

ET-M8196F
Motion Control Module
User Manual
(Version 1.0)

API Library



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 2007-2016 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.

Contents

1	Preface	6
1.1	Function Overview	7
1.2	Command Flow Chart.....	13
1.2.1	Axis Digital I/O Setting	14
1.2.2	Motion Control Initiation	15
1.2.3	Single Axis Motion Control.....	16
1.2.4	Interpolation Motion Control	17
1.2.5	Initiate Simple Interpolation Motion	18
1.2.6	Initiate Continuous Interpolation Motion.....	19
2	Ethernet Connection Settings	20
2.1	TCP/IP Communication.....	20
2.1.1	etm96_connectW	20
2.1.2	etm96_connectA.....	21
2.1.3	etm96_disconnect	22
3	Motion Commands	23
3.1	Axis Digital I/O Function	23
3.1.1	etm96_set_servo_on	23
3.1.2	etm96_set_erc	24
3.1.3	etm96_set_alarm_reset.....	25
3.1.4	etm96_set_alarm.....	26
3.1.5	etm96_set_inp	27
3.1.6	etm96_set_ready	29
3.1.7	etm96_set_limit.....	30
3.1.8	etm96_get_io_status	32
3.1.9	etm96_get_io_status_all	34
3.1.10	etm96_get_servo_on_status	36
3.1.11	etm96_get_erc_status	37
3.1.12	etm96_get_alarm_reset_status	38
3.2	Motion Control Pulse Setting Function	39
3.2.1	etm96_set_pls_cfg.....	39
3.2.2	etm96_set_enc_cfg.....	41
3.2.3	etm96_set_cmdcounter	42

3.2.4	etm96_get_cmdcounter	43
3.2.5	etm96_get_cmdcounter_all	44
3.2.6	etm96_set_enccounter	45
3.2.7	etm96_get_enccounter	46
3.2.8	etm96_get_enccounter_all.....	47
3.2.9	etm96_set_vring_counter	48
3.2.10	etm96_get_vring_counter	49
3.2.11	etm96_disable_vring_counter	50
4	Automatic Home Search	51
4.1	Automatic Home Search Configuration.....	51
4.1.1	etm96_set_home_cfg	51
4.2	Automatic Home Execution.....	54
4.2.1	etm96_home_start	54
5	Motion Control Instructions.....	55
5.1	Read Motion Status	55
5.1.1	etm96_get_motion_done.....	55
5.1.2	etm96_get_motion_done_all.....	57
5.1.3	etm96_get_speed	59
5.1.4	etm96_get_speed_all	60
5.1.5	etm96_get_acc.....	61
5.1.6	etm96_get_acc_all.....	62
5.2	Single Axis Motion Commands.....	63
5.2.1	Introduction	63
5.2.2	etm96_t_move.....	64
5.2.3	etm96_abs_t_move	65
5.2.4	etm96_s_move	66
5.2.5	etm96_abs_s_move.....	67
5.2.6	etm96_velocity_move	68
5.3	Two Axes Linear Interpolation Commands	69
5.3.1	etm96_t_line2_move.....	69
5.3.2	etm96_abs_t_line2_move	71
5.3.3	etm96_s_line2_move	73
5.3.4	etm96_abs_s_line2_move.....	75

5.4	Three Axis Linear Interpolation Commands.....	77
5.4.1	etm96_t_line3_move.....	77
5.4.2	etm96_abs_t_line3_move	79
5.4.3	etm96_s_line3_move	81
5.4.4	etm96_abs_s_line3_move.....	83
5.5	Multi-Dimensional Linear Interpolation Commands	85
5.5.1	etm96_t_lines_move	85
5.5.2	etm96_abs_t_lines_move.....	87
5.5.3	etm96_s_lines_move.....	89
5.5.4	etm96_abs_s_lines_move	91
5.6	Two Dimensional Circular Interpolation Functions.....	93
5.6.1	etm96_t_arc2_move.....	93
5.6.2	etm96_abs_t_arc2_move	95
5.7	Three Dimensional Helical Interpolation Functions.....	97
5.7.1	etm96_t_helical_move	97
5.7.2	etm96_abs_t_helical_move.....	100
5.8	Continuous Interpolation Functions.....	103
5.8.1	etm96_set_conti_interp_cfg.....	103
5.8.2	etm96_send_conti_packet	108
5.8.3	etm96_clear_conti_packet	110
5.8.4	etm96_get_conti_packet_size	111
5.9	Motion Stop Functions.....	112
5.9.1	etm96_stop_move	112
5.9.2	etm96_set_softlimit.....	113
5.9.3	etm96_set_softlimit_disable	114
5.10	Multi-Axis Hold/Release Functions.....	115
5.10.1	etm96_drv_hold.....	115
5.10.2	etm96_drv_start.....	116
6	Other Functions	117
6.1	Compare Function	117
6.1.1	etm96_set_compare_trig_cfg.....	117
6.2	Latch Function	120
6.2.1	etm96_set_latch_cfg.....	120

6.2.2	etm96_get_latch.....	121
7	General Purpose IO and Remote FRnet IO Functions.....	122
7.1	General Purpose IO	122
7.1.1	etm96_set_general_do.....	122
7.1.2	etm96_get_general_do.....	123
7.1.3	etm96_set_all_general_do.....	124
7.1.4	etm96_get_all_general_do.....	125
7.1.5	etm96_get_general_di.....	126
7.1.6	etm96_get_all_general_di.....	127
7.2	Remote FRnet IO	128
7.2.1	etm96_set_frnet_group_do	128
7.2.2	etm96_get_frnet_group_do	129
7.2.3	etm96_get_frnet_group_do_all	130
7.2.4	etm96_scan_frnet_di.....	131
7.2.5	etm96_get_frnet_group_di	132
7.2.6	etm96_get_frnet_group_di_all.....	133
8	Hardware and Software Version Functions	134
8.1	ET-M8196F Version	134
8.1.1	etm96_get_firmware_version	134
8.2	I-8196F Version.....	135
8.2.1	etm96_get_i8196f_version.....	135
8.3	DLL Version.....	136
8.3.1	etm96_get_dll_pathW.....	136
8.3.2	etm96_get_dll_pathA	137
8.3.3	etm96_get_dll_version	138
9	Appendix.....	139
9.1	Variable data type definition.....	139
9.2	Function Error Code	140

1 Preface

The ET-M8196F is a compact remote motion control device which uses Modbus TCP as its communication protocol. The ET-M8196F acts as a server in a Modbus TCP network and supports all standard Modbus function codes defined by the Modbus TCP protocol. Nowadays many PCs have got limited PCI slots; therefore the ET-M8196F can be used to replace PCI motion control cards. The ET-M8196F has got two Ethernet ports which allow daisy chaining.

In addition the ET-M8196F acts as an FRnet master and can control up to 128 digital outputs and 128 digital inputs. FRnet is a two-wire serial bus and has a scan interval of 0.72 ms and it is specifically designed for easy and cost effective wiring. ICPDAS provides a large range of FRnet I/O terminal boards and modules.

1.1 Function Overview

Libraries are provided for the ET-M8196F which allows the user to remotely control the device without having to understand the Modbus communication. The libraries (DLL, Dynamic-link library) support all Windows operation systems currently on the market. Examples are provided to show how to use the libraries in the different integrated development environment (IDE), like Visual Studio, Delphi, Borland, LabVIEW, Visual Basic 6.0, etc.

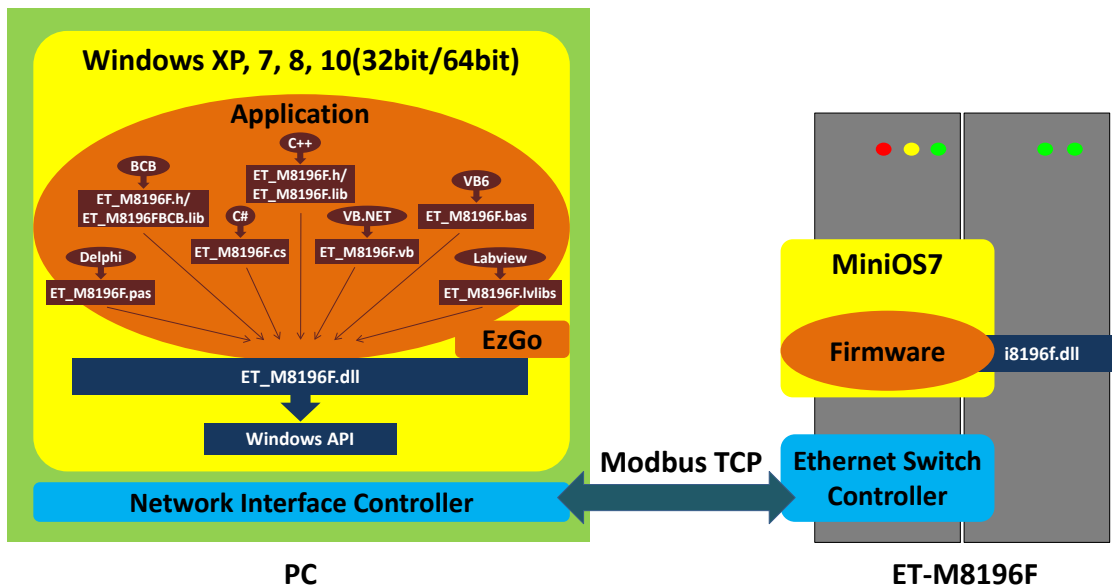


Figure 1: Software application

The following table gives a short overview of the DLL functions supported by ET-M8196F:

Function	Description
TCP/IP Communication	
etm96_connectW	Establish an Ethernet connection between PC and ET-M8196F. It is important to first create a connection with ET-M8196F before calling any other command.
etm96_connectA	Establish an Ethernet connection between PC and ET-M8196F. It is important to first create a connection with ET-M8196F before calling any other command.
etm96_disconnect	Close the established Ethernet connection and release internal memory assigned for the connection.
Axis Digital I/O Function	
etm96_set_servo_on	Set the SRV_ON channel output signal to enable/disable the servo drive for pulse command input.
etm96_set_erc	The ERC signal sets the servo drive deflection counter to zero.
etm96_set_alarm_reset	Reset the servo drive alarm signal (ALM_RST).
etm96_set_alarm	Enable/Disable the alarm function and set the active level of

	the servo alarm signal (ALARM). If the servo drive encounters an abnormality while driving, it sends a signal to the ALARM channel of the ET-M8196F. If the servo alarm function of the ET-M8196F has been enabled and the servo alarm signal is active then no pulses will be outputted.
etm96_set_inp	Enable/Disable the in-position function and set the active level of the servo in-position signal (INP). In general, when servo drive is set to position mode (P mode), the servo issues a (INP) pulse signal to controller when movement get into position. If the servo in-position function of the ET-M8196F has been enabled, then the controller waits until the in-position signal of the servo has been triggered before continuing to execute the next motion command.
etm96_set_ready	Enables/Disables the ET-M8196F to check for the servo drive “ready” state and sets the active level of the servo RDY signal.
etm96_set_limit	Sets the logic levels of the LMT+ and LMT- channels and the stop mode. The hardware limit signals (LMT+, LMT-) are used for stopping the pulse output when the limit switches are triggered. The hardware limit switches are being used for mechanical protection of the system. If the positive switch (LMT+) is being triggered while the movement is in positive direction the motion stops according to the set stop mode. On the other hand the motion will stop when moving in negative direction and the negative limit switch (LMT-) is active.
etm96_get_io_status	Reads the current axis I/O channels state.
etm96_get_io_status_all	Reads the current I/O channels state of all six axes.
etm96_get_servo_on_status	Reads the output signal of the SRV_ON channel. The SRV_ON channel determines whether the servo drive has been enabled to control the motor. The SRV_ON signal is set by “etm96_set_servo_on()”.
etm96_get_erc_status	Reads the ERC output signal. The ERC signal clears the deviation counter of the servo drive and is set by calling “etm96_set_erc()”.
etm96_get_alarm_reset_status	Reads the status of the ALM_RST output channel. The ALM_RST signal resets the alarm state of the servo drive. “etm96_set_alarm_reset()” sets the output signal.
Motion Control Pulse Setting Function	
etm96_set_pls_cfg	Set the pulse output mode for each axis.
etm96_set_enc_cfg	Set the parameters of the encoder pulse input.
etm96_set_cmdcounter	Set the command counter (position command) value.
etm96_get_cmdcounter	Get the current command counter (position command) value.
etm96_get_cmdcounter_all	Get the current command counter (position command) for all six axes.
etm96_set_enccounter	Set the encoder counter value.
etm96_get_enccounter	Get the current encoder counter value.
etm96_get_enccounter_all	Get the current encoder counter value for all six axes.
etm96_set_vring_counter	Set the maximum ring counter position for both the encoder and commanded position counter.
etm96_get_vring_counter	Read the maximum ring counter setting.
etm96_disable_vring_counter	Disable the ring counter setting.
Automatic Home Search Configuration	
etm96_set_home_cfg	Set automatic home search parameters.
Automatic Home Execution	

etm96_home_start	Start searching for the home position.
Read Motion Status	
etm96_get_motion_done	Read the current motion status of the axis.
etm96_get_motion_done_all	Read the current motion status of all six axes.
etm96_get_speed	Get the current axis speed.
etm96_get_speed_all	Get the current speed of all six axes.
etm96_get_acc	Get the current axis acceleration.
etm96_get_acc_all	Get the current acceleration of all six axes.
Single Axis Motion Commands	
etm96_t_move	Execute a single axis, relative position motion command with a trapezoidal velocity profile (T-curve). The etm96_t_move instruction moves the axis the specified travel distance from the current position.
etm96_abs_t_move	Execute a single axis, absolute position motion command with a trapezoidal velocity profile (T-curve). The etm96_abs_t_move instruction moves the axis to a specified absolute target position. You can execute this instruction even if home is not defined.
etm96_s_move	Execute a single axis motion command with an S-curve velocity profile. This command initiates a relative motion. When received, the selected axis will move, with the predefined acceleration and velocity, to a relative position from the current position.
etm96_abs_s_move	Execute a single axis, absolute position motion command with an S-curve velocity profile.
etm96_velocity_move	Starts a single axis continues pulse driving. Once the axis has reached the driving speed it will indefinitely output pulses at a constant rate until a stop command has been encountered.
Two Axes Linear Interpolation Commands	
etm96_t_line2_move	Executes a two axes relative distance linear interpolation motion command with a T-curve velocity profile. The etm96_t_line2_move instruction performs linear interpolation for two axes. The target position is specified as a relative position.
etm96_abs_t_line2_move	Executes a two axes absolute position interpolation motion command with a T-Curve velocity profile.
etm96_s_line2_move	Executes a two axes relative distance interpolation motion command with an S-Curve velocity profile.
etm96_abs_s_line2_move	Executes a two axis absolute position interpolation motion command with an S-Curve velocity profile.
Three Axis Linear Interpolation Commands	
etm96_t_line3_move	Executes a three axes linear interpolation motion command with a T-curve velocity profile. The target position is specified as a relative distance to the current position.
etm96_abs_t_line3_move	Executes a three axes linear interpolation motion command with a T-curve velocity profile. This instruction moves the axes to an absolute position relative to (0) zero.
etm96_s_line3_move	Executes a three axes linear interpolation motion command with an S-curve velocity profile. The etm96_s_line3_move instruction moves the axes the specified travel distance from the current position.
etm96_abs_s_line3_move	Executes a three axes absolute position interpolation motion command with an S-curve velocity profile.
Multi-Dimensional Linear Interpolation Commands	

etm96_t_lines_move	Executes a multidimensional relative position motion command. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.
etm96_abs_t_lines_move	Executes a multidimensional absolute position motion command. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.
etm96_s_lines_move	Executes a multidimensional relative position motion command with a s-curve velocity profile. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.
etm96_abs_s_lines_move	Executes a multidimensional absolute position motion command using the s-curve velocity profile. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.
Two Dimensional Circular Interpolation Functions	
etm96_t_arc2_move	Performs circular interpolation for two axes with a T-curve velocity profile. The center and end position are specified relative to the current position.
etm96_abs_t_arc2_move	Executes a two axes circular interpolation motion command with a T-curve velocity profile. The center and target position are specified in absolute position.
Three Dimensional Helical Interpolation Functions	
etm96_t_helical_move	Executes a three dimensional helical (screw like) movement. The movement is a composition of a rotation by an angle about a helical axis with a translation distance along this axis. The circle center position is the relative distance from the current position.
etm96_abs_t_helical_move	Executes a three dimensional helical (screw like) movement. The movement is a composition of a rotation by an angle about a helical axis with a translation distance along this axis. The circle center position is an absolute position.
Continuous Interpolation Functions	
etm96_set_conti_interp_cfg	Assigns axes to an interpolation group and sets the axes group to continuous interpolation mode (see Figure 5, Figure 6 and Figure 7). Once the group has switch to continuous mode, all the arriving commands are being treated as continuous interpolation commands. In continuous interpolation mode more than one command can be sent at a time. If a new command is being sent while the previous commands is still executing, then the arriving command will first be written to the internal FIFO buffer and starts to executed once the running command has finished. Up to 5000 commands can be stored in the FIFO buffer.
etm96_send_conti_packet	Transmits a TCP/IP packet with multiple continuous interpolation commands to the remote ET-M8196F. Select the packet mode in order to add multiple commands in one TCP/IP packet.
etm96_clear_conti_packet	Clears all the continuous interpolation commands which have been written to the TCP/IP packet so far for sending to the ET-M8196F.
etm96_get_conti_packet_size	Clears all the continuous interpolation commands which have

