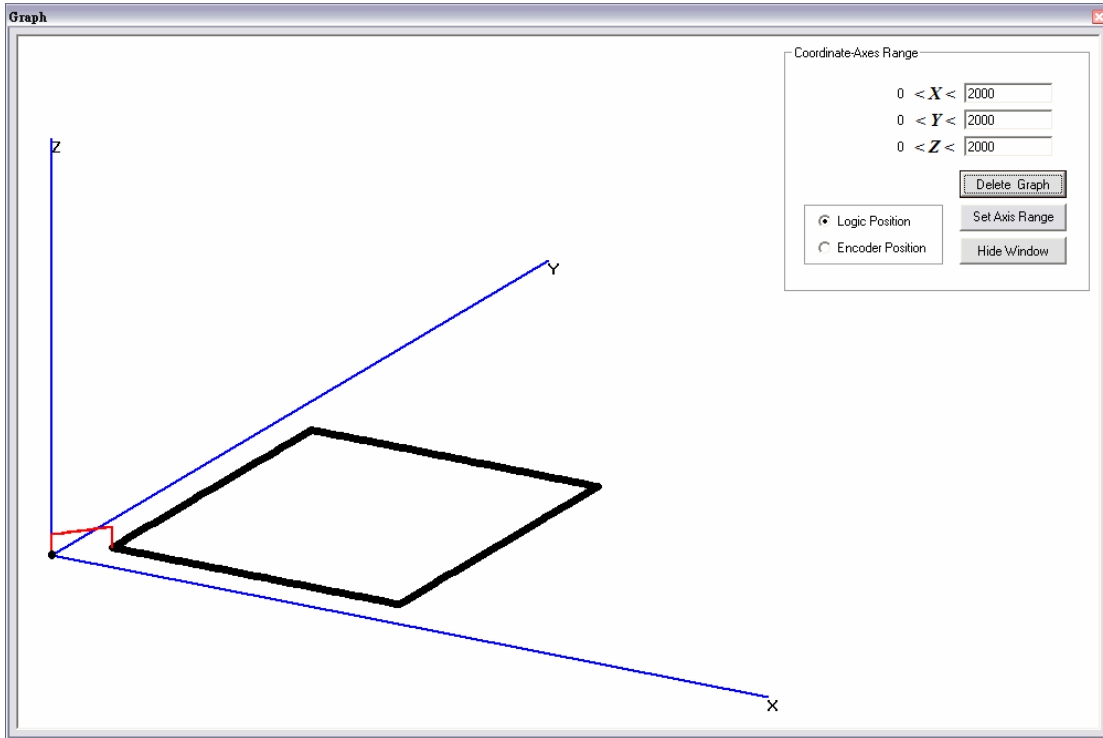
Step 6: Set both the logical and encoder position to zero by clicking the “**Zero LP**” and “**Zero EP**” buttons.

Now call and execute a macro:

Step 7: Click the “**Call MP**” button in the toolbar and select a macro of your choice. Select the macro you like to execute. In this example macro “MP93” will be called as this is the only macro we have so far downloaded.

Step 8: Click “Send” to run the macro. The macro will now be executed.

Step 9: Click again the “**Graph**” menu in the menu bar. The graph shows the current position in the coordinate system.



6.7 Sending single commands

Once a batch or macro has been created individual commands can be selected from the editor and send to the ET-M8194H device for immediate execution. This feature is only supported by commands which name do not start with “MP”. This feature enables the programmer to debug his batch/macro by sending the commands one by one.

Example:

Open the **MP93_Square.dat** in the directory
C:\ICPDAS\ET_M8194H\EzMove_Utility\Demo

The following steps refer to Figure 15:

- Step 1: Click **SET_MAX_V** and then “**Send**”.
- Step 2: Click **NORMAL_SPEED** and then “**Send**”.
- Step 3: Click **SET_V** and then “**Send**”.

After a motion command like FIXED_MOVE has been send you can wait until the command has been executed and check whether the motion system has really reached its target position. Then you can dispatch the next motion command.