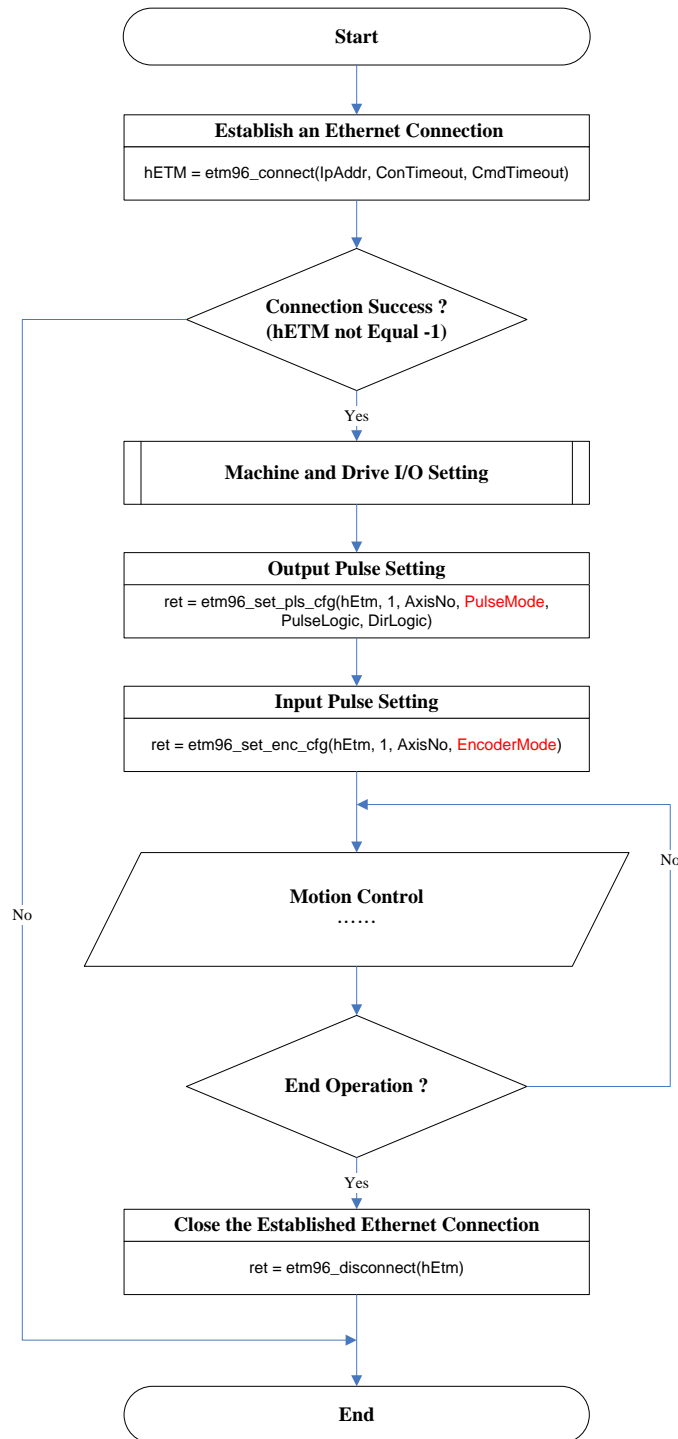
	been written to the TCP/IP packet so far for sending to the ET-M8196F.
Motion Stop Functions	
etm96_stop_move	Stops the current executing motion command for the specified axis. Stops motion before reaching the destination.
etm96_set_softlimit	Sets the software limits for the positive and negative direction. Once a software limit position is specified, the ET-M8196F will not accept position commands beyond the limit and motion will stop once the limit is hit.
etm96_set_softlimit_disable	Disables the axis limits settings.
Multi-Axis Hold/Release Functions	
etm96_drv_hold	This command sets the specified axes in holding mode after the current running command has reached its target position. Therefore this instruction takes effect for the next command. The execution of the next command will be put on hold until the “etm96_drv_start()” releases the hold operation.
etm96_drv_start	Terminates the axes hold operation. Axes which have been put on hold by “etm96_drv_hold()” will continue to execute the next motion command stored in the command FIFO buffer.
Compare Function	
etm96_set_compare_trig_cfg	Configures and enables the compare trigger function. The compare function outputs a signal when the compare condition has been met. Two compare modes are being supported: 1. One time compare mode (Single compare mode) 2. Auto increment compare mode.
Latch Function	
etm96_set_latch_cfg	Configures and enables position Latch. The latch function captures the encoder counter value at an instant when the latch signal activates. The LTC channel is used to receive the latch pulse. The latch function is hardware implemented and executes at very high speed.
etm96_get_latch	Reads the present latched position of the specified axis. Returns the captured position triggered by the latch LTC signal.
General Purpose IO	
etm96_set_general_do	Sets the output state of one general purpose digital output channel.
etm96_get_general_do	Reads the current output state of the specified general purpose DO channel.
etm96_set_all_general_do	Sets all the output signal state of all three general-purpose output channels.
etm96_get_all_general_do	Reads the current output state of all three general purpose DO channels.
etm96_get_general_di	Gets the input state of the specified general purpose DI channel.
etm96_get_all_general_di	Gets the input state of all general purpose DIs.
Remote FRnet IO	
etm96_set_frnet_group_do	Sets the output state of a remote FRnet DO module. Each FRnet DO module has 16 DO channels.
etm96_get_frnet_group_do	Gets the output state of a remote FRnet DO module. Each FRnet DO module has 16 DO channels.
etm96_get_frnet_group_do_all	Gets the output state of a remote FRnet DO module. Each FRnet DO module has 16 DO channels.

etm96_scan_frnet_di	Scans the FRnet network for remote FRnet DI modules and returns which DI modules are connected and active. A maximum number of 8 FRnet DI modules can be used in a FRnet network.
etm96_get_frnet_group_di	Reads 16 channel states of a remote FRnet DI module. Each FRnet DI module has 16 DI channels.
etm96_get_frnet_group_di_all	Reads 16 channel states of a remote FRnet DI module. Each FRnet DI module has 16 DI channels.
ET-M8196F Version	
etm96_get_firmware_version	Gets the firmware version number of the ET-M8196F.
I-8196F Version	
etm96_get_i8196f_version	Read the firmware and hardware version of the I-8196F.
DLL Version	
etm96_get_dll_pathW	Gets the file path location of the loaded "ET-M8196F.dll" in Unicode format.
etm96_get_dll_pathA	Gets the file path location of the loaded "ET-M8196F.dll" in ANSI format.
etm96_get_dll_version	Gets the DLL version.

Table 1: ET-M8196F DLL functions

1.2 Command Flow Chart

This section illustrates the basic function call sequence required for the initialization of the ET-M8196F and motion command execution.



1.2.1 Axis Digital I/O Setting

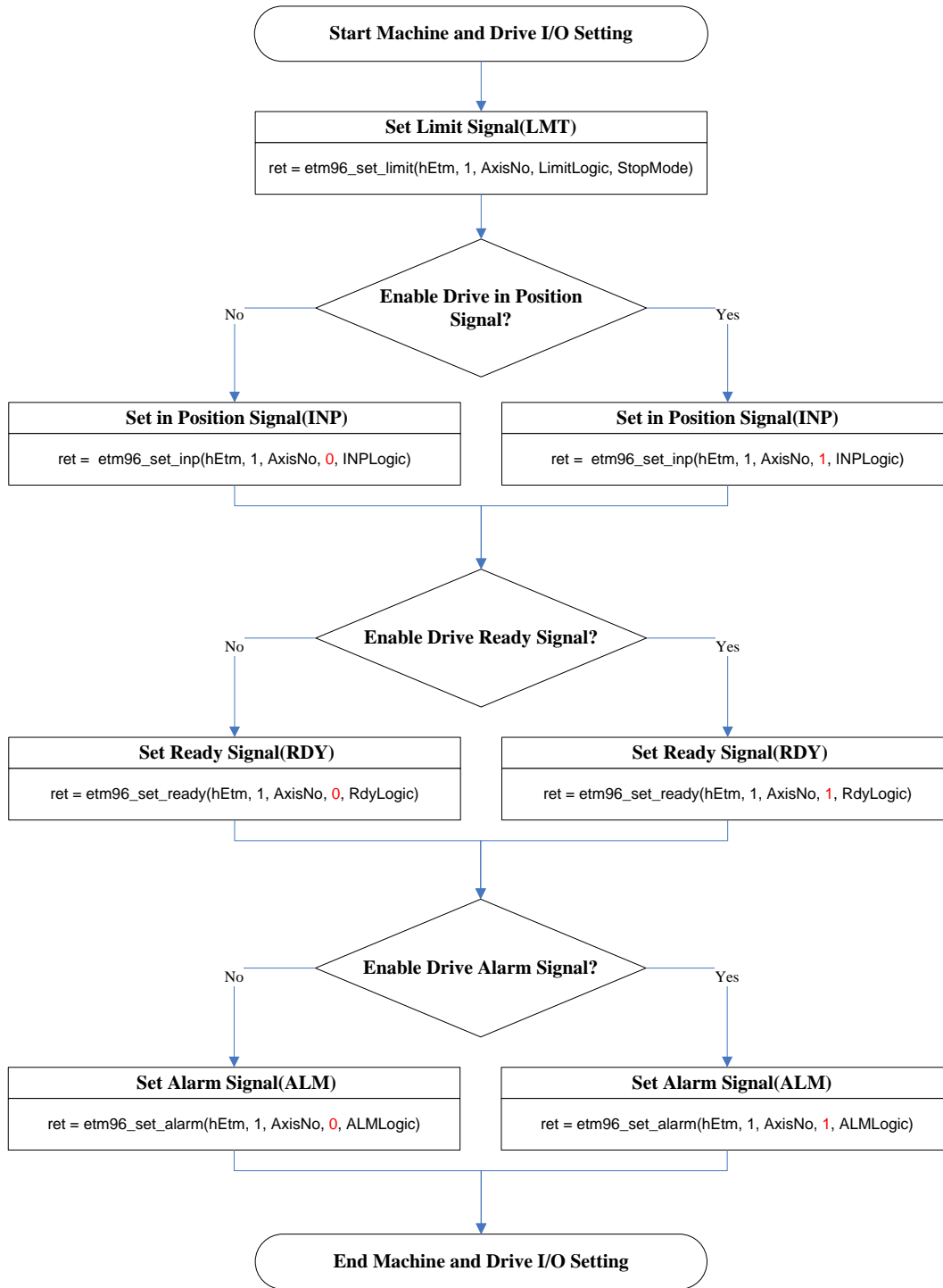


Figure 2: Axis digital I/O setting

1.2.2 Motion Control Initiation

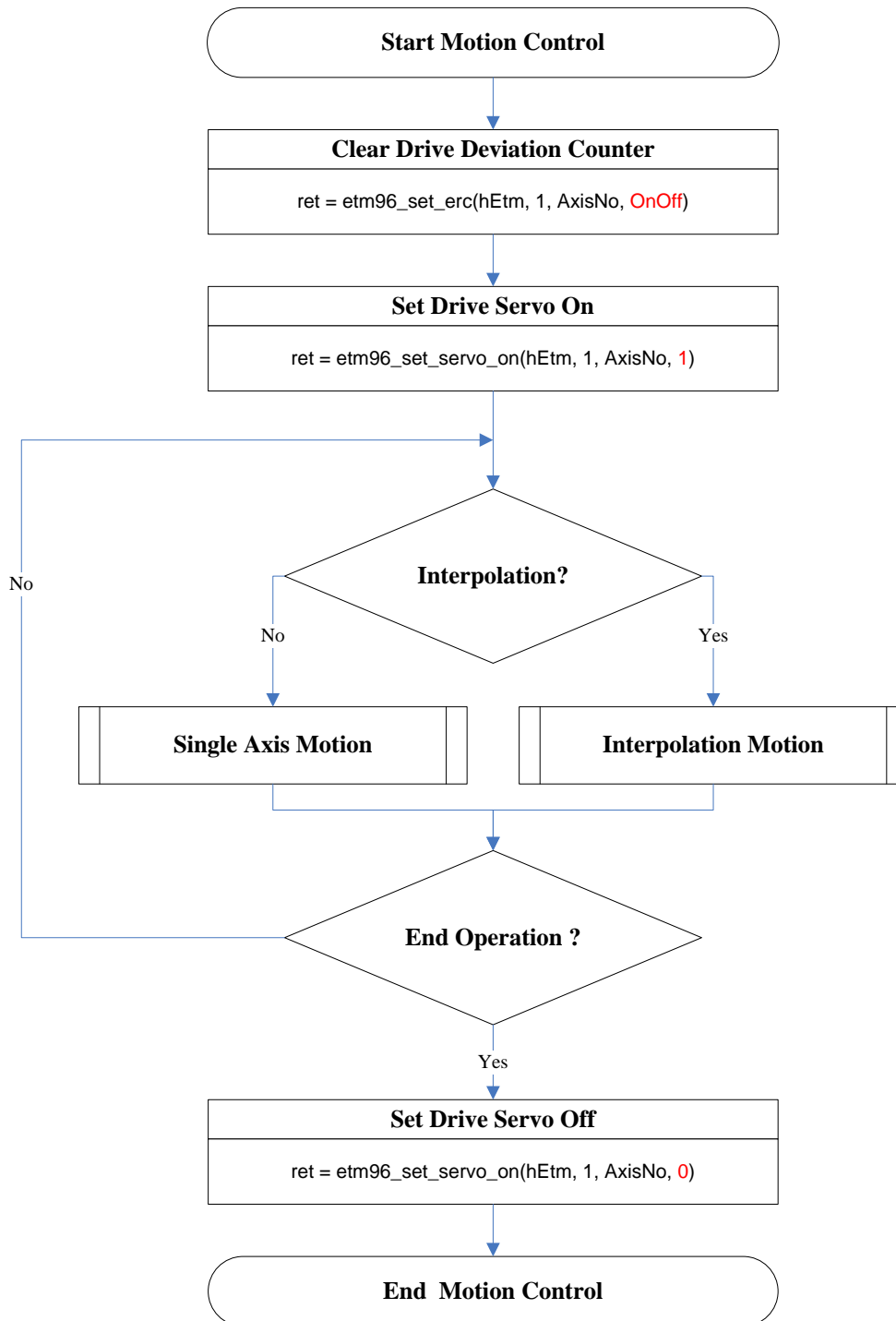


Figure 3: Initiate motion control

1.2.3 Single Axis Motion Control

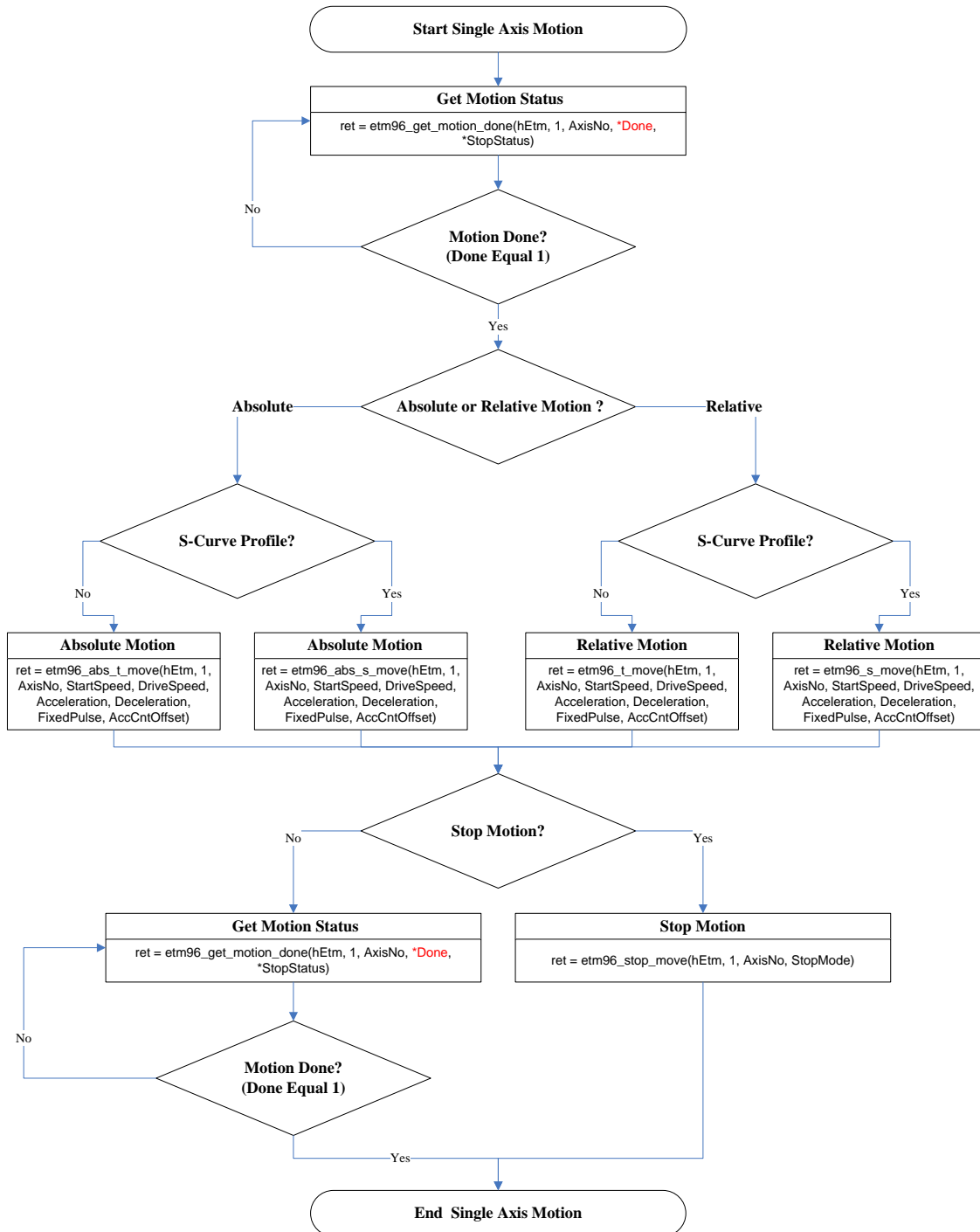


Figure 4: Single axis motion control

1.2.4 Interpolation Motion Control

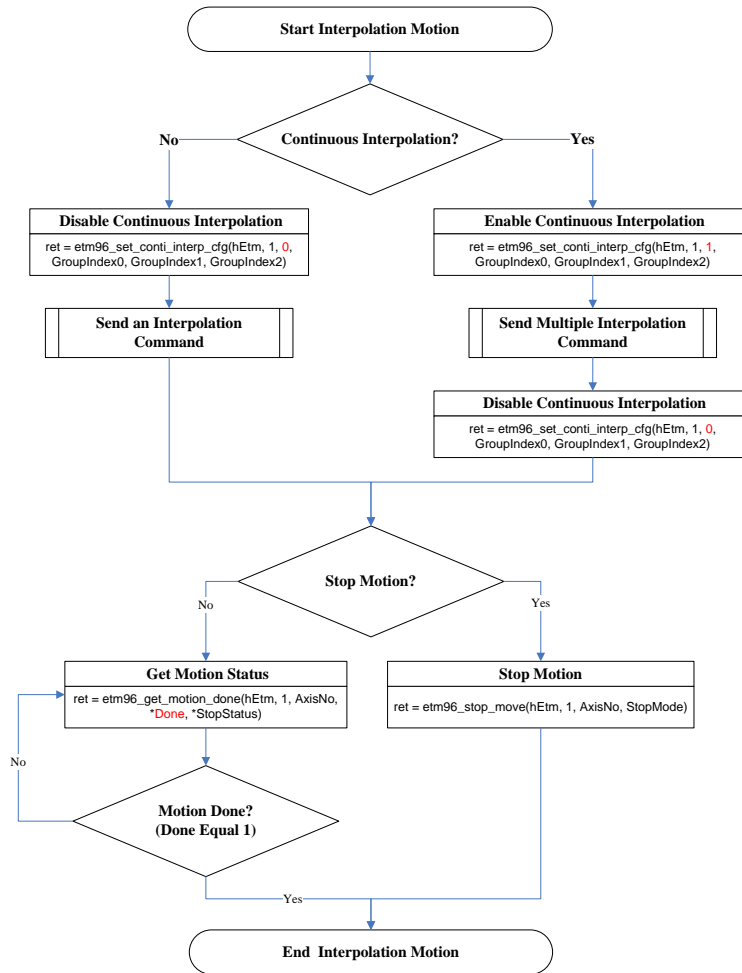


Figure 5: Interpolation motion control settings

1.2.5 Initiate Simple Interpolation Motion

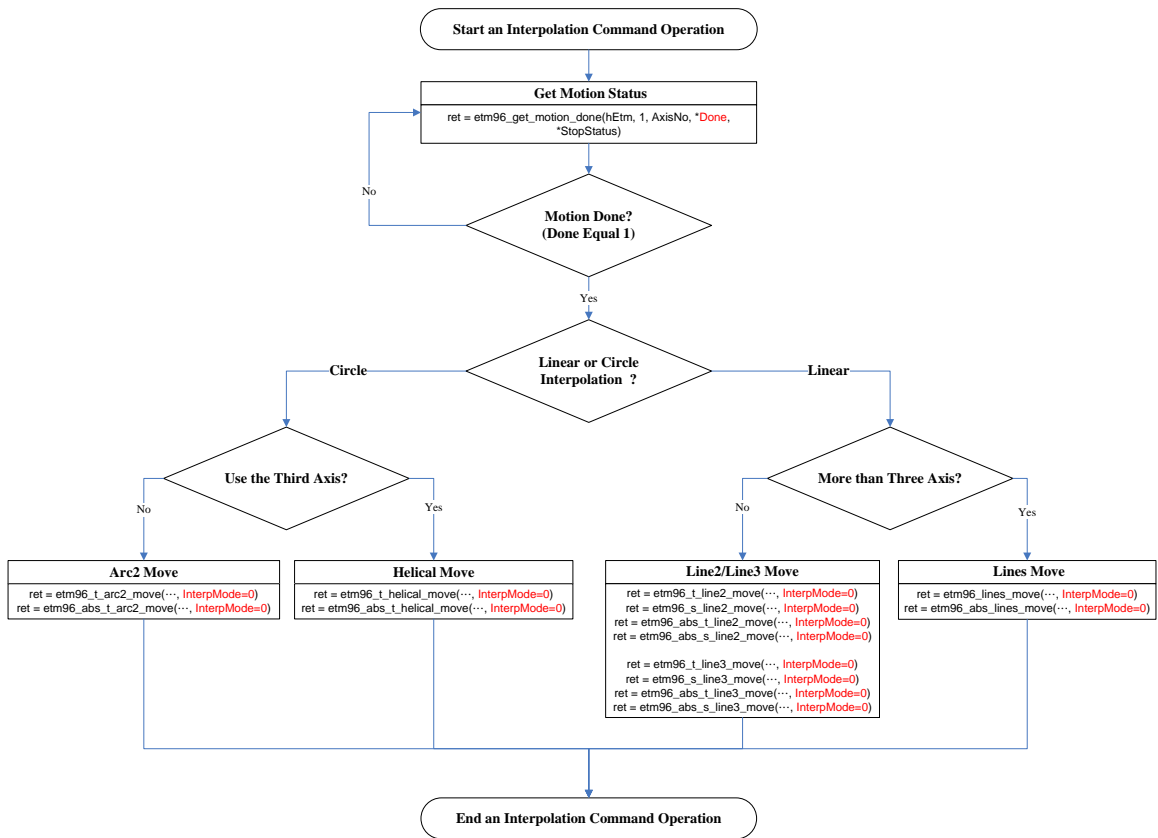


Figure 6: Initiate simple interpolation motion

1.2.6 Initiate Continuous Interpolation Motion

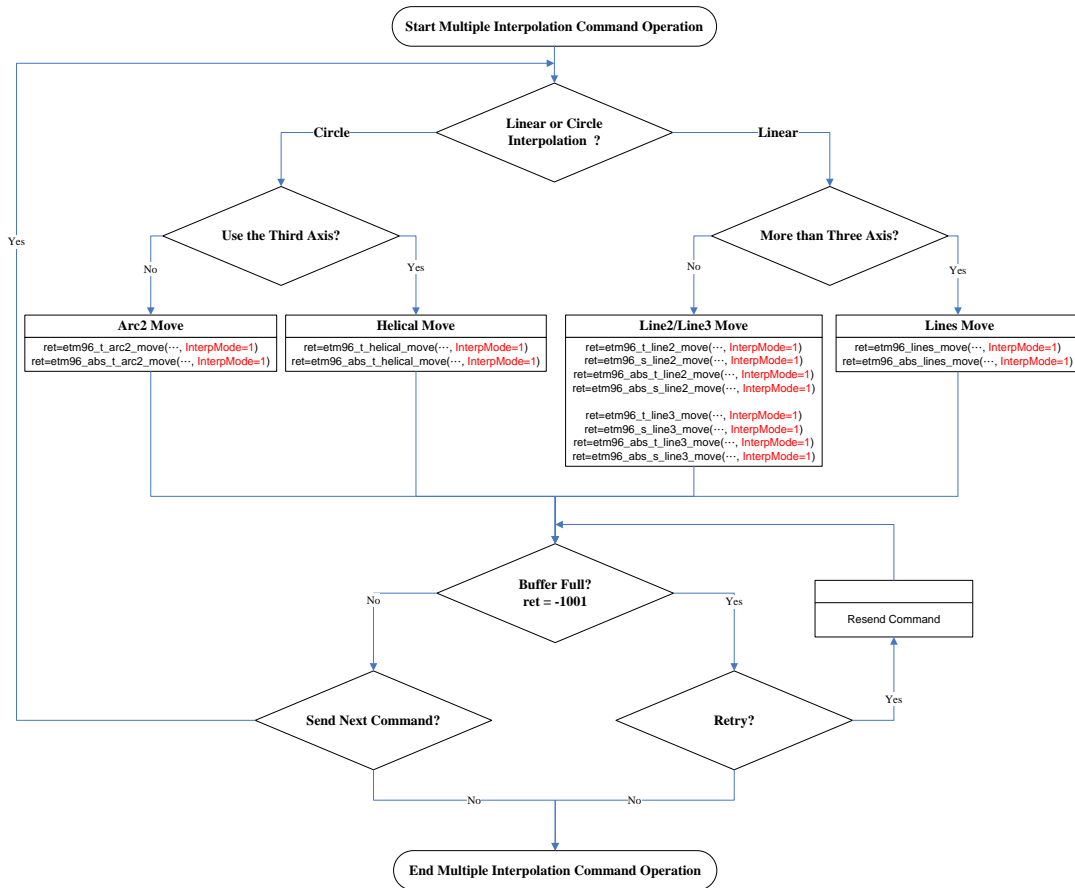


Figure 7: Initiate continuous interpolation motion

2 Ethernet Connection Settings

2.1 TCP/IP Communication

2.1.1 etm96_connectW

Establish an Ethernet connection between PC and ET-M8196F. It is important to first create a connection with ET-M8196F before calling any other command.

Syntax:

```
HANDLE etm96_connectW (    wchar_t* IpAddr,
                          U32 ConTimeout,
                          U32 CmdTimeout);
```

Parameters:

Name	Description
<i>IpAddr:</i>	ET-M8196F IP address
<i>ConTimeout:</i>	Connection timeout (seconds)
<i>CmdTimeout:</i>	Modbus communication timeout (milliseconds)

Return:

- -1 (INVALID_HANDLE_VALUE): Failed to establish connection
- Others: Connection success, return TCP/IP handle

Remarks:

- Although the ET-M8194F allows up to 29 Clients connection, but it is suggested to just use one connection for motion control.

Example:

```
[VC++]
HANDLE hEtm1 = NULL;
wchar_t* pAddr = L"192.168.0.16";

hEtm1 = etm96_connectW(pAddr, 10, 1000); //10 second, 1000
millisecond

if (hEtm1 == INVALID_HANDLE_VALUE)
{MessageBox(L"Establishing a connection to ET-M8196F failed!!");}
```

2.1.2 etm96_connectA

Establish an Ethernet connection between PC and ET-M8196F. It is important to first create a connection with ET-M8196F before calling any other command.

Syntax:

```
HANDLE etm96_connectA (    char* IpAddr,  
                          U32 ConTimeout,  
                          U32 CmdTimeout);
```

Parameters:

Name	Description
<i>IpAddr:</i>	ET-M8196F IP address
<i>ConTimeout:</i>	Connection timeout (seconds)
<i>CmdTimeout:</i>	Modbus communication timeout (milliseconds)

Return:

- -1 (INVALID_HANDLE_VALUE): Failed to establish connection
- Others: Connection success, return TCP/IP handle

Remarks:

Example:

[VC++]

```
HANDLE hEtm1 = NULL;  
char* pAddr = "192.168.0.16";  
  
hEtm1 = etm96_connectA(pAddr, 10, 1000); //10 second, 1000 millisecond  
  
if (hEtm1 == INVALID_HANDLE_VALUE)  
{MessageBox("Establishing a connection to ET-M8196F failed!!");}
```

2.1.3 etm96_disconnect

Close the established Ethernet connection and release internal memory assigned for the connection.

Syntax:

```
HANDLE etm96_disconnect ( HANDLE hEtm);
```

Parameters:

Name	Description
<i>hEtm:</i>	TCP/IP handle

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3 Motion Commands

3.1 Axis Digital I/O Function

The digital IO discussed in this chapter influences the axis operation.

3.1.1 `etm96_set_servo_on`

Set the SRV_ON channel output signal to enable/disable the servo drive for pulse command input.

Syntax:

```
I16 etm96_set_servo_on ( HANDLE hEtm,  
                        U8 CardNo,  
                        U8 AxisNo,  
                        U8 OnOff );
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>OnOff:</i>	<table border="1"><thead><tr><th>Status</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	Status	Value	OFF	OFF (0x00)	ON	ON (0x01)								
Status	Value														
OFF	OFF (0x00)														
ON	ON (0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.1.2 etm96_set_erc

The ERC signal sets the servo drive deflection counter to zero.

Syntax:

```
l16 etm96_set_erc ( HANDLE hEtm,  
                   U8 CardNo,  
                   U8 AxisNo,  
                   U8 OnOff);
```

Parameters:

Name	Description														
<i>hEtm</i> :	TCP/IP handle														
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo</i> :	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>OnOff</i> :	<table border="1"><thead><tr><th>Status</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	Status	Value	OFF	OFF (0x00)	ON	ON (0x01)								
Status	Value														
OFF	OFF (0x00)														
ON	ON (0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- Each servo drive has a deviation counter, which determines the difference between input pulse and feedback pulse. The ERC signal will set the deviation counter of the servo drive to zero and stops the motion if no new pulse command is being issued.

3.1.3 etm96_set_alarm_reset

Reset the servo drive alarm signal (ALM_RST).

Syntax:

```
I16 etm96_set_alarm_reset ( HANDLE hEtm,  
                             U8 CardNo,  
                             U8 AxisNo,  
                             U8 OnOff);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>OnOff:</i>	<table border="1"><thead><tr><th>Status</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	Status	Value	OFF	OFF (0x00)	ON	ON (0x01)								
Status	Value														
OFF	OFF (0x00)														
ON	ON (0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- Set the alarm reset DO signal to on to clear the servo drive alarm fault.

3.1.4 etm96_set_alarm

Enable/Disable the alarm function and set the active level of the servo alarm signal (ALARM). If the servo drive encounters an abnormality while driving, it sends a signal to the ALARM channel of the ET-M8196F. If the servo alarm function of the ET-M8196F has been enabled and the servo alarm signal is active then no pulses will be outputted.

Syntax:

```

116 etm96_set_alarm ( HANDLE hEtm,
                      U8 CardNo,
                      U8 AxisNo,
                      U8 EnableDisable,
                      U8 ALMLogic);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EnableDisable:</i>	Enable the use of the servo alarm signal <table border="1"> <thead> <tr> <th>State</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Disable</td> <td>(0x00)</td> </tr> <tr> <td>Enable</td> <td>(0x01)</td> </tr> </tbody> </table>	State	Value	Disable	(0x00)	Enable	(0x01)								
State	Value														
Disable	(0x00)														
Enable	(0x01)														
<i>ALMLogic:</i>	<table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.1.5 etm96_set_inp

Enable/Disable the in-position function and set the active level of the servo in-position signal (INP). In general, when servo drive is set to position mode (P mode), the servo issues a (INP) pulse signal to controller when movement get into position. If the servo in-position function of the ET-M8196F has been enabled, then the controller waits until the in-position signal of the servo has been triggered before continuing to execute the next motion command.

Syntax:

```

116 etm96_set_inp (      HANDLE hEtm,
                        U8 CardNo,
                        U8 AxisNo,
                        U8 EnableDisable,
                        U8 INPLogic);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1" data-bbox="500 1039 846 1289"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EnableDisable:</i>	Enable the use of the servo in-position signal <table border="1" data-bbox="500 1329 750 1442"> <thead> <tr> <th>State</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Disable</td> <td>(0x00)</td> </tr> <tr> <td>Enable</td> <td>(0x01)</td> </tr> </tbody> </table>	State	Value	Disable	(0x00)	Enable	(0x01)								
State	Value														
Disable	(0x00)														
Enable	(0x01)														
<i>INPLogic:</i>	<table border="1" data-bbox="500 1482 867 1591"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- Call “`etm96_get_io_status()`” to get the INP status and “`etm96_get_motion_done()`” to check whether the motion command has finished executing.

3.1.6 etm96_set_ready

Enables/Disables the ET-M8196F to check for the servo drive “ready” state and sets the active level of the servo RDY signal.

Syntax:

```

I16 etm96_set_ready ( HANDLE hEtm,
                     U8 CardNo,
                     U8 AxisNo,
                     U8 EnableDisable,
                     U8 RdyLogic);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EnableDisable:</i>	Enable the use of the servo ready signal <table border="1"> <thead> <tr> <th>State</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Disable</td> <td>(0x00)</td> </tr> <tr> <td>Enable</td> <td>(0x01)</td> </tr> </tbody> </table>	State	Value	Disable	(0x00)	Enable	(0x01)								
State	Value														
Disable	(0x00)														
Enable	(0x01)														
<i>RdyLogic:</i>	<table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- If the servo ready function has been enabled and the RDY signal is OFF then the ET-M8196F will not execute a motion command until the RDY signal turns ON.
- Call “etm96_get_io_status()” to get the RDY status.

3.1.7 etm96_set_limit

Sets the logic levels of the LMT+ and LMT- channels and the stop mode. The hardware limit signals (LMT+, LMT-) are used for stopping the pulse output when the limit switches are triggered. The hardware limit switches are being used for mechanical protection of the system. If the positive switch (LMT+) is being triggered while the movement is in positive direction the motion stops according to the set stop mode. On the other hand the motion will stop when moving in negative direction and the negative limit switch (LMT-) is active.

Syntax:

```
I16 etm96_set_limit ( HANDLE hEtm,
                     U8 CardNo,
                     U8 AxisNo,
                     U8 LimitLogic,
                     U16 StopMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>LimitLogic:</i>	<table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														
<i>StopMode:</i>	<table border="1"> <thead> <tr> <th>Stop Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Disable</td> <td>(0)</td> </tr> <tr> <td>Deceleration stop</td> <td>(1)</td> </tr> <tr> <td>Immediate stop</td> <td>(2)</td> </tr> </tbody> </table>	Stop Mode	Value	Disable	(0)	Deceleration stop	(1)	Immediate stop	(2)						
Stop Mode	Value														
Disable	(0)														
Deceleration stop	(1)														
Immediate stop	(2)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- If the axis moves in positive direction and triggers the positive limit switch (LMT+) then the motion will stop but it will not stop when activating the negative limit switch (LMT-) and moving in positive direction. The axis will stop when moving in negative direction and the negative limit switch (LMT-) is active.
- Call “`etm96_get_io_status()`” to get the “LMT+” and “LMT-“ status (LMTP, LMTM).

3.1.8 etm96_get_io_status

Reads the current axis I/O channels state.

Syntax:

```

116 etm96_get_io_status ( HANDLE hEtm,
                          U8 CardNo,
                          U8 AxisNo,
                          U16* IOStatus);
    
```

Parameters:

Name	Description																																																			
<i>hEtm:</i>	TCP/IP handle																																																			
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																																																			
<i>AxisNo:</i>	Axis definition: <table border="1" data-bbox="501 808 846 1058"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)																																					
Axis	Value																																																			
Axis0	AXIS_0 (0x01)																																																			
Axis1	AXIS_1 (0x02)																																																			
Axis2	AXIS_2 (0x04)																																																			
Axis3	AXIS_3 (0x08)																																																			
Axis4	AXIS_4 (0x10)																																																			
Axis5	AXIS_5 (0x20)																																																			
<i>IOStatus:</i>	<table border="1" data-bbox="501 1098 1360 1801"> <thead> <tr> <th>Bit Position</th> <th>Corresponding Signal</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Bit 0</td> <td>reserved</td> <td></td> </tr> <tr> <td>Bit 1</td> <td>LMT+</td> <td>Positive limit switch</td> </tr> <tr> <td>Bit 2</td> <td>LMT-</td> <td>Negative limit switch</td> </tr> <tr> <td>Bit 3</td> <td>EMG</td> <td>Emergency stop switch</td> </tr> <tr> <td>Bit 4</td> <td>ALARM</td> <td>Servo drive alarm signal</td> </tr> <tr> <td>Bit 5</td> <td>HOME (ORG)</td> <td>Home switch</td> </tr> <tr> <td>Bit 6</td> <td>SLD (NHOME)</td> <td>Slow down switch</td> </tr> <tr> <td>Bit 7</td> <td>INP</td> <td>Servo drive in-position signal</td> </tr> <tr> <td>Bit 8</td> <td>EZ</td> <td>Servo drive Z phase (Index signal)</td> </tr> <tr> <td>Bit 9</td> <td>RDY</td> <td>Servo drive ready signal</td> </tr> <tr> <td>Bit 10</td> <td>LTC</td> <td>Latch input</td> </tr> <tr> <td>Bit 11</td> <td>reserved</td> <td></td> </tr> <tr> <td>Bit 12</td> <td>reserved</td> <td></td> </tr> <tr> <td>Bit 13</td> <td>SRV_ON</td> <td>Signal for activating servo drive</td> </tr> <tr> <td>Bit 14</td> <td>ERC</td> <td>Output signal for clearing position deflection of servo drive</td> </tr> <tr> <td>Bit 15</td> <td>ALM_RST</td> <td>Signal for resetting servo drive alarm</td> </tr> </tbody> </table> <p>If bit is zero: the corresponding signal is OFF If bit is one: the corresponding signal is ON</p>	Bit Position	Corresponding Signal	Description	Bit 0	reserved		Bit 1	LMT+	Positive limit switch	Bit 2	LMT-	Negative limit switch	Bit 3	EMG	Emergency stop switch	Bit 4	ALARM	Servo drive alarm signal	Bit 5	HOME (ORG)	Home switch	Bit 6	SLD (NHOME)	Slow down switch	Bit 7	INP	Servo drive in-position signal	Bit 8	EZ	Servo drive Z phase (Index signal)	Bit 9	RDY	Servo drive ready signal	Bit 10	LTC	Latch input	Bit 11	reserved		Bit 12	reserved		Bit 13	SRV_ON	Signal for activating servo drive	Bit 14	ERC	Output signal for clearing position deflection of servo drive	Bit 15	ALM_RST	Signal for resetting servo drive alarm
Bit Position	Corresponding Signal	Description																																																		
Bit 0	reserved																																																			
Bit 1	LMT+	Positive limit switch																																																		
Bit 2	LMT-	Negative limit switch																																																		
Bit 3	EMG	Emergency stop switch																																																		
Bit 4	ALARM	Servo drive alarm signal																																																		
Bit 5	HOME (ORG)	Home switch																																																		
Bit 6	SLD (NHOME)	Slow down switch																																																		
Bit 7	INP	Servo drive in-position signal																																																		
Bit 8	EZ	Servo drive Z phase (Index signal)																																																		
Bit 9	RDY	Servo drive ready signal																																																		
Bit 10	LTC	Latch input																																																		
Bit 11	reserved																																																			
Bit 12	reserved																																																			
Bit 13	SRV_ON	Signal for activating servo drive																																																		
Bit 14	ERC	Output signal for clearing position deflection of servo drive																																																		
Bit 15	ALM_RST	Signal for resetting servo drive alarm																																																		

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

Example:

[VC++]

```
I16 ret = 0;
U16 IOStatus = 0;

ret = etm96_get_io_status(hEtm1, 1, 0x01, &IOStatus);
if (ret == 0) // ETM96_ERR_NO_ERROR
{
    if (IOStatus & 0x0008)
    {MessageBox("Emergency Stop!!");}
    else if (IOStatus & 0x0002)
    {MessageBox("Reach Positive Limit!!");}
    else if (IOStatus & 0x0004)
    {MessageBox("Reach Negative Limit!!");}
}
else
{MessageBox("Get IO Status Error !!!");}
```

3.1.9 etm96_get_io_status_all

Reads the current I/O channels state of all six axes.