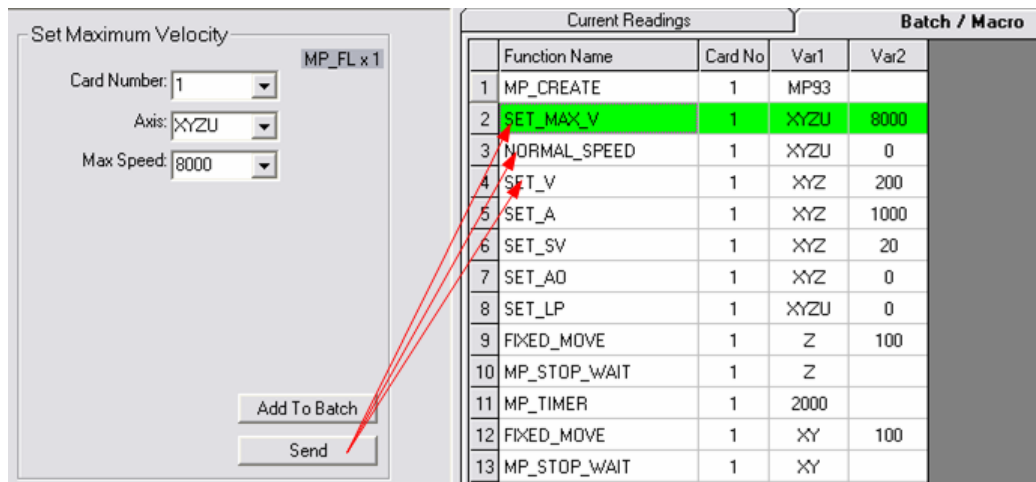


Figure 15: Sending and executing commands one by one

6.8 Modbus message structure

The tab window at the bottom of the EzMove utility shows for each dispatched command the actual data field send to the ET-M8194H device and the corresponding response data field of the ET-M8194H device.

6.8.1 Reading holding register

This Modbus data format is being used for polling data such as reading velocity, acceleration, logical and encoder position.

Read Holding Register										Write Multiple Register																	
Request Sent										Response:																	
No.	TxD	PID	Fld L	UID	FC	St. Addr	No. Reg.			No.	TxD	PID	Fld L	UID	FC	BC	Reg. 1	Reg. 2	Reg. 3	Reg. 4	Reg. 5	Reg. 6	Reg. 7	Reg. 8	Reg. 9	F	
756	07 2F	00 00	00 06	01	03	00 5A	00 10			756	07 2F	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	F
757	07 30	00 00	00 06	01	03	00 5A	00 10			757	07 30	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	
758	07 31	00 00	00 06	01	03	00 5A	00 10			758	07 31	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	
759	07 32	00 00	00 06	01	03	00 5A	00 10			759	07 32	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	
760	07 33	00 00	00 06	01	03	00 5A	00 10			760	07 33	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	
761	07 34	00 00	00 06	01	03	00 5A	00 10			761	07 34	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	
762	07 35	00 00	00 06	01	03	00 5A	00 10			762	07 35	00 00	00 23	01	03	20	00 00	00 64	00 00	00 64	00 00	00 00	00 00	00 00	00 00	00 00	

The data field of the following Table 2 reads the logical and encoder position of the four axes. As each position is stored in two registers altogether 16 register have to be read.

Request			Response		
Field Name	Length (Bytes)	(Hex) Example	Field Name	Length (Bytes)	(Hex) Example
Transaction ID:	2	07 2F	Transaction ID:	2	07 2F
Protocol ID:	2	00 00	Protocol ID:	2	00 00
Field Length:	2	00 06	Field Length:	2	00 23
Unit ID	1	01 (cardNo)	Unit ID	1	01 (cardNo)
Function Code:	1	03	Function Code:	1	03
Address of first input Register:	2	00 5A (90)	Byte Count:	1	20
Number of Register:	2	00 10 (16)	Input Register (00 5A):	2	00 00 (MSW LP_X)
			Input Register (00 5B):	2	00 64 (LSW LP_X)
			Input Register (00 5C):	2	00 00 (MSW LP_Y)
			Input Register (00 5D):	2	00 64 (LSW LP_Y)
			...	2
			Input Register (00 69):	2	...

MSW- most significant word

LSW – least significant word

Table 2: Reading logical and encoder position

Function	Holding Register	Description
----------	------------------	-------------

	Address	
LP_X	90 (0x005A)	Logical position of X-axis. It takes two registers. The MSW is located at address 90.
LP_Y	92 (0x005C)	Logical position of Y-axis. It takes two registers. The MSW is located at address 92.
LP_Z	94 (0x005E)	Logical position of Z-axis. It takes two registers. The MSW is located at address 94.
LP_U	96 (0x0060)	Logical position of U-axis. It takes two registers. The MSW is located at address 96.
EP_X	98 (0x0062)	Encoder feedback position of X-axis. It takes two registers. The MSW is located at address 98.
EP_Y	100 (0x0064)	Encoder feedback position of Y-axis. It takes two registers. The MSW is located at address 100.
EP_Z	102 (0x0066)	Encoder feedback position of Z-axis. It takes two registers. The MSW is located at address 102.
EP_U	104 (0x0068)	Encoder feedback position of U-axis. It takes two registers. The MSW is located at address 104.