Syntax:

```
U16 etm96_get_io_status_all ( HANDLE hEtm,
                             U8 CardNo,
                             U16* ArrIOStatus);
```

Parameters:

Name	Description																																																																	
<i>hEtm:</i>	TCP/IP handle																																																																	
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																																																																	
<i>ArrIOStatus:</i>	<p>Pointer to an array with 6 elements: U16 ArrIOStatus[6] ; Each array element contains the status of all I/O channels of one axis.</p> <table border="1"> <thead> <tr> <th>Array Index</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>AXIS_0 I/O status</td> </tr> <tr> <td>[1]</td> <td>AXIS_1 I/O status</td> </tr> <tr> <td>[2]</td> <td>AXIS_2 I/O status</td> </tr> <tr> <td>[3]</td> <td>AXIS_3 I/O status</td> </tr> <tr> <td>[4]</td> <td>AXIS_4 I/O status</td> </tr> <tr> <td>[5]</td> <td>AXIS_5 I/O status</td> </tr> </tbody> </table> <p>The status definition of one axis:</p> <table border="1"> <thead> <tr> <th>Bit Position</th> <th>Corresponding Signal</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Bit 0</td> <td>reserved</td> <td></td> </tr> <tr> <td>Bit 1</td> <td>LMT+</td> <td>Positive limit switch</td> </tr> <tr> <td>Bit 2</td> <td>LMT-</td> <td>Negative limit switch</td> </tr> <tr> <td>Bit 3</td> <td>EMG</td> <td>Emergency stop switch</td> </tr> <tr> <td>Bit 4</td> <td>ALARM</td> <td>Servo drive alarm signal</td> </tr> <tr> <td>Bit 5</td> <td>HOME (ORG)</td> <td>Home switch</td> </tr> <tr> <td>Bit 6</td> <td>SLD (NHOME)</td> <td>Slow down switch</td> </tr> <tr> <td>Bit 7</td> <td>INP</td> <td>Servo drive in-position signal</td> </tr> <tr> <td>Bit 8</td> <td>EZ</td> <td>Servo drive Z phase (Index signal)</td> </tr> <tr> <td>Bit 9</td> <td>RDY</td> <td>Servo drive ready signal</td> </tr> <tr> <td>Bit 10</td> <td>LTC</td> <td>Latch input</td> </tr> <tr> <td>Bit 11</td> <td>reserved</td> <td></td> </tr> <tr> <td>Bit 12</td> <td>reserved</td> <td></td> </tr> <tr> <td>Bit 13</td> <td>SRV_ON</td> <td>Signal for activating servo drive</td> </tr> <tr> <td>Bit 14</td> <td>ERC</td> <td>Output signal for clearing position deflection of servo drive</td> </tr> <tr> <td>Bit 15</td> <td>ALM_RST</td> <td>Signal for resetting servo drive</td> </tr> </tbody> </table>	Array Index	Status	[0]	AXIS_0 I/O status	[1]	AXIS_1 I/O status	[2]	AXIS_2 I/O status	[3]	AXIS_3 I/O status	[4]	AXIS_4 I/O status	[5]	AXIS_5 I/O status	Bit Position	Corresponding Signal	Description	Bit 0	reserved		Bit 1	LMT+	Positive limit switch	Bit 2	LMT-	Negative limit switch	Bit 3	EMG	Emergency stop switch	Bit 4	ALARM	Servo drive alarm signal	Bit 5	HOME (ORG)	Home switch	Bit 6	SLD (NHOME)	Slow down switch	Bit 7	INP	Servo drive in-position signal	Bit 8	EZ	Servo drive Z phase (Index signal)	Bit 9	RDY	Servo drive ready signal	Bit 10	LTC	Latch input	Bit 11	reserved		Bit 12	reserved		Bit 13	SRV_ON	Signal for activating servo drive	Bit 14	ERC	Output signal for clearing position deflection of servo drive	Bit 15	ALM_RST	Signal for resetting servo drive
Array Index	Status																																																																	
[0]	AXIS_0 I/O status																																																																	
[1]	AXIS_1 I/O status																																																																	
[2]	AXIS_2 I/O status																																																																	
[3]	AXIS_3 I/O status																																																																	
[4]	AXIS_4 I/O status																																																																	
[5]	AXIS_5 I/O status																																																																	
Bit Position	Corresponding Signal	Description																																																																
Bit 0	reserved																																																																	
Bit 1	LMT+	Positive limit switch																																																																
Bit 2	LMT-	Negative limit switch																																																																
Bit 3	EMG	Emergency stop switch																																																																
Bit 4	ALARM	Servo drive alarm signal																																																																
Bit 5	HOME (ORG)	Home switch																																																																
Bit 6	SLD (NHOME)	Slow down switch																																																																
Bit 7	INP	Servo drive in-position signal																																																																
Bit 8	EZ	Servo drive Z phase (Index signal)																																																																
Bit 9	RDY	Servo drive ready signal																																																																
Bit 10	LTC	Latch input																																																																
Bit 11	reserved																																																																	
Bit 12	reserved																																																																	
Bit 13	SRV_ON	Signal for activating servo drive																																																																
Bit 14	ERC	Output signal for clearing position deflection of servo drive																																																																
Bit 15	ALM_RST	Signal for resetting servo drive																																																																

			alarm
If bit is zero: the corresponding signal is OFF			
If bit is one: the corresponding signal is ON			

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.1.10 etm96_get_servo_on_status

Reads the output signal of the SRV_ON channel. The SRV_ON channel determines whether the servo drive has been enabled to control the motor. The SRV_ON signal is set by “etm96_set_servo_on()”.

Syntax:

```
I16 etm96_get_servo_on_status ( HANDLE hEtm,  
                                U8 CardNo,  
                                U8 AxisNo,  
                                U8* OnOff);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>OnOff:</i>	<table border="1"><thead><tr><th>Status</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	Status	Value	OFF	OFF (0x00)	ON	ON (0x01)								
Status	Value														
OFF	OFF (0x00)														
ON	ON (0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.1.11 etm96_get_erc_status

Reads the ERC output signal. The ERC signal clears the deviation counter of the servo drive and is set by calling “etm96_set_erc()”.

Syntax:

```
l16 etm96_get_erc_status ( HANDLE hEtm,  
                          U8 CardNo,  
                          U8 AxisNo,  
                          U8* OnOff);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>OnOff:</i>	<table border="1"><thead><tr><th>Status</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	Status	Value	OFF	OFF (0x00)	ON	ON (0x01)								
Status	Value														
OFF	OFF (0x00)														
ON	ON (0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.1.12 etm96_get_alarm_reset_status

Reads the status of the ALM_RST output channel. The ALM_RST signal resets the alarm state of the servo drive. "etm96_set_alarm_reset()" sets the output signal.

Syntax:

```
I16 etm96_get_alarm_reset_status ( HANDLE hEtm,  
                                   U8 CardNo,  
                                   U8 AxisNo,  
                                   U8* OnOff);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>OnOff:</i>	<table border="1"><thead><tr><th>Status</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	Status	Value	OFF	OFF (0x00)	ON	ON (0x01)								
Status	Value														
OFF	OFF (0x00)														
ON	ON (0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2 Motion Control Pulse Setting Function

3.2.1 etm96_set_pls_cfg

Set the pulse output mode for each axis.

Syntax:

```

I16 etm96_set_pls_cfg ( HANDLE hEtm,
                        U8 CardNo,
                        U8 AxisNo,
                        U16 PulseMode,
                        U8 PulseLogic,
                        U8 DirectionLogic);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>PulseMode:</i>	<table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Pulse/Direction</td> <td>(0)</td> </tr> <tr> <td>CW/CCW</td> <td>(1)</td> </tr> <tr> <td>A/B Phase</td> <td>(2)</td> </tr> </tbody> </table>	Mode	Value	Pulse/Direction	(0)	CW/CCW	(1)	A/B Phase	(2)						
Mode	Value														
Pulse/Direction	(0)														
CW/CCW	(1)														
A/B Phase	(2)														
<i>PulseLogic:</i>	<table border="1"> <thead> <tr> <th>Logic Level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>(0x1)</td> </tr> <tr> <td>High</td> <td>(0x0)</td> </tr> </tbody> </table>	Logic Level	Value	Low	(0x1)	High	(0x0)								
Logic Level	Value														
Low	(0x1)														
High	(0x0)														
<i>DirectionLogic:</i>	Direction logic-enable signal level: <table border="1"> <thead> <tr> <th>Logic Level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>(0x1)</td> </tr> <tr> <td>High</td> <td>(0x0)</td> </tr> </tbody> </table> Note that in "CW / CCW" and "A / B Phase" mode this parameter is invalid	Logic Level	Value	Low	(0x1)	High	(0x0)								
Logic Level	Value														
Low	(0x1)														
High	(0x0)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

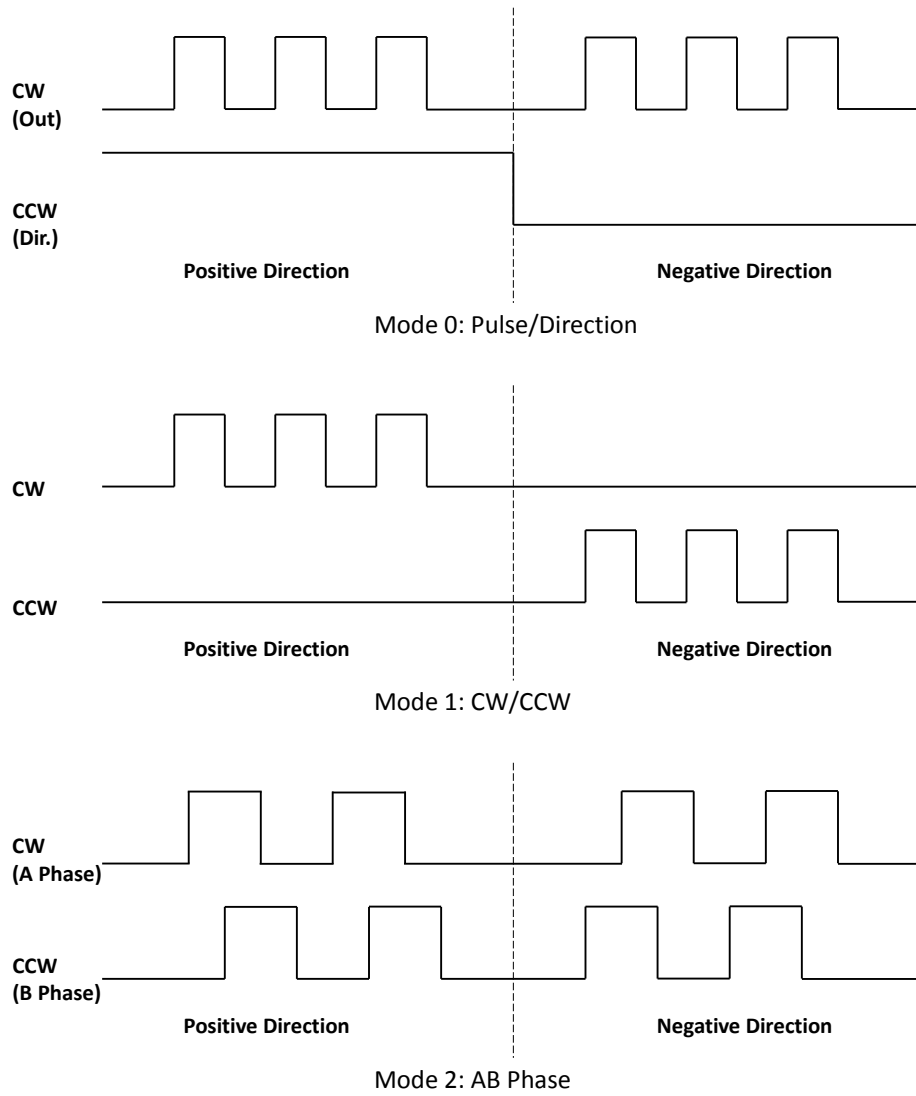


Figure 8: Pulse output modes

3.2.2 etm96_set_enc_cfg

Set the parameters of the encoder pulse input.

Syntax:

```
I16 etm96_set_enc_cfg ( HANDLE hEtm,  
                       U8 CardNo,  
                       U8 AxisNo,  
                       U16 EncoderMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EncoderMode:</i>	<table border="1"><thead><tr><th>Mode</th><th>Value</th></tr></thead><tbody><tr><td>CW/CCW</td><td>(1)</td></tr><tr><td>A/B Phase</td><td>(2)</td></tr><tr><td>A/B Phase divide 2</td><td>(3)</td></tr><tr><td>A/B Phase divide 4</td><td>(4)</td></tr></tbody></table>	Mode	Value	CW/CCW	(1)	A/B Phase	(2)	A/B Phase divide 2	(3)	A/B Phase divide 4	(4)				
Mode	Value														
CW/CCW	(1)														
A/B Phase	(2)														
A/B Phase divide 2	(3)														
A/B Phase divide 4	(4)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.3 etm96_set_cmdcounter

Set the command counter (position command) value.

Syntax:

```
I16 etm96_set_cmdcounter ( HANDLE hEtm,  
                           U8 CardNo,  
                           U8 AxisNo,  
                           I32 LogicPos );
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>LogicPos:</i>	Command counter value														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- This function can only be called if no pulses are being outputted.

3.2.4 etm96_get_cmdcounter

Get the current command counter (position command) value.

Syntax:

```
I16 etm96_get_cmdcounter ( HANDLE hEtm,  
                          U8 CardNo,  
                          U8 AxisNo,  
                          I32* LogicPosCount );
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>LogicPosCount:</i>	Pointer to current command counter (commanded position) value														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.5 etm96_get_cmdcounter_all

Get the current command counter (position command) for all six axes.

Syntax:

```
I16 etm96_get_cmdcounter_all ( HANDLE hEtm,  
                               U8 CardNo,  
                               I32* ArrLogicPos);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>ArrLogicPos:</i>	Pointer to an array with 6 elements: I32 ArrLogicPos[6]; Each array element contains the current command counter (commanded position) of the corresponding axis. <table border="1" data-bbox="505 871 1130 1178"><thead><tr><th>Array Index</th><th>Status</th></tr></thead><tbody><tr><td>[0]</td><td>AXIS_0 commanded position</td></tr><tr><td>[1]</td><td>AXIS_1 commanded position</td></tr><tr><td>[2]</td><td>AXIS_2 commanded position</td></tr><tr><td>[3]</td><td>AXIS_3 commanded position</td></tr><tr><td>[4]</td><td>AXIS_4 commanded position</td></tr><tr><td>[5]</td><td>AXIS_5 commanded position</td></tr></tbody></table>	Array Index	Status	[0]	AXIS_0 commanded position	[1]	AXIS_1 commanded position	[2]	AXIS_2 commanded position	[3]	AXIS_3 commanded position	[4]	AXIS_4 commanded position	[5]	AXIS_5 commanded position
Array Index	Status														
[0]	AXIS_0 commanded position														
[1]	AXIS_1 commanded position														
[2]	AXIS_2 commanded position														
[3]	AXIS_3 commanded position														
[4]	AXIS_4 commanded position														
[5]	AXIS_5 commanded position														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.6 etm96_set_encounter

Set the encoder counter value.

Syntax:

```
I16 etm96_set_encounter ( HANDLE hEtm,  
                          U8 CardNo,  
                          U8 AxisNo,  
                          I32 EncPos);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EncPos:</i>	Encoder counter value														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.7 etm96_get_encounter

Get the current encoder counter value.

Syntax:

```
I16 etm96_get_encounter ( HANDLE hEtm,  
                          U8 CardNo,  
                          U8 AxisNo,  
                          I32* EncPos);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EncPos:</i>	Pointer to current encoder counter														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.8 etm96_get_enccounter_all

Get the current encoder counter value for all six axes.

Syntax:

```
I16 etm96_get_enccounter_all ( HANDLE hEtm,  
                             U8 CardNo,  
                             I32* ArrEncPos);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>ArrEncPos:</i>	Pointer to an array with 6 elements: I32 ArrEncPos[6]; Each array element contains the current encoder value of the corresponding axis. <table border="1" data-bbox="537 871 1166 1178"><thead><tr><th>Array Index</th><th>Status</th></tr></thead><tbody><tr><td>[0]</td><td>AXIS_0 encoder value</td></tr><tr><td>[1]</td><td>AXIS_1 encoder value</td></tr><tr><td>[2]</td><td>AXIS_2 encoder value</td></tr><tr><td>[3]</td><td>AXIS_3 encoder value</td></tr><tr><td>[4]</td><td>AXIS_4 encoder value</td></tr><tr><td>[5]</td><td>AXIS_5 encoder value</td></tr></tbody></table>	Array Index	Status	[0]	AXIS_0 encoder value	[1]	AXIS_1 encoder value	[2]	AXIS_2 encoder value	[3]	AXIS_3 encoder value	[4]	AXIS_4 encoder value	[5]	AXIS_5 encoder value
Array Index	Status														
[0]	AXIS_0 encoder value														
[1]	AXIS_1 encoder value														
[2]	AXIS_2 encoder value														
[3]	AXIS_3 encoder value														
[4]	AXIS_4 encoder value														
[5]	AXIS_5 encoder value														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.9 etm96_set_vring_counter

Set the maximum ring counter position for both the encoder and commanded position counter.

Syntax:

```

I16 etm96_set_vring_counter ( HANDLE hEtm,
                               U8 CardNo,
                               U8 AxisNo,
                               U32 RingValue);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>RingValue:</i>	The upper limit of the encoder counter value (range: 2~2147483647)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- The counter setting will disable the software limit setting
- Ring counter function is not support when axis is in compare trigger mode
- Use the function "etm96_disable_vring_counter()" to turn off the ring counter setting
- The ring position counter operation:

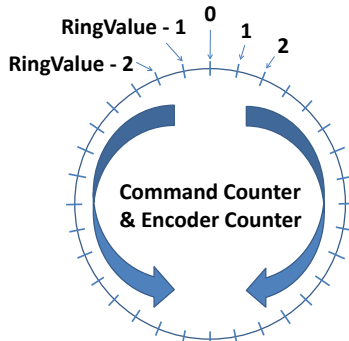


Figure 9: Ring position counter

3.2.10 etm96_get_vring_counter

Read the maximum ring counter setting.

Syntax:

```
I16 etm96_get_vring_counter ( HANDLE hEtm,  
                             U8 CardNo,  
                             U8 AxisNo,  
                             U32* RingValue);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>RingValue:</i>	<ul style="list-style-type: none">• Pointer to current ring counter value setting (range: 2~2147483647)• Returns the value set by “etm96_set_vring_counter()”														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

3.2.11 etm96_disable_vring_counter

Disable the ring counter setting.

Syntax:

```
I16 etm96_disable_vring_counter ( HANDLE hEtm,  
                                U8 CardNo,  
                                U8 AxisNo);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- The counter value range will be set back to -2,147,483,648 to 2,147,483,647.

4 Automatic Home Search

4.1 Automatic Home Search Configuration

4.1.1 `etm96_set_home_cfg`

Set automatic home search parameters.

Syntax:

```

I16 etm96_set_home_cfg ( HANDLE hEtm,
                        U8 CardNo,
                        U8 AxisNo,
                        U8 HomeLogic,
                        U8 SLDLogic,
                        U8 EZLogic,
                        U8 HomeSteps,
                        I32 Step4Offset);

```

Parameters:

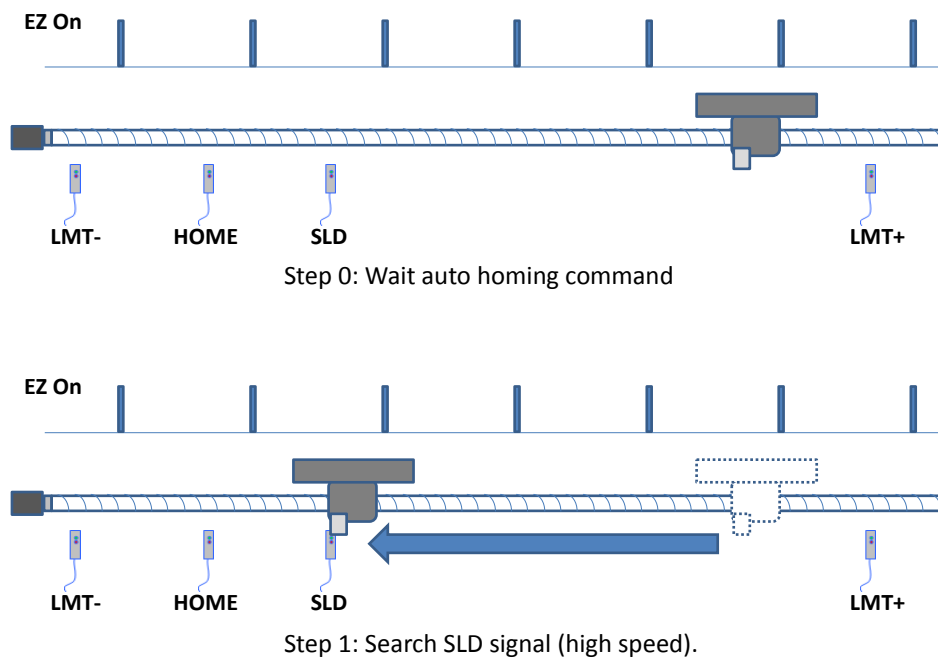
Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>HomeLogic:</i>	Logic level of origin (HOME) input signal: <table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														
<i>SLDLogic:</i>	Logic level of the slow down (SLD) input signal: <table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														
<i>EZLogic:</i>	Logic level of servo drive Z phase (Index signal) <table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)										
Trigger level	Value														
Active low	(0x00)														

	Active high	(0x01)
<i>HomeSteps:</i>	Bit Position	Corresponding Home Step
	Bit 1	Step1: High speed near home search(Search SLD)
	Bit 2	Reserved
	Bit 3	Step2: Low speed home search (Search HOME)
	Bit 4	Reserved
	Bit 5	Step3: Low speed servo drive Z phase (Index signal) search (Search EZ)
	Bit 6	Reserved
	Bit 7	Step4: High speed offset drive (Offset)
	Bit 8	Reserved
<i>Step4Offset:</i>	Offset position	

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:



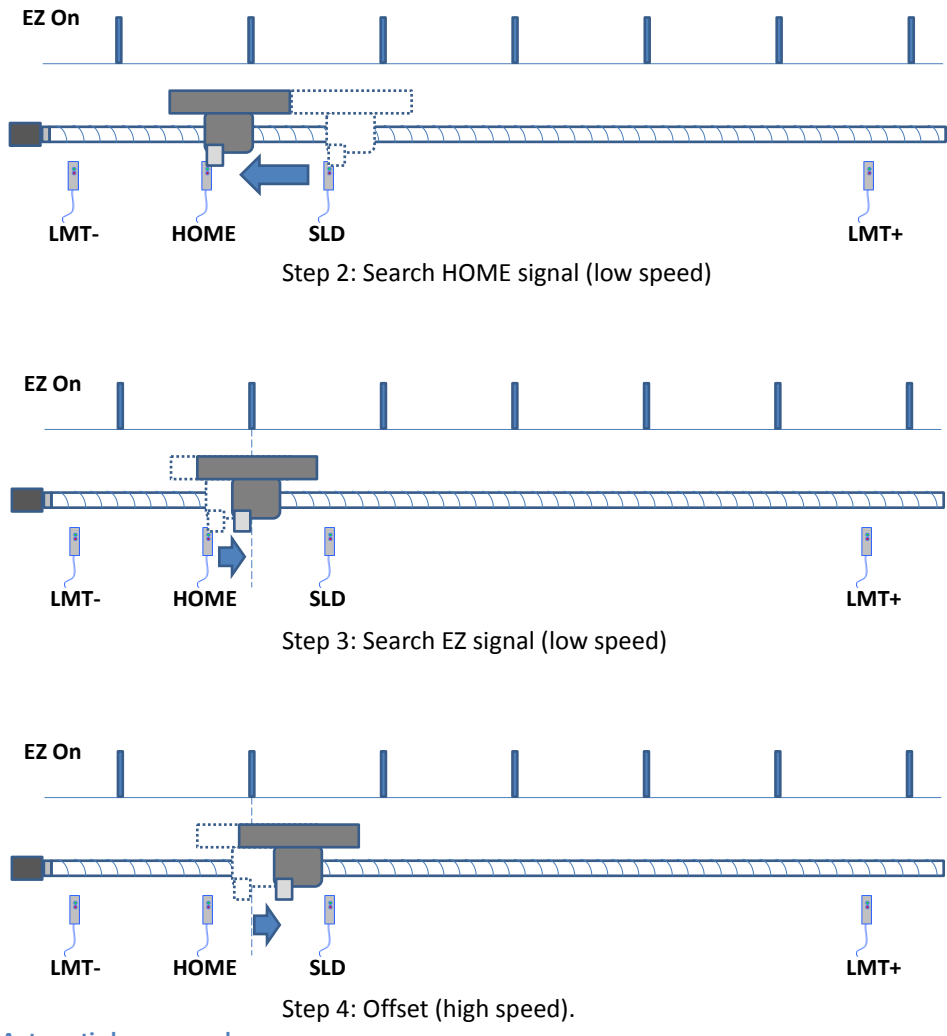


Figure 10: Automatic home search

4.2 Automatic Home Execution

4.2.1 etm96_home_start

Start searching for the home position.

Syntax:

```

I16 etm96_home_start ( HANDLE hEtm,
                       U8 CardNo,
                       U8 AxisNo,
                       U32 StartSpeed,
                       U32 Acceleration,
                       U32 Deceleration,
                       U32 HighSearchSpeed,
                       U32 LowSearchSpeed,
                       U8 HomingDirection);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1" data-bbox="571 1079 915 1329"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Start speed (PPS)														
<i>Acceleration:</i>	Acceleration (PPS/Sec)														
<i>Deceleration:</i>	Deceleration (PPS/Sec)														
<i>HighSearchSpeed:</i>	Near home search (step 1) and offset drive (step 4) speed (PPS)														
<i>LowSearchSpeed:</i>	Home search (step 2) and servo drive servo drive Z phase (Index) search (step 3) speed														
<i>HomingDirection:</i>	Home search direction <table border="1" data-bbox="571 1593 850 1703"> <thead> <tr> <th>Direction</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Negative</td> <td>(0)</td> </tr> <tr> <td>Positive</td> <td>(1)</td> </tr> </tbody> </table>	Direction	Value	Negative	(0)	Positive	(1)								
Direction	Value														
Negative	(0)														
Positive	(1)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

5 Motion Control Instructions

5.1 Read Motion Status

5.1.1 `etm96_get_motion_done`

Read the current motion status of the axis.

Syntax:

```
l16 etm96_get_motion_done ( HANDLE hEtm,  
                             U8 CardNo,  
                             U8 AxisNo,  
                             U8* Done,  
                             U16* StopStatus);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>Done:</i>	0: a motion command is being executed (axis is outputting pulse) 1: motion has finished														
<i>StopStatus:</i>	Indicates the cause of a motion stop: If StopStatus = 0 then motion command is still running <table border="1"><thead><tr><th>Bit</th><th>Corresponding stop cause</th></tr></thead><tbody><tr><td>Bit 0</td><td>Command has reached the target position. Motion has finished without any error</td></tr><tr><td>Bit 1</td><td>Automatic home search has finished (see “<code>etm96_home_start()</code>”)</td></tr><tr><td>Bit 2</td><td>Motion command has been interrupted by a stop command (“<code>etm96_stop_move()</code>”)</td></tr><tr><td>Bit 3</td><td>The axis finished outputting pulse commands and waits for the in-position signal of the servo drive (see “<code>etm96_set_inp()</code>”)</td></tr><tr><td>Bit 4</td><td>Motion has been aborted because the maximum positive position has been exceeded (see “<code>etm96_set_softlimit()</code>”)</td></tr><tr><td>Bit 5</td><td>Maximum negative position has been exceeded (see</td></tr></tbody></table>	Bit	Corresponding stop cause	Bit 0	Command has reached the target position. Motion has finished without any error	Bit 1	Automatic home search has finished (see “ <code>etm96_home_start()</code> ”)	Bit 2	Motion command has been interrupted by a stop command (“ <code>etm96_stop_move()</code> ”)	Bit 3	The axis finished outputting pulse commands and waits for the in-position signal of the servo drive (see “ <code>etm96_set_inp()</code> ”)	Bit 4	Motion has been aborted because the maximum positive position has been exceeded (see “ <code>etm96_set_softlimit()</code> ”)	Bit 5	Maximum negative position has been exceeded (see
Bit	Corresponding stop cause														
Bit 0	Command has reached the target position. Motion has finished without any error														
Bit 1	Automatic home search has finished (see “ <code>etm96_home_start()</code> ”)														
Bit 2	Motion command has been interrupted by a stop command (“ <code>etm96_stop_move()</code> ”)														
Bit 3	The axis finished outputting pulse commands and waits for the in-position signal of the servo drive (see “ <code>etm96_set_inp()</code> ”)														
Bit 4	Motion has been aborted because the maximum positive position has been exceeded (see “ <code>etm96_set_softlimit()</code> ”)														
Bit 5	Maximum negative position has been exceeded (see														

		“etm96_set_softlimit()”
	Bit 6	The positive limit switch has been activated (LMT+)
	Bit 7	The negative limit switch has been activated (LMT-)
	Bit 8	The servo drive alarm signal has been activated (ALM)
	Bit 9	The alarm has been activated (EMG)
	Bit 10 ~ Bit 15	reserved

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.1.2 etm96_get_motion_done_all

Read the current motion status of all six axes.

Syntax:

```

I16 etm96_get_motion_done_all ( HANDLE hEtm,
                                U8 CardNo,
                                U8* ArrDone,
                                U16* ArrStopStatus);

```

Parameters:

Name	Description																		
<i>hEtm:</i>	TCP/IP handle																		
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																		
<i>ArrDone:</i>	<p>Pointer to an array with 6 elements: U8 ArrDone[6] ; Each array element contains the done status of the corresponding axis.</p> <table border="1"> <thead> <tr> <th>Array Index</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>AXIS_0 done status</td> </tr> <tr> <td>[1]</td> <td>AXIS_1 done status</td> </tr> <tr> <td>[2]</td> <td>AXIS_2 done status</td> </tr> <tr> <td>[3]</td> <td>AXIS_3 done status</td> </tr> <tr> <td>[4]</td> <td>AXIS_4 done status</td> </tr> <tr> <td>[5]</td> <td>AXIS_5 done status</td> </tr> </tbody> </table> <p>0: a motion command is being executed (axis is outputting pulse) 1: motion has finished</p>	Array Index	Status	[0]	AXIS_0 done status	[1]	AXIS_1 done status	[2]	AXIS_2 done status	[3]	AXIS_3 done status	[4]	AXIS_4 done status	[5]	AXIS_5 done status				
Array Index	Status																		
[0]	AXIS_0 done status																		
[1]	AXIS_1 done status																		
[2]	AXIS_2 done status																		
[3]	AXIS_3 done status																		
[4]	AXIS_4 done status																		
[5]	AXIS_5 done status																		
<i>ArrStopStatus:</i>	<p>Pointer to an array with 6 elements: U16 ArrStopStatus[6] ; Each array element contains the stop status of the corresponding axis.</p> <table border="1"> <thead> <tr> <th>Array Index</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>AXIS_0 stop status</td> </tr> <tr> <td>[1]</td> <td>AXIS_1 stop status</td> </tr> <tr> <td>[2]</td> <td>AXIS_2 stop status</td> </tr> <tr> <td>[3]</td> <td>AXIS_3 stop status</td> </tr> <tr> <td>[4]</td> <td>AXIS_4 stop status</td> </tr> <tr> <td>[5]</td> <td>AXIS_5 stop status</td> </tr> </tbody> </table> <p>If StopStatus = 0 then motion command is still running</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Corresponding stop cause</th> </tr> </thead> <tbody> <tr> <td>Bit 0</td> <td>Command has reached the target position. Motion has</td> </tr> </tbody> </table>	Array Index	Status	[0]	AXIS_0 stop status	[1]	AXIS_1 stop status	[2]	AXIS_2 stop status	[3]	AXIS_3 stop status	[4]	AXIS_4 stop status	[5]	AXIS_5 stop status	Bit	Corresponding stop cause	Bit 0	Command has reached the target position. Motion has
Array Index	Status																		
[0]	AXIS_0 stop status																		
[1]	AXIS_1 stop status																		
[2]	AXIS_2 stop status																		
[3]	AXIS_3 stop status																		
[4]	AXIS_4 stop status																		
[5]	AXIS_5 stop status																		
Bit	Corresponding stop cause																		
Bit 0	Command has reached the target position. Motion has																		

	finished without any error
Bit 1	Automatic home search has finished (see “etm96_home_start()”)
Bit 2	Motion command has been interrupted by a stop command (“etm96_stop_move()”)
Bit 3	The axis finished outputting pulse commands and waits for the in-position signal of the servo drive (see “etm96_set_inp()”)
Bit 4	Motion has been aborted because the maximum positive position has been exceeded (see “etm96_set_softlimit()”)
Bit 5	Maximum negative position has been exceeded (see “etm96_set_softlimit()”)
Bit 6	The positive limit switch has been activated (LMT+)
Bit 7	The negative limit switch has been activated (LMT-)
Bit 8	The servo drive alarm signal has been activated (ALM)
Bit 9	The alarm has been activated (EMG)
Bit 10 ~ Bit 15	reserved

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.1.3 etm96_get_speed

Get the current axis speed.

Syntax:

```
I16 etm96_get_speed ( HANDLE hEtm,  
                    U8 CardNo,  
                    U8 AxisNo,  
                    I32* Speed);
```

Parameters:

Name	Description														
<i>hEtm</i> :	TCP/IP handle														
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo</i> :	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>Speed</i> :	Pointer to current axis speed (PPS)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.1.4 etm96_get_speed_all

Get the current speed of all six axes.

Syntax:

```
I16 etm96_get_speed_all ( HANDLE hEtm,  
                          U8 CardNo,  
                          I32* ArrSpeed);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>ArrSpeed:</i>	Pointer to an array with 6 elements: I32 ArrSpeed[6] ; Each array element contains the current speed of the corresponding axis. <table border="1" data-bbox="500 871 1127 1178"><thead><tr><th>Array Index</th><th>Status</th></tr></thead><tbody><tr><td>[0]</td><td>AXIS_0 speed</td></tr><tr><td>[1]</td><td>AXIS_1 speed</td></tr><tr><td>[2]</td><td>AXIS_2 speed</td></tr><tr><td>[3]</td><td>AXIS_3 speed</td></tr><tr><td>[4]</td><td>AXIS_4 speed</td></tr><tr><td>[5]</td><td>AXIS_5 speed</td></tr></tbody></table>	Array Index	Status	[0]	AXIS_0 speed	[1]	AXIS_1 speed	[2]	AXIS_2 speed	[3]	AXIS_3 speed	[4]	AXIS_4 speed	[5]	AXIS_5 speed
Array Index	Status														
[0]	AXIS_0 speed														
[1]	AXIS_1 speed														
[2]	AXIS_2 speed														
[3]	AXIS_3 speed														
[4]	AXIS_4 speed														
[5]	AXIS_5 speed														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.1.5 etm96_get_acc

Get the current axis acceleration.

Syntax:

```
I16 etm96_get_acc ( HANDLE hEtm,  
                   U8 CardNo,  
                   U8 AxisNo,  
                   I32* Acc);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>Acc:</i>	Pointer to current axis acceleration (PPS/Sec)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.1.6 etm96_get_acc_all

Get the current acceleration of all six axes.

Syntax:

```
I16 etm96_get_acc_all ( HANDLE hEtm,  
                       U8 CardNo,  
                       I32* ArrAcc);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>ArrAcc:</i>	Pointer to an array with 6 elements: I32 ArrAcc[6] ; Each array element contains the current acceleration of the corresponding axis. <table border="1" data-bbox="500 871 1127 1178"><thead><tr><th>Array Index</th><th>Status</th></tr></thead><tbody><tr><td>[0]</td><td>AXIS_0 acceleration</td></tr><tr><td>[1]</td><td>AXIS_1 acceleration</td></tr><tr><td>[2]</td><td>AXIS_2 acceleration</td></tr><tr><td>[3]</td><td>AXIS_3 acceleration</td></tr><tr><td>[4]</td><td>AXIS_4 acceleration</td></tr><tr><td>[5]</td><td>AXIS_5 acceleration</td></tr></tbody></table>	Array Index	Status	[0]	AXIS_0 acceleration	[1]	AXIS_1 acceleration	[2]	AXIS_2 acceleration	[3]	AXIS_3 acceleration	[4]	AXIS_4 acceleration	[5]	AXIS_5 acceleration
Array Index	Status														
[0]	AXIS_0 acceleration														
[1]	AXIS_1 acceleration														
[2]	AXIS_2 acceleration														
[3]	AXIS_3 acceleration														
[4]	AXIS_4 acceleration														
[5]	AXIS_5 acceleration														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.2 Single Axis Motion Commands

5.2.1 Introduction

This chapter describes the independent axis positioning motion commands. The motion between the specified axes is independent, and each axis follows its own profile. The user specifies the desired absolute position or relative position, acceleration ramp, and deceleration ramp, for each axis. Two speed profiles are being supported: trapezoidal t and s- curve:

T-Curve:

- The drive speed accelerates from the initial speed in a linear form with the specified acceleration slope to the constant driving speed. When the remaining number of output pulses becomes less than the deceleration pulses, deceleration starts. Deceleration continues until the initial speed has been reached and driving stops.

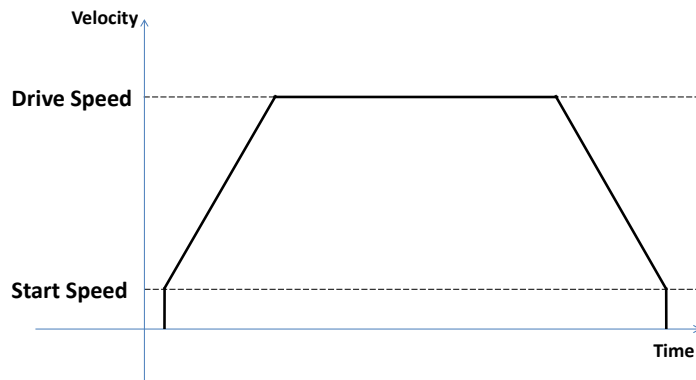


Figure 11: T-Curve velocity profile

S-Curve:

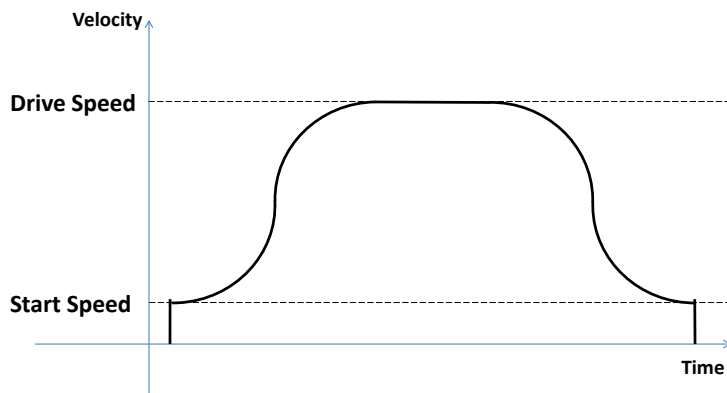


Figure 12: S-Curve velocity profile

5.2.2 etm96_t_move

Execute a single axis, relative position motion command with a trapezoidal velocity profile (T-curve). The *etm96_t_move* instruction moves the axis the specified travel distance from the current position.

Syntax:

```
I16 etm96_t_move ( HANDLE hEtm,
                   U8 CardNo,
                   U8 AxisNo,
                   U32 StartSpeed,
                   U32 DriveSpeed,
                   U32 EndSpeed,
                   U32 Acceleration,
                   U32 Deceleration,
                   I32 FixedPulse );
```

Parameters:

Name	Description														
<i>hEtm</i> :	TCP/IP handle														
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo</i> :	Axis definition: <table border="1" data-bbox="500 1071 846 1320"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed</i> :	Start speed (PPS)														
<i>DriveSpeed</i> :	Drive speed (PPS)														
<i>EndSpeed</i> :	End speed (PPS)														
<i>Acceleration</i> :	Acceleration (PPS/Sec)														
<i>Deceleration</i> :	Deceleration (PPS/Sec)														
<i>FixedPulse</i> :	Relative moving distance (Pulse) > 0: driving in positive direction < 0: driving in negative direction														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.2.3 etm96_abs_t_move

Execute a single axis, absolute position motion command with a trapezoidal velocity profile (T-curve). The *etm96_abs_t_move* instruction moves the axis to a specified absolute target position. You can execute this instruction even if home is not defined.

Syntax:

```
I16 etm96_abs_t_move ( HANDLE hEtm,
                       U8 CardNo,
                       U8 AxisNo,
                       U32 StartSpeed,
                       U32 DriveSpeed,
                       U32 EndSpeed,
                       U32 Acceleration,
                       U32 Deceleration,
                       I32 FixedPulse );
```

Parameters:

Name	Description														
<i>hEtm</i> :	TCP/IP handle														
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo</i> :	Axis definition: <table border="1" data-bbox="500 1077 846 1329"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed</i> :	Start speed (PPS)														
<i>DriveSpeed</i> :	Drive speed (PPS)														
<i>EndSpeed</i> :	End speed (PPS)														
<i>Acceleration</i> :	Acceleration (PPS/Sec)														
<i>Deceleration</i> :	Deceleration (PPS/Sec)														
<i>FixedPulse</i> :	Absolute position (Pulse)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- The direction is being determined by the relative position between the start position and absolute target position.

5.2.4 etm96_s_move

Execute a single axis motion command with an S-curve velocity profile. This command initiates a relative motion. When received, the selected axis will move, with the predefined acceleration and velocity, to a relative position from the current position.

Syntax:

```
I16 etm96_s_move ( HANDLE hEtm,
                   U8 CardNo,
                   U8 AxisNo,
                   U32 StartSpeed,
                   U32 DriveSpeed,
                   U32 EndSpeed,
                   U32 Acceleration,
                   U32 Deceleration,
                   I32 FixedPulse );
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1" data-bbox="500 1077 846 1329"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Start speed (PPS)														
<i>DriveSpeed:</i>	Drive speed (PPS)														
<i>EndSpeed:</i>	End speed (PPS)														
<i>Acceleration:</i>	Acceleration (PPS/Sec)														
<i>Deceleration:</i>	Deceleration (PPS/Sec)														
<i>FixedPulse:</i>	Relative moving distance (Pulse) > 0: driving in positive direction < 0: driving in negative direction														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.2.5 etm96_abs_s_move

Execute a single axis, absolute position motion command with an S-curve velocity profile.