Table 3: Holding register addresses for logical and encoder positions

6.8.2 Writing multiple register

This format is being used for writing data to the remote device e.g. set velocity, acceleration, software limit etc. The majority of the motion functions sends data to the ET-M8194H device and therefore makes use of the Modbus function code 16.

Read Holding Register														Write Multiple Register											
Request Sent:														Response:											
No.	TxD	PID	FidL	UID	FC	St. Addr	No.Reg	BC	Reg. 1	Reg. 2	Reg. 3	Reg. 4		No.	TxD	PID	FidL	UID	FC	St. Addr	No.Reg				
102	07 47	00 00	00 0F	01 10	1F 40	00 04	08	0A 4E	00 01	00 00	03 20			102	07 47	00 00	00 06	01 10	1F 40	00 04					
103	07 48	00 00	00 0B	01 10	1F 40	00 02	04	0A DB	00 01					103	07 48	00 00	00 06	01 10	1F 40	00 02					
104	07 49	00 00	00 0F	01 10	1F 40	00 04	08	0A 4E	00 02	FF FF	FC E0			104	07 49	00 00	00 06	01 10	1F 40	00 04					
105	07 4A	00 00	00 0B	01 10	1F 40	00 02	04	0A DB	00 02					105	07 4A	00 00	00 06	01 10	1F 40	00 02					
106	07 4B	00 00	00 0F	01 10	1F 40	00 04	08	0A 4E	00 01	FF FF	FC E0			106	07 4B	00 00	00 06	01 10	1F 40	00 04					
107	07 4C	00 00	00 0B	01 10	1F 40	00 02	04	0A DB	00 01					107	07 4C	00 00	00 06	01 10	1F 40	00 02					
108	07 4D	00 00	00 09	01 10	1F 40	00 01	02	0A CA						108	07 4D	00 00	00 06	01 10	1F 40	00 01					

The following example (Table 4) shows the data field of the function call

	Function Name	Card No	Var1	Var2
1				
2	SET_PULSE	1	X	800
3				

SET_PULSE (BYTE cardNo, BYTE axis, DWORD data);

With

cardNo = 1
axis = 1 (x-axis)
data = 800

Request			Response		
Field Name	Length (Bytes)	(Hex) Example	Field Name	Length (Bytes)	(Hex) Example
Transaction ID:	2	07 47	Transaction ID:	2	07 47
Protocol ID:	2	00 00	Protocol ID:	2	00 00
Field Length:	2	00 0F	Field Length:	2	00 06
Unit ID	1	01 (cardNo)	Unit ID	1	01
Function Code:	1	10	Function Code:	1	10
Register Starting Address:	2	1F 40 (8000)	Register Starting Address:	2	1F 40 (8000)
Number of Register to be written:	2	00 04	Number of Register changed:	2	00 04
Number of bytes	1	08			
Value of Register (1F 40)	2	0A 4E (SET_PULSE())			
Value of Register (1F 41)	2	00 01 (axis)			
Value of Register (1F 42)	2	00 00 (MSW data)			
Value of Register (1F 42)	2	03 20 (LSW data)			

MSW- most significant word

LSW – least significant word

Table 4: SET_PULSE function call

7 Libraries

ICPDAS provides libraries for the software developer to write his own motion control program and also libraries for creating and downloading of macros. Libraries for Visual Basic 6.0, Visual C++ and Borland C++ Builder are available. These libraries are only for the Windows platform and are especially useful for programmers who are not familiar with the Modbus protocol as they manage the communication with the remote ET-M8194H device. No knowledge about the Modbus data frame is required. Non-Windows programmers have to understand the Modbus protocol and have to write the Modbus communication part themselves. Consult the ET_M8194H manual for the data field definition of each command.

Libraries are located in the following directory:
C:\ICPDAS\ET_M8194H\API_Lib_Demo\Lib

Demo programs show how to implement and use the libraries. They are located at the following directory:
C:\ICPDAS\ET_M8194H\API_Lib_Demo\Demo