Syntax:

```
I16 etm96_abs_s_move ( HANDLE hEtm,  
                        U8 CardNo,  
                        U8 AxisNo,  
                        U32 StartSpeed,  
                        U32 DriveSpeed,  
                        U32 EndSpeed,  
                        U32 Acceleration,  
                        U32 Deceleration,  
                        I32 FixedPulse );
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Start speed (PPS)														
<i>DriveSpeed:</i>	Drive speed (PPS)														
<i>EndSpeed:</i>	End speed (PPS)														
<i>Acceleration:</i>	Acceleration (PPS/Sec)														
<i>Deceleration:</i>	Deceleration (PPS/Sec)														
<i>FixedPulse:</i>	Absolute position (Pulse)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.2.6 etm96_velocity_move

Starts a single axis continues pulse driving. Once the axis has reached the driving speed it will indefinitely output pulses at a constant rate until a stop command has been encountered.

Syntax:

```
I16 etm96_velocity_move ( HANDLE hEtm,
                          U8 CardNo,
                          U8 AxisNo,
                          U32 StartSpeed,
                          U32 DriveSpeed,
                          U32 Acceleration,
                          U8 Direction);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1" data-bbox="500 1003 846 1255"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Start speed (PPS)														
<i>DriveSpeed:</i>	Drive speed (PPS)														
<i>Acceleration:</i>	Acceleration (PPS/Sec)														
<i>Direction:</i>	Driving direction: <table border="1" data-bbox="500 1402 1349 1512"> <thead> <tr> <th>Direction</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Negative</td> <td>(0)</td> </tr> <tr> <td>Positive</td> <td>(1)</td> </tr> </tbody> </table>	Direction	Value	Negative	(0)	Positive	(1)								
Direction	Value														
Negative	(0)														
Positive	(1)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.3 Two Axes Linear Interpolation Commands

In linear interpolation mode, motion between the axes is coordinated to maintain the prescribed vector speed, acceleration, and deceleration along the specified path.

5.3.1 `etm96_t_line2_move`

Executes a two axes relative distance linear interpolation motion command with a T-curve velocity profile. The `etm96_t_line2_move` instruction performs linear interpolation for two axes. The target position is specified as a relative position.

Syntax:

```

I16 etm96_t_line2_move ( HANDLE hEtm,
                          U8 CardNo,
                          U8 MainAxis,
                          U8 SlaveAxis,
                          U32 StartSpeed,
                          U32 DriveSpeed,
                          U32 EndSpeed,
                          U32 Acceleration,
                          U32 Deceleration,
                          I32 MainAxisRelDist,
                          I32 SlaveAxisRelDist,
                          U16 InterpMode);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
<i>SlaveAxis:</i>	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														

<i>MainAxisRelDist:</i>	Relative distance of the main axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction										
<i>SlaveAxisRelDist:</i>	Relative distance of the slave axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>(2)</td> </tr> <tr> <td>Packet</td> <td>(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored Use this mode to generate a continuous motion path. 	Reserved	(2)	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”).
Mode	Value										
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 										
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored Use this mode to generate a continuous motion path. 										
Reserved	(2)										
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). 										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.3.2 etm96_abs_t_line2_move

Executes a two axes absolute position interpolation motion command with a T-Curve velocity profile.

Syntax:

```

I16 etm96_abs_t_line2_move ( HANDLE hEtm,
                             U8 CardNo,
                             U8 MainAxis,
                             U8 SlaveAxis,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             I32 MainAxisFinishPoint,
                             I32 SlaveAxisFinishPoint,
                             U16 InterpMode);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
<i>SlaveAxis:</i>	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisFinishPoint:</i>	Absolute end position of the main axis (Pulse)														
<i>SlaveAxisFinishPoint:</i>	Absolute end position of the slave axis (Pulse)														
<i>InterpMode:</i>	Command execution mode: <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 										
Mode	Value														
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 														

	Continuous interpolation	(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
	Packet	(3) <ul style="list-style-type: none"> • Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()").

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.3.3 etm96_s_line2_move

Executes a two axes relative distance interpolation motion command with an S-Curve velocity profile.

Syntax:

```

I16 etm96_s_line2_move ( HANDLE hEtm,
                        U8 CardNo,
                        U8 MainAxis,
                        U8 SlaveAxis,
                        U32 StartSpeed,
                        U32 DriveSpeed,
                        U32 EndSpeed,
                        U32 Acceleration,
                        U32 Deceleration,
                        I32 MainAxisRelDist,
                        I32 SlaveAxisRelDist,
                        U16 InterpMode);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
<i>SlaveAxis:</i>	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisRelDist:</i>	Relative distance of the main axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction														
<i>SlaveAxisRelDist:</i>	Relative distance of the slave axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction														
<i>InterpMode:</i>	Command execution mode:														

Mode	Value
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished.
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path.
Reserved	CONTI_FIFO_BUFFER_MODE (2)
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”).

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.3.4 etm96_abs_s_line2_move

Executes a two axis absolute position interpolation motion command with an S-Curve velocity profile.

Syntax:

```

I16 etm96_abs_s_line2_move ( HANDLE hEtm,
                             U8 CardNo,
                             U8 MainAxis,
                             U8 SlaveAxis,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             I32 MainAxisFinishPoint,
                             I32 SlaveAxisFinishPoint,
                             U16 InterpMode);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
<i>SlaveAxis:</i>	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisFinishPoint:</i>	Absolute end position of the main axis (Pulse)														
<i>SlaveAxisFinishPoint:</i>	Absolute end position of the slave axis (Pulse)														
<i>InterpMode:</i>	Command execution mode: <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 										
Mode	Value														
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 														

	Continuous interpolation	(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
	Packet	(3) <ul style="list-style-type: none"> • Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()").

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.4 Three Axis Linear Interpolation Commands

5.4.1 etm96_t_line3_move

Executes a three axes linear interpolation motion command with a T-curve velocity profile. The target position is specified as a relative distance to the current position.

Syntax:

```
I16 etm96_t_line3_move ( HANDLE hEtm,
                          U8 CardNo,
                          U8 MainAxis,
                          U8 SecondAxis,
                          U8 ThirdAxis,
                          U32 StartSpeed,
                          U32 DriveSpeed,
                          U32 EndSpeed,
                          U32 Acceleration,
                          U32 Deceleration,
                          I32 MainAxisRelDist,
                          I32 SecondAxisRelDist,
                          I32 ThirdAxisRelDist,
                          U16 InterpMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>SecondAxis:</i>															
<i>ThirdAxis:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisRelDist:</i>	Relative distance of the main axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction														
<i>SecondAxisRelDist:</i>	Relative distance of the second axis (Pulse)														

	> 0: relative distance in positive direction < 0: relative distance in negative direction										
<i>ThirdAxisRelDist:</i>	Relative distance of the third axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td>(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “<code>etm96_send_conti_packet()</code>”). </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “<code>etm96_send_conti_packet()</code>”).
Mode	Value										
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 										
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 										
Reserved	CONTI_FIFO_BUFFER_MODE (2)										
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “<code>etm96_send_conti_packet()</code>”). 										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.4.2 etm96_abs_t_line3_move

Executes a three axes linear interpolation motion command with a T-curve velocity profile. This instruction moves the axes to an absolute position relative to (0) zero.

Syntax:

```
I16 etm96_abs_t_line3_move ( HANDLE hEtm,
                             U8 CardNo,
                             U8 MainAxis,
                             U8 SecondAxis,
                             U8 ThirdAxis,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             I32 MainAxisFinishPoint,
                             I32 SecondAxisFinishPoint,
                             I32 ThirdAxisFinishPoint,
                             U16 InterpMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>SecondAxis:</i>															
<i>ThirdAxis:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisFinishPoint:</i>	Absolute end position of the main axis (Pulse)														
<i>SecondAxisFinishPoint:</i>	Absolute end position of the second axis (Pulse)														
<i>ThirdAxisFinishPoint:</i>	Absolute end position of the third axis (Pulse)														
<i>InterpMode:</i>	Command execution mode:														
	<table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> </tbody> </table>	Mode	Value												
Mode	Value														

	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished.
	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”).

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.4.3 etm96_s_line3_move

Executes a three axes linear interpolation motion command with an S-curve velocity profile. The etm96_s_line3_move instruction moves the axes the specified travel distance from the current position.

Syntax:

```

I16 etm96_s_line3_move ( HANDLE hEtm,
                          U8 CardNo,
                          U8 MainAxis,
                          U8 SecondAxis,
                          U8 ThirdAxis,
                          U32 StartSpeed,
                          U32 DriveSpeed,
                          U32 EndSpeed,
                          U32 Acceleration,
                          U32 Deceleration,
                          I32 MainAxisRelDist,
                          I32 SecondAxisRelDist,
                          I32 ThirdAxisRelDist,
                          U16 InterpMode);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>SecondAxis:</i>															
<i>ThirdAxis:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisRelDist:</i>	Relative distance of the main axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction														
<i>SecondAxisRelDist:</i>	Relative distance of the second axis (Pulse) > 0: relative distance in positive direction														

	< 0: relative distance in negative direction										
<i>ThirdAxisRelDist:</i>	Relative distance of the third axis (Pulse) > 0: relative distance in positive direction < 0: relative distance in negative direction										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td>(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()"). </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()").
Mode	Value										
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 										
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 										
Reserved	CONTI_FIFO_BUFFER_MODE (2)										
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()"). 										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.4.4 etm96_abs_s_line3_move

Executes a three axes absolute position interpolation motion command with an S-curve velocity profile.

Syntax:

```
I16 etm96_abs_s_line3_move ( HANDLE hEtm,
                             U8 CardNo,
                             U8 MainAxis,
                             U8 SecondAxis,
                             U8 ThirdAxis,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             I32 MainAxisFinishPoint,
                             I32 SecondAxisFinishPoint,
                             I32 ThirdAxisFinishPoint,
                             U16 InterpMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>SecondAxis:</i>															
<i>ThirdAxis:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>MainAxisFinishPoint:</i>	Absolute end position of the main axis (Pulse)														
<i>SecondAxisFinishPoint:</i>	Absolute end position of the second axis (Pulse)														
<i>ThirdAxisFinishPoint:</i>	Absolute end position of the third axis (Pulse)														
<i>InterpMode:</i>	Command execution mode:														
	<table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> </tbody> </table>	Mode	Value												
Mode	Value														

	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished.
	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()").

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.5 Multi-Dimensional Linear Interpolation Commands

5.5.1 etm96_t_lines_move

Executes a multidimensional relative position motion command. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.

Syntax:

```

I16 etm96_t_lines_move ( HANDLE hEtm,
                          U8 CardNo,
                          U16 Axes,
                          U32 StartSpeed,
                          U32 DriveSpeed,
                          U32 EndSpeed,
                          U32 Acceleration,
                          U32 Deceleration,
                          const I32 RelativeDistance [6],
                          U16 InterpMode);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>Axes:</i>	For multiple axes select the corresponding axis bit combination: <table border="1" data-bbox="548 1192 894 1444"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table> <i>Axes</i> = AXIS_0 AXIS_2 AXIS_5 ;	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>RelativeDistance[]:</i>	Pointer to an array with 6 elements: I32 RelativeDistance [6]; Each array element contains the relative distance to move for the corresponding axis: > 0: relative distance in positive direction < 0: relative distance in negative direction														

	Array	Relative distance (Pulse)
	[0]	AXIS_0 relative moving distance
	[1]	AXIS_1 relative moving distance
	[2]	AXIS_2 relative moving distance
	[3]	AXIS_3 relative moving distance
	[4]	AXIS_4 relative moving distance
	[5]	AXIS_5 relative moving distance
<i>InterpMode:</i>	Command execution mode:	
	Mode	Value
	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished.
	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()"). 	

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

Example:

```

I16 ret;
U16 BitMultiAxes = (0x01 | 0x02 | 0x04 | 0x10); //Axis0,1,2,4
I32 FixedPulse[6] = {10000, 20000, 20000, 0, 10000, 0};

ret = etm96_t_lines_move(hEtM1, 1, BitMultiAxes, 0, 10000, 0,
20000, 20000, FixedPulse, 0);

if (ret != 0)
{MessageBox("Lines Move Error!!");}

```

5.5.2 etm96_abs_t_lines_move

Executes a multidimensional absolute position motion command. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.

Syntax:

```

I16 etm96_abs_t_lines_move ( HANDLE hEtm,
                             U8 CardNo,
                             U16 Axes,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             const I32 AbsolutePosition [6],
                             U16 InterpMode);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>Axes:</i>	For multiple axes select the corresponding axis bit combination: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table> <i>Axes</i> = AXIS_0 AXIS_2 AXIS_5 ;	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>AbsolutePosition []:</i>	Pointer to an array with 6 elements: I32 AbsolutePosition[6] ; Each array element contains the end position to which the corresponding axis has to move to: <table border="1"> <thead> <tr> <th>Array</th> <th>Absolute Position (Pulse)</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>AXIS_0 absolute position</td> </tr> <tr> <td>[1]</td> <td>AXIS_1 absolute position</td> </tr> <tr> <td>[2]</td> <td>AXIS_2 absolute position</td> </tr> </tbody> </table>	Array	Absolute Position (Pulse)	[0]	AXIS_0 absolute position	[1]	AXIS_1 absolute position	[2]	AXIS_2 absolute position						
Array	Absolute Position (Pulse)														
[0]	AXIS_0 absolute position														
[1]	AXIS_1 absolute position														
[2]	AXIS_2 absolute position														

	[3]	AXIS_3 absolute position
	[4]	AXIS_4 absolute position
	[5]	AXIS_5 absolute position
<i>InterpMode:</i>	Command execution mode:	
	Mode	Value
	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished.
	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). 	

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.5.3 etm96_s_lines_move

Executes a multidimensional relative position motion command with a s-curve velocity profile. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.

Syntax:

```

116 etm96_s_lines_move ( HANDLE hEtm,
                        U8 CardNo,
                        U16 Axes,
                        U32 StartSpeed,
                        U32 DriveSpeed,
                        U32 EndSpeed,
                        U32 Acceleration,
                        U32 Deceleration,
                        const I32 RelativeDistance [6],
                        U16 InterpMode);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>Axes:</i>	For multiple axes select the corresponding axis bit combination: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table> <i>Axes</i> = AXIS_0 AXIS_2 AXIS_5 ;	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>RelativeDistance[]:</i>	Pointer to an array with 6 elements: I32 RelativeDistance [6]; Each array element contains the relative distance to move for the corresponding axis: > 0: relative distance in positive direction < 0: relative distance in negative direction <table border="1"> <thead> <tr> <th>Array</th> <th>Relative distance (Pulse)</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>AXIS_0 relative moving distance</td> </tr> </tbody> </table>	Array	Relative distance (Pulse)	[0]	AXIS_0 relative moving distance										
Array	Relative distance (Pulse)														
[0]	AXIS_0 relative moving distance														

	[1]	AXIS_1 relative moving distance										
	[2]	AXIS_2 relative moving distance										
	[3]	AXIS_3 relative moving distance										
	[4]	AXIS_4 relative moving distance										
	[5]	AXIS_5 relative moving distance										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td>(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()"). </td> </tr> </tbody> </table>		Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()").
Mode	Value											
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 											
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 											
Reserved	CONTI_FIFO_BUFFER_MODE (2)											
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()"). 											

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.5.4 etm96_abs_s_lines_move

Executes a multidimensional absolute position motion command using the s-curve velocity profile. Positioning is performed on up to six axes with linear interpolation at the specified interpolation speed. The number of interpolation axes can be selected.

Syntax:

```
I16 etm96_abs_s_lines_move ( HANDLE hEtm,
                             U8 CardNo,
                             U16 Axes,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             const I32 AbsolutePosition [6],
                             U16 InterpMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>Axes:</i>	For multiple axes select the corresponding axis bit combination: <table border="1" data-bbox="548 1115 894 1367"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table> <i>Axes</i> = <i>AXIS_0</i> <i>AXIS_2</i> <i>AXIS_5</i> ;	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>AbsolutePosition []:</i>	Pointer to an array with 6 elements: I32 <i>AbsolutePosition</i> [6] ; Each array element contains the end position to which the corresponding axis has to move to: <table border="1" data-bbox="548 1759 1263 1900"> <thead> <tr> <th>Array</th> <th>Absolute Position (Pulse)</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>AXIS_0 absolute position</td> </tr> <tr> <td>[1]</td> <td>AXIS_1 absolute position</td> </tr> <tr> <td>[2]</td> <td>AXIS_2 absolute position</td> </tr> </tbody> </table>	Array	Absolute Position (Pulse)	[0]	AXIS_0 absolute position	[1]	AXIS_1 absolute position	[2]	AXIS_2 absolute position						
Array	Absolute Position (Pulse)														
[0]	AXIS_0 absolute position														
[1]	AXIS_1 absolute position														
[2]	AXIS_2 absolute position														

	[3]	AXIS_3 absolute position
	[4]	AXIS_4 absolute position
	[5]	AXIS_5 absolute position
<i>InterpMode:</i>	Command execution mode:	
	Mode	Value
	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished.
	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path.
	Reserved	CONTI_FIFO_BUFFER_MODE (2)
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). 	

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.6 Two Dimensional Circular Interpolation Functions

5.6.1 `etm96_t_arc2_move`

Performs circular interpolation for two axes with a T-curve velocity profile. The center and end position are specified relative to the current position.

Syntax:

```

I16 etm96_t_arc2_move ( HANDLE hEtm,
                        U8 CardNo,
                        U8 MainAxis,
                        U8 SlaveAxis,
                        U32 StartSpeed,
                        U32 DriveSpeed,
                        U32 EndSpeed,
                        U32 Acceleration,
                        U32 Deceleration,
                        U8 ArcDirection,
                        I32 MainAxisCenterPoint,
                        I32 SlaveAxisCenterPoint,
                        I32 MainAxisFinishPoint,
                        I32 SlaveAxisFinishPoint,
                        U16 InterpMode);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>SlaveAxis:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>ArcDirection:</i>	Rotation direction														
	<table border="1"> <thead> <tr> <th>Direction</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>CW</td> <td>(0)</td> </tr> </tbody> </table>	Direction	Value	CW	(0)										
Direction	Value														
CW	(0)														

	CCW	(1)										
<i>MainAxisCenterPoint:</i>	Relative center point of the main axis (Pulse)											
<i>SlaveAxisCenterPoint:</i>	Relative center point of the slave axis (Pulse)											
<i>MainAxisFinishPoint:</i>	Relative end point of the main axis (Pulse)											
<i>SlaveAxisFinishPoint:</i>	Relative end point of the slave axis (Pulse)											
<i>InterpMode:</i>	Command execution mode: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Mode</th> <th style="width: 70%;">Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td> (0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td> (1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td> (3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). </td> </tr> </tbody> </table>		Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”).
Mode	Value											
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 											
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 											
Reserved	CONTI_FIFO_BUFFER_MODE (2)											
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). 											

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

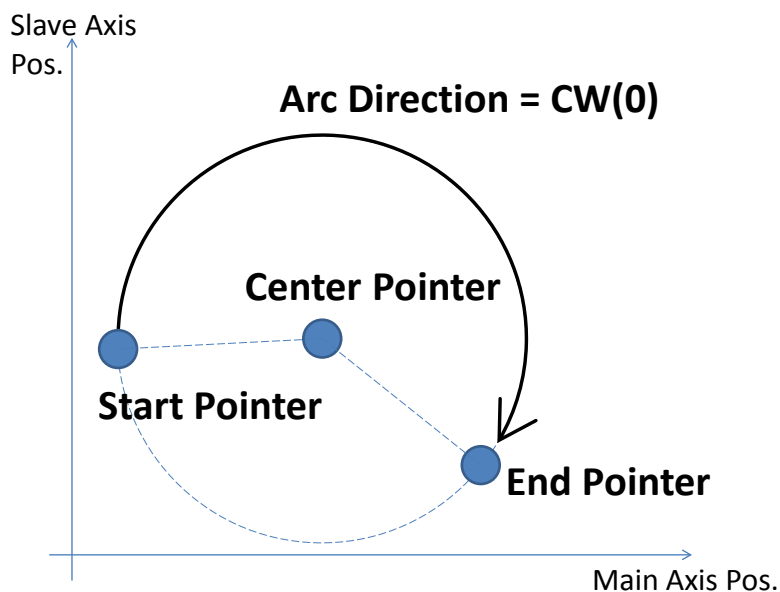


Figure 13: Clockwise circular interpolation

5.6.2 etm96_abs_t_arc2_move

Executes a two axes circular interpolation motion command with a T-curve velocity profile. The center and target position are specified in absolute position.

Syntax:

```
I16 etm96_abs_t_arc2_move ( HANDLE hEtm,
                             U8 CardNo,
                             U8 MainAxis,
                             U8 SlaveAxis,
                             U32 StartSpeed,
                             U32 DriveSpeed,
                             U32 EndSpeed,
                             U32 Acceleration,
                             U32 Deceleration,
                             U8 ArcDirection,
                             I32 MainAxisCenterPoint,
                             I32 SlaveAxisCenterPoint,
                             I32 MainAxisFinishPoint,
                             I32 SlaveAxisFinishPoint,
                             U16 InterpMode);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MainAxis:</i>	Axis definition:														
<i>SlaveAxis:</i>	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)														
<i>ArcDirection:</i>	Rotation direction														
	<table border="1"> <thead> <tr> <th>Direction</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>CW</td> <td>(0)</td> </tr> <tr> <td>CCW</td> <td>(1)</td> </tr> </tbody> </table>	Direction	Value	CW	(0)	CCW	(1)								
Direction	Value														
CW	(0)														
CCW	(1)														
<i>MainAxisCenterPoint:</i>	Absolute center point of the main axis (Pulse)														

<i>SlaveAxisCenterPoint:</i>	Absolute center point of the slave axis (Pulse)										
<i>MainAxisFinishPoint:</i>	Absolute end point of the main axis (Pulse)										
<i>SlaveAxisFinishPoint:</i>	Absolute end point of the slave axis (Pulse)										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td>(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “<code>etm96_send_conti_packet()</code>”). </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “<code>etm96_send_conti_packet()</code>”).
Mode	Value										
General	(0) <ul style="list-style-type: none"> No command buffering takes place. A new command can only be executed if the previous command has finished. 										
Continuous interpolation	(1) <ul style="list-style-type: none"> Command buffering; up to 5000 command can be stored. Use this mode to generate a continuous motion path. 										
Reserved	CONTI_FIFO_BUFFER_MODE (2)										
Packet	(3) <ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “<code>etm96_send_conti_packet()</code>”). 										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.7 Three Dimensional Helical Interpolation Functions

5.7.1 etm96_t_helical_move

Executes a three dimensional helical (screw like) movement. The movement is a composition of a rotation by an angle about a helical axis with a translation distance along this axis. The circle center position is the relative distance from the current position.


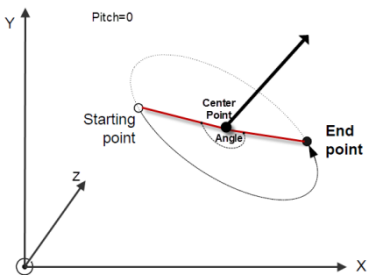
Syntax:

```

I16 etm96_t_helical_move ( HANDLE hEtm,
                           U8 CardNo,
                           U8 Axis0,
                           U8 Axis1,
                           U8 Axis2,
                           U32 StartSpeed,
                           U32 DriveSpeed,
                           U32 EndSpeed,
                           U32 Acceleration,
                           U32 Deceleration,
                           I32 CenterPoint0,
                           I32 CenterPoint1,
                           I32 CenterPoint2,
                           I32 NormalDir0,
                           I32 NormalDir1,
                           I32 NormalDir2,
                           F32 Angle,
                           F32 Pitch,
                           U16 InterpMode);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>Axis0:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>Axis1:</i>															
<i>Axis2:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														

<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)										
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)										
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)										
<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)										
<i>CenterPoint0:</i>	Relative center point of Axis0 (Pulse)										
<i>CenterPoint1:</i>	Relative center point of Axis1 (Pulse)										
<i>CenterPoint2:</i>	Relative center point of Axis2 (Pulse)										
<i>NormalDir0:</i>	<p>The parameters describe a three dimensional rotation vector (helical axis).</p> <p>The rotation vector determines:</p> <ul style="list-style-type: none"> • The positive direction of the translational movement (pitch direction). • The positive direction of rotation. The positive direction of rotation is being determined by the right-hand rule: the right thumb points along the positive direction of the rotation axis and the curl of your fingers represents the direction of rotation 										
<i>NormalDir1:</i>											
<i>NormalDir2:</i>											
<i>Angle:</i>	<p>Rotation angle in degrees (360 indicates one full revolution, 720 will result in two full revolution, etc.)</p> <p>> 0: positive direction rotation vector (right-hand rule)</p> <p>< 0: negative direction of the rotation vector</p>										
<i>Pitch:</i>	<p>The relative distance to move along the helical axis after each full revolution.</p> <p>If this parameter is zero a three dimensional arc will be executed</p> 										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> • No command buffering takes place. • A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td>(3)</td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> • No command buffering takes place. • A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3)
Mode	Value										
General	(0) <ul style="list-style-type: none"> • No command buffering takes place. • A new command can only be executed if the previous command has finished. 										
Continuous interpolation	(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path. 										
Reserved	CONTI_FIFO_BUFFER_MODE (2)										
Packet	(3)										

		<ul style="list-style-type: none"> Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see "etm96_send_conti_packet()").
--	--	--

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

Example:

The Figure 14 shows a helical path which helical direction is perpendicular to the Axis0 - Axis1 plane:

(CenterPoint2 = 0, NormalDir0 = 0, NormalDir1 = 0, NormalDir2 = 1)

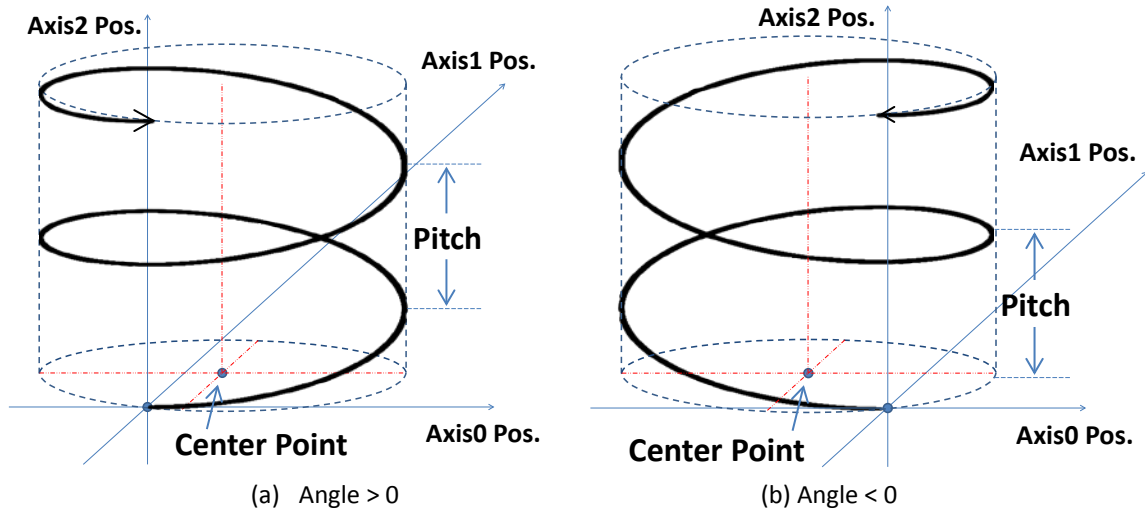


Figure 14: Helical axis is perpendicular to the Axis0 and Axis1 plane

5.7.2 etm96_abs_t_helical_move

Executes a three dimensional helical (screw like) movement. The movement is a composition of a rotation by an angle about a helical axis with a translation distance along this axis. The circle center position is an absolute position.

Syntax:


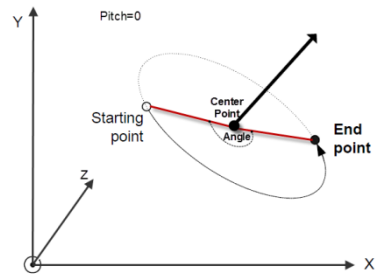
```

I16 etm96_abs_t_helical_move ( HANDLE hEtm,
                                U8 CardNo,
                                U8 Axis0,
                                U8 Axis1,
                                U8 Axis2,
                                U32 StartSpeed,
                                U32 DriveSpeed,
                                U32 EndSpeed,
                                U32 Acceleration,
                                U32 Deceleration,
                                I32 CenterPoint0,
                                I32 CenterPoint1,
                                I32 CenterPoint2,
                                I32 NormalDir0,
                                I32 NormalDir1,
                                I32 NormalDir2,
                                F32 Angle,
                                F32 Pitch,
                                U16 InterpMode);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>Axis0:</i>	Axis definition:														
	<table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>Axis1:</i>															
<i>Axis2:</i>															
<i>StartSpeed:</i>	Magnitude of start velocity vector (PPS)														
<i>DriveSpeed:</i>	Magnitude of drive velocity vector (PPS)														
<i>EndSpeed:</i>	Magnitude of end velocity vector (PPS)														
<i>Acceleration:</i>	Magnitude of acceleration vector (PPS/Sec)														

<i>Deceleration:</i>	Magnitude of deceleration vector (PPS/Sec)										
<i>CenterPoint0:</i>	Absolute center point of Axis0 (Pulse)										
<i>CenterPoint1:</i>	Absolute center point of Axis1 (Pulse)										
<i>CenterPoint2:</i>	Absolute center point of Axis2 (Pulse)										
<i>NormalDir0:</i>	The parameters describe a three dimensional rotation vector.										
<i>NormalDir1:</i>	The rotation vector determines:										
<i>NormalDir2:</i>	<ul style="list-style-type: none"> • The positive direction of the translational movement (pitch direction). • The positive direction of rotation. The positive direction of rotation is being determined by the right-hand rule: the right thumb points along the positive direction of the rotation axis and the curl of your fingers represents the direction of rotation 										
<i>Angle:</i>	<p>Rotation angle in degrees (360 indicates one full revolution, 720 will result in two full revolution, etc.)</p> <p>> 0: positive direction rotation vector (right-hand rule)</p> <p>< 0: negative direction of the rotation vector</p>										
<i>Pitch:</i>	<p>The relative distance to move along the helical axis after each full revolution.</p> <p>If this parameter is zero a three dimensional arc will be executed</p> 										
<i>InterpMode:</i>	<p>Command execution mode:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>General</td> <td>(0) <ul style="list-style-type: none"> • No command buffering takes place. • A new command can only be executed if the previous command has finished. </td> </tr> <tr> <td>Continuous interpolation</td> <td>(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path. </td> </tr> <tr> <td>Reserved</td> <td>CONTI_FIFO_BUFFER_MODE (2)</td> </tr> <tr> <td>Packet</td> <td>(3) <ul style="list-style-type: none"> • Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). </td> </tr> </tbody> </table>	Mode	Value	General	(0) <ul style="list-style-type: none"> • No command buffering takes place. • A new command can only be executed if the previous command has finished. 	Continuous interpolation	(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path. 	Reserved	CONTI_FIFO_BUFFER_MODE (2)	Packet	(3) <ul style="list-style-type: none"> • Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”).
Mode	Value										
General	(0) <ul style="list-style-type: none"> • No command buffering takes place. • A new command can only be executed if the previous command has finished. 										
Continuous interpolation	(1) <ul style="list-style-type: none"> • Command buffering; up to 5000 command can be stored. • Use this mode to generate a continuous motion path. 										
Reserved	CONTI_FIFO_BUFFER_MODE (2)										
Packet	(3) <ul style="list-style-type: none"> • Write multiple continuous interpolation commands in one TCP/IP packet before transmitting it to the remote ET-M8196F controller (see “etm96_send_conti_packet()”). 										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.8 Continuous Interpolation Functions

5.8.1 etm96_set_conti_interp_cfg

Assigns axes to an interpolation group and sets the axes group to continuous interpolation mode (see Figure 5, Figure 6 and Figure 7). Once the group has switch to continuous mode, all the arriving commands are being treated as continuous interpolation commands.

In continuous interpolation mode more than one command can be sent at a time. If a new command is being sent while the previous commands is still executing, then the arriving command will first be written to the internal FIFO buffer and starts to executed once the running command has finished. Up to 5000 commands can be stored in the FIFO buffer.

Syntax:

```

I16 etm96_set_conti_interp_cfg ( HANDLE hEtm,
                                U8 CardNo,
                                U8 CfgEnable,
                                U16 GroupIndex0,
                                U16 GroupIndex1,
                                U16 GroupIndex2);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>CfgEnable:</i>	Enable continuous interpolation: <table border="1"> <thead> <tr> <th>Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Disable</td> <td>(0)</td> </tr> <tr> <td>Enable</td> <td>(1)</td> </tr> </tbody> </table>	Mode	Value	Disable	(0)	Enable	(1)								
Mode	Value														
Disable	(0)														
Enable	(1)														
<i>GroupIndex0:</i>	Select the axis which belongs to the first interpolation group <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table> <p>Example: GroupIndex0 = AXIS_0 AXIS_1;</p>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>GroupIndex1:</i>	Select the axis which belongs to the second interpolation group Example: GroupIndex1 = AXIS_3 AXIS_5;														
<i>GroupIndex2:</i>	Select the axis which belongs to the third interpolation group														


```
Example:  
GroupIndex2 = AXIS_2 | AXIS_4;
```

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- If the user first want to fill the command FIFO buffer before starting to execute the motion commands then follow the following steps (see example 2):
 1. first call “etm96_drv_hold” to hold the next command from being executed
 2. Fill the command buffer with commands
 3. Call “etm96_drv_start” to start executing the command in the buffer
- The start and end speed of each command has to be lower or equal to the driving speed (Figure 15). Figure 16 shows velocity profiles which are currently not supported by continuous interpolation mode.

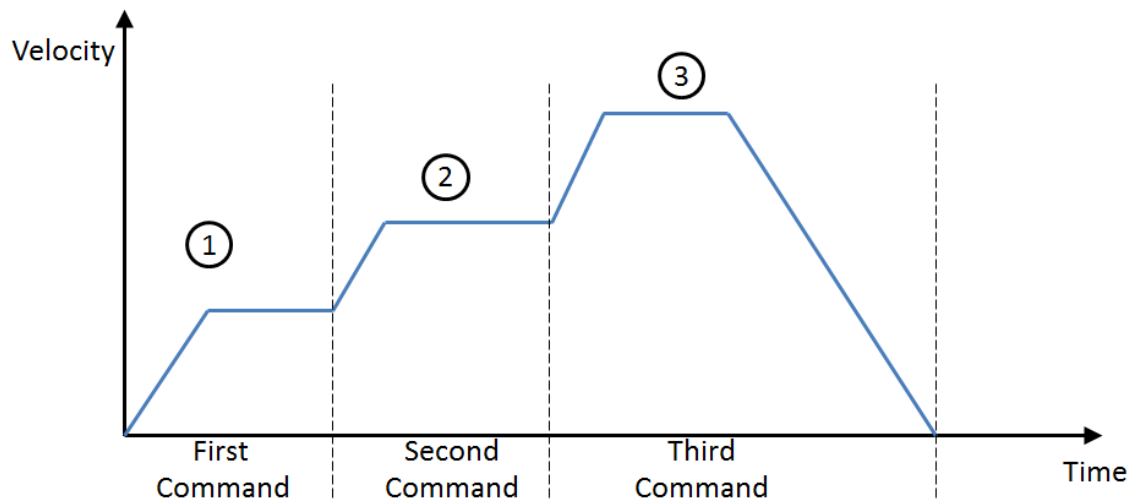


Figure 15: Velocity profile supported in continuous interpolation mode

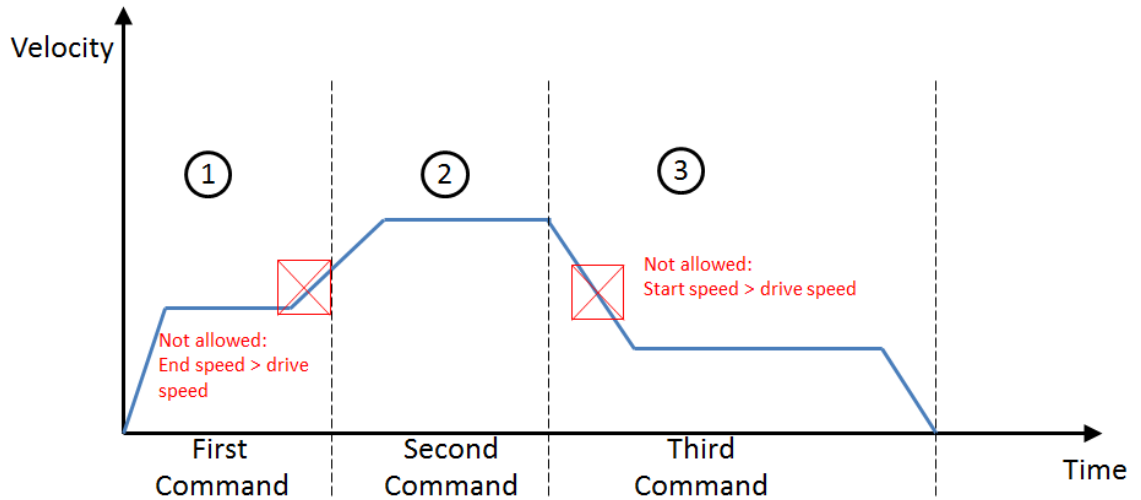


Figure 16: Velocity profiles which are not supported

Example:

Example 1:

```

//=====
// Three Dimensional Continue Interpolation
//=====
I16 eRet;
HANDLE hEtm;
U32 dwStartSpeed, dwDriveSpeed, dwEndSpeed, dwAcceleration,
dwDeceleration;
U16 wGroupIndex0;
U8 bDone = 0;
U16 wStopStatus;
I32 x, y, z;
U8 bSlot = 1;

dwStartSpeed    = 1000;
dwDriveSpeed    = 20000;
dwEndSpeed      = 1000;
dwAcceleration  = 5000;
dwDeceleration  = 5000;
x = 100;
y = 100;
z = 100;

//Assign three axis to a continuous interpolation group:
wGroupIndex0 = AXIS_0|AXIS_1|AXIS_2;
eRet = etm96_set_conti_interp_cfg(hEtm, bSlot, 1, wGroupIndex0, 0,
0 );

//Write 120 interpolation commands to the buffer for execution
// Once a command arrives at an empty buffer it will be executed
for(int i=0; i<120; i++) //120
{
    x *= i;
    y *= x;
}

```

```

    z *= 100;
    eRet = etm96_abs_t_line3_move(hEtm, bSlot,
        AXIS_0, AXIS_1, AXIS_2,
        dwStartSpeed, dwDriveSpeed, dwEndSpeed, dwAcceleration,
        dwDeceleration, x, y, z, CONTI_INTERP_CONFIG);
}

//Wait until the continuous interpolation command have been executed:
while( bDone != 1)
{
    eRet = etm96_get_motion_done(hEtm, bSlot, AXIS_0, &bDone,
    &wStopStatus);
    ::Sleep(10);
}

//Disable the continuous interpolation mode:
eRet = etm96_set_conti_interp_cfg( hEtm, bSlot, 0, wGroupIndex0, 0,
0 );

```

Example 2:

```

//=====
// Three Dimensional Continue Interpolation
//=====
I16 eRet;
U32 dwStartSpeed, dwDriveSpeed, dwEndSpeed, dwAcceleration,
dwDeceleration;
U16 wGroupIndex0;
U8 bDone = 0;
U16 wStopStatus;
I32 x, y, z;
U8 bSlot = 1;

dwStartSpeed = 1000;
dwDriveSpeed = 20000;
dwEndSpeed = 1000;
dwAcceleration = 5000;
dwDeceleration = 5000;
x = 100;
y = 100;
z = 100;

//Create two continuous interpolation groups:
wGroupIndex0 = AXIS_0|AXIS_1|AXIS_2;
wGroupIndex1 = AXIS_3|AXIS_4|AXIS_5;
eRet = etm96_set_conti_interp_cfg(hEtm, bSlot, 1, wGroupIndex0,
wGroupIndex1, 0 );

//Put the groups into hold mode
eRet = etm96_drv_hold(hEtm, bSlot,
AXIS_0|AXIS_1|AXIS_2|AXIS_3|AXIS_4|AXIS_5 );

//Write 120 interpolation commands to the buffer for execution
for(int i=0; i<120; i++) //120
{
    x *= i;

```

```

y *= x;
z *= 100;
eRet = etm96_abs_t_line3_move(hEtm, bSlot,
    AXIS_0, AXIS_1, AXIS_2,
    dwStartSpeed, dwDriveSpeed, dwEndSpeed, dwAcceleration,
    dwDeceleration, x, y, z, CONTI_INTERP_CONFIG);

eRet = etm96_abs_t_line3_move(hEtm, bSlot,
    AXIS_3, AXIS_4, AXIS_5,
    dwStartSpeed, dwDriveSpeed, dwEndSpeed, dwAcceleration,
    dwDeceleration, x, y, z, CONTI_INTERP_CONFIG);
}

//Start the execution of the commands in the buffer:
eRet = etm96_drv_start(hEtm, bSlot,
    AXIS_0|AXIS_1|AXIS_2|AXIS_3|AXIS_4|AXIS_5);

//Wait for the first continuous interpolation group to finish:
while( bDone != 1)
{
    eRet = etm96_get_motion_done(hEtm, bSlot, AXIS_0, &bDone,
    &wStopStatus);
    ::Sleep(10);
}
//Wait for the second continuous interpolation group to finish:
while( bDone != 1)
{
    eRet = etm96_get_motion_done(hEtm, bSlot, AXIS_3, &bDone,
    &wStopStatus);
    ::Sleep(10);
}

//Disable the continuous interpolation mode:
eRet = etm96_set_conti_interp_cfg(hEtm, bSlot, 0, wGroupIndex0,
    wGroupIndex1, 0 );

```

5.8.2 etm96_send_conti_packet

Transmits a TCP/IP packet with multiple continuous interpolation commands to the remote ET-M8196F. Select the packet mode in order to add multiple commands in one TCP/IP packet.

Syntax:

```
I16 etm96_send_conti_packet ( HANDLE hEtm,  
                             U8 CardNo);
```

Parameters:

Name	Description
<i>hEtm</i> :	TCP/IP handle
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

```
I16 ret = 0;  
U8 AxisX = 0x01;  
U8 AxisY = 0x02;  
U16 MultiAxis = AxisX | AxisY;  
I32 BaseDist = 10000;  
U32 Speed = 10000;  
U32 Acc = 20000, Dec = 20000;  
U8 ArcDir = 1;  
U16 InterpMode = 3; //CONTI_SEND_MULTI_CMD  
  
ret = etm96_clear_conti_packet( hEtm1, 1 );  
  
ret = etm96_t_line2_move( hEtm1, 1, AxisX, AxisY, Speed/10, Speed,  
Speed, Acc, Dec, BaseDist*2, 0, InterpMode );  
ret = etm96_t_arc2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed, Speed,  
Acc, Dec, ArcDir, 0, BaseDist, BaseDist, BaseDist, InterpMode);  
ret = etm96_t_line2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed,  
Speed, Acc, Dec, 0, BaseDist, InterpMode );  
ret = etm96_t_arc2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed, Speed,  
Acc, Dec, ArcDir, BaseDist*(-1), 0, BaseDist*(-1), BaseDist,  
InterpMode);  
ret = etm96_t_line2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed,  
Speed, Acc, Dec, BaseDist*(-2), 0, InterpMode );  
ret = etm96_t_arc2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed, Speed,  
Acc, Dec, ArcDir, 0, BaseDist*(-1), BaseDist*(-1), BaseDist*(-1),  
InterpMode);  
ret = etm96_t_line2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed,  
Speed, Acc, Dec, 0, BaseDist*(-1), InterpMode );  
ret = etm96_t_arc2_move( hEtm1, 1, AxisX, AxisY, Speed, Speed,
```

```
Speed/10, Acc, Dec, ArcDir, BaseDist, 0, BaseDist, BaseDist*(-1),  
InterpMode);  
ret = etm96_conti_interp_config(hEtm1, 1, TRUE, MultiAxis, 0, 0);  
ret = etm96_send_conti_packet(hEtm1, 1);  
ret = etm96_conti_interp_config(hEtm1, 1, FALSE, MultiAxis, 0, 0);
```

5.8.3 etm96_clear_conti_packet

Clears all the continuous interpolation commands which have been written to the TCP/IP packet so far for sending to the ET-M8196F.

Syntax:

```
I16 etm96_clear_conti_packet ( HANDLE hEtm,  
                               U8 CardNo );
```

Parameters:

Name	Description
<i>hEtm:</i>	TCP/IP handle
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.8.4 etm96_get_conti_packet_size

Clears all the continuous interpolation commands which have been written to the TCP/IP packet so far for sending to the ET-M8196F.

Syntax:

```
U16 etm96_get_conti_packet_size ( HANDLE hEtm,
                                U8 CardNo,
                                U16* Length );
```

Parameters:

Name	Description
<i>hEtm:</i>	TCP/IP handle
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)
<i>Length:</i>	The current TCP/IP packet data length (bytes) The maximum value is 1300. The data length required for each functions can be obtained from Table 2.

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

Function	Data length (Bytes)
etm96_t_line2_move	38
etm96_abs_t_line2_move	
etm96_s_line2_move	
etm96_abs_s_line2_move	
etm96_t_line3_move	44
etm96_abs_t_line3_move	
etm96_s_line3_move	
etm96_abs_s_line3_move	
etm96_t_lines_move	54
etm96_abs_t_lines_move	
etm96_s_lines_move	
etm96_abs_s_lines_move	
etm96_t_arc2_move	48
etm96_abs_t_arc2_move	
etm96_t_helical_move	(Currently no supported)
etm96_abs_t_helical_move	

Table 2: TCP/IP Continuous interpolation command size

5.9 Motion Stop Functions

5.9.1 etm96_stop_move

Stops the current executing motion command for the specified axis. Stops motion before reaching the destination.

Syntax:

```
I16 etm96_stop_move ( HANDLE hEtm,  
                      U8 CardNo,  
                      U8 AxisNo,  
                      U16 StopMode);
```

Parameters:

Name	Description														
<i>hEtm</i> :	TCP/IP handle														
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo</i> :	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StopMode</i> :	Stop mode <table border="1"><thead><tr><th>Mode</th><th>Value</th></tr></thead><tbody><tr><td>Deceleration stop</td><td>(1)</td></tr><tr><td>Sudden stop</td><td>(2)</td></tr></tbody></table>	Mode	Value	Deceleration stop	(1)	Sudden stop	(2)								
Mode	Value														
Deceleration stop	(1)														
Sudden stop	(2)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.9.2 etm96_set_softlimit

Sets the software limits for the positive and negative direction. Once a software limit position is specified, the ET-M8196F will not accept position commands beyond the limit and motion will stop once the limit is hit.

Syntax:

```

I16 etm96_set_softlimit ( HANDLE hEtm,
                          U8 CardNo,
                          U8 AxisNo,
                          U16 StopMode,
                          U16 RefSource,
                          I32 LimitPositive,
                          I32 LimitNegative);

```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>StopMode:</i>	Stop mode <table border="1"> <thead> <tr> <th>Stop Mode</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Deceleration stop</td> <td>(1)</td> </tr> <tr> <td>Sudden stop</td> <td>(2)</td> </tr> </tbody> </table>	Stop Mode	Value	Deceleration stop	(1)	Sudden stop	(2)								
Stop Mode	Value														
Deceleration stop	(1)														
Sudden stop	(2)														
<i>RefSource:</i>	Position counter source <table border="1"> <thead> <tr> <th>Source</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Encoder Pulse Counter</td> <td>(1)</td> </tr> <tr> <td>Commanded Position Pulse Counter</td> <td>(2)</td> </tr> </tbody> </table>	Source	Value	Encoder Pulse Counter	(1)	Commanded Position Pulse Counter	(2)								
Source	Value														
Encoder Pulse Counter	(1)														
Commanded Position Pulse Counter	(2)														
<i>LimitPositive:</i>	Positive direction soft limit														
<i>LimitNegative:</i>	Negative direction soft limit														

Return:

- 0: Success
- Others: Error (refer to error documentation)

5.9.3 etm96_set_softlimit_disable

Disables the axis limits settings.

Syntax:

```
I16 etm96_set_softlimit_disable ( HANDLE hEtm,  
                                U8 CardNo,  
                                U8 AxisNo);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

5.10 Multi-Axis Hold/Release Functions

5.10.1 etm96_drv_hold

This command sets the specified axes in holding mode after the current running command has reached its target position. Therefore this instruction takes effect for the next command. The execution of the next command will be put on hold until the “etm96_drv_start()” releases the hold operation.

Syntax:

```
I16 etm96_drv_hold ( HANDLE hEtm,  
                    U8 CardNo,  
                    U16 BitMultiAxes);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>BitMultiAxes:</i>	Each bit in the variable represents an axis: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- After calling “etm96_drv_hold()” the current running interpolation command will finish first, but the next command will not start until “etm96_drv_start()” has been called.
- The command will hold the interpolation group if only one or more of the hold axes (*BitMultiAxes*) belongs to the group.

5.10.2 etm96_drv_start

Terminates the axes hold operation. Axes which have been put on hold by “etm96_drv_hold()” will continue to execute the next motion command stored in the command FIFO buffer.

Syntax:

```
I16 etm96_drv_start ( HANDLE hEtm,  
                    U8 CardNo,  
                    U16 BitMultiAxes);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>BitMultiAxes:</i>	Each bit of the variable represents an axis: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

6 Other Functions

6.1 Compare Function

6.1.1 etm96_set_compare_trig_cfg

Configures and enables the compare trigger function. The compare function outputs a signal when the compare condition has been met. Two compare modes are being supported:

1. One time compare mode (Single compare mode)
2. Auto increment compare mode.

Syntax:

I16 etm96_set_compare_trig_cfg (HANDLE hEtm, U8 CardNo, U8 AxisNo, U8 CmpTrigEnable, U8 OutputLogic, U16 PulseWidth, U8 MoveDirection, U8 CmpIncEnable, U16 ConstPitch, I32 CmpData);
---	--

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>CmpTrigEnable:</i>	Enable compare function <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>State</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Disable</td> <td>(0x00)</td> </tr> <tr> <td>Enable</td> <td>(0x01)</td> </tr> </tbody> </table>	State	Value	Disable	(0x00)	Enable	(0x01)								
State	Value														
Disable	(0x00)														
Enable	(0x01)														
<i>OutputLogic:</i>	Compare (CMP) active level <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Trigger level	Value												
Trigger level	Value														

	Active low	(0x00)
	Active high	(0x01)
<i>PulseWidth:</i>	Pulse width trigger signals (see remarks)	
<i>MoveDirection:</i>	Axis moving direction	
	Direction	Value
	Negative	(0)
	Positive	(1)
	The compare function will only trigger if the axis moves in the specified direction and the compare condition is being met.	
<i>CmpIncEnable:</i>	Select the compare mode:	
	Compare Mode	Value
	One time compare mode: Triggers only one output signal	(0x00)
	Auto-increment compare position: Set the compare trigger to continuously trigger a output signal at equidistant position	(0x01)
<i>ConstPitch:</i>	The auto-increment distance; The distance between two compare signal (pulse) (Only valid if “ <i>CmpIncEnable</i> ” is enabled otherwise this parameter will be ignored)	
<i>CmpData:</i>	The first position at which the compare function will trigger an output signal. (If “ <i>CmpIncEnable</i> ” is disabled then one output signal will be triggered at the “ <i>CmpData</i> ” position)	

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- Compare function cannot be used when axis is in Vring counter mode (see “*etm96_set_vring_counter()*”)
- Pulse width table:

Pulse Width	Minimum trigger period *	Value
160ns~320ns	640ns	TRIG_PULSE_WIDTH_160nsTo320ns (0x01)
320ns~640ns	1.28us	TRIG_PULSE_WIDTH_320nsTo640ns (0x02)
640ns~1.28us	2.56us	TRIG_PULSE_WIDTH_640nsTo1p28us (0x03)
1.28us~2.56us	5.12us	TRIG_PULSE_WIDTH_1p28usTo2p56us (0x04)
2.56us~5.12us	10.24us	TRIG_PULSE_WIDTH_2p56usTo5p12us (0x05)
5.12us~10.24us	20.48us	TRIG_PULSE_WIDTH_5p12usTo10p24us (0x06)
10.24us~20.48us	40.96us	TRIG_PULSE_WIDTH_10p24usTo20p48us

		(0x07)
20.48us~40.96us	81.92us	TRIG_PULSE_WIDTH_20p48usTo40p96us (0x08)
40.96us~81.92us	163.84us	TRIG_PULSE_WIDTH_40p96usTo81p92us (0x09)
81.92us~163.84us	327.68us	TRIG_PULSE_WIDTH_81p92usTo163p84us (0x0A)
163.84us~327.68us	655.36us	TRIG_PULSE_WIDTH_163p84usTo327p68us (0x0B)
327.68us~655.36us	1.31072ms	TRIG_PULSE_WIDTH_327p68usTo655p36us (0x0C)
655.36us~1.31072ms	2.62144ms	TRIG_PULSE_WIDTH_655p36usTo1p31072ms (0x0D)
1.31072ms~2.62144ms	5.24288ms	TRIG_PULSE_WIDTH_1p31072msTo2p62144ms (0x0E)
2.62144ms~5.24288ms	10.48576ms	TRIG_PULSE_WIDTH_2p62144msTo5p24288ms (0x0F)

* If the trigger signal output period is less than the minimum trigger cycles then sporadically no output signal will be generated.

Table 3: Pulse width setting of the trigger signal

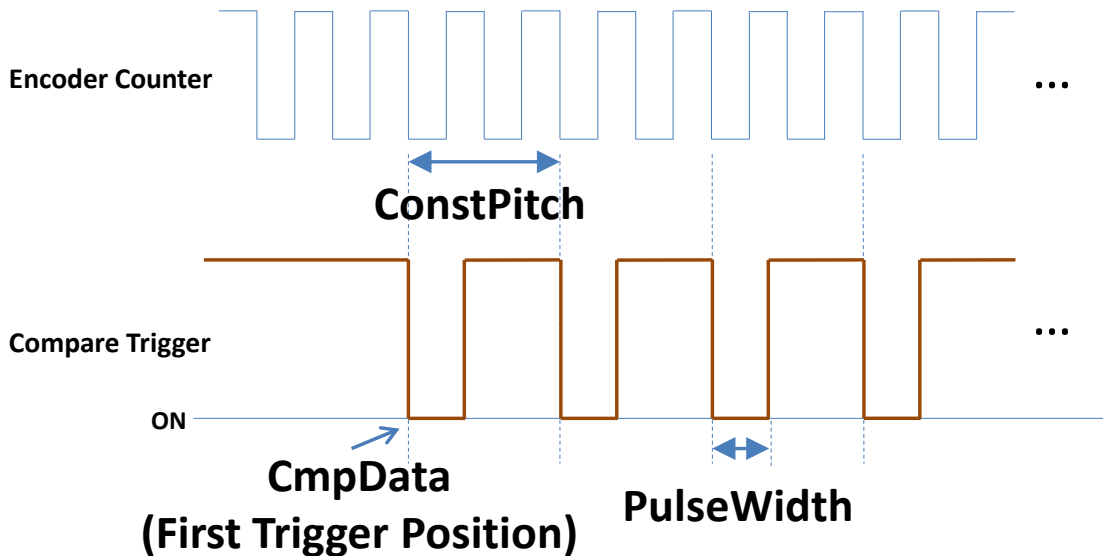


Figure 17: Continuous equidistant spaced trigger output function

6.2 Latch Function

6.2.1 etm96_set_latch_cfg

Configures and enables position Latch. The latch function captures the encoder counter value at an instant when the latch signal activates. The LTC channel is used to receive the latch pulse. The latch function is hardware implemented and executes at very high speed.

Syntax:

```

116 etm96_set_latch_cfg ( HANDLE hEtm,
                          U8 CardNo,
                          U8 AxisNo,
                          U8 EnableDisable,
                          U8 LatchLogic);
    
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"> <thead> <tr> <th>Axis</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Axis0</td> <td>AXIS_0 (0x01)</td> </tr> <tr> <td>Axis1</td> <td>AXIS_1 (0x02)</td> </tr> <tr> <td>Axis2</td> <td>AXIS_2 (0x04)</td> </tr> <tr> <td>Axis3</td> <td>AXIS_3 (0x08)</td> </tr> <tr> <td>Axis4</td> <td>AXIS_4 (0x10)</td> </tr> <tr> <td>Axis5</td> <td>AXIS_5 (0x20)</td> </tr> </tbody> </table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>EnableDisable:</i>	Enable/Disable the latch function <table border="1"> <thead> <tr> <th>State</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>treat LTC PIN as a general input</td> <td>(0x00)</td> </tr> <tr> <td>treat LTC PIN as a dedicated external trigger to latch input</td> <td>(0x01)</td> </tr> </tbody> </table>	State	Value	treat LTC PIN as a general input	(0x00)	treat LTC PIN as a dedicated external trigger to latch input	(0x01)								
State	Value														
treat LTC PIN as a general input	(0x00)														
treat LTC PIN as a dedicated external trigger to latch input	(0x01)														
<i>LatchLogic:</i>	Latch (LTC) active level <table border="1"> <thead> <tr> <th>Trigger level</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Active low</td> <td>(0x00)</td> </tr> <tr> <td>Active high</td> <td>(0x01)</td> </tr> </tbody> </table>	Trigger level	Value	Active low	(0x00)	Active high	(0x01)								
Trigger level	Value														
Active low	(0x00)														
Active high	(0x01)														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

- Only when Latch DI activates will the current encoder position be latched.

6.2.2 etm96_get_latch

Reads the present latched position of the specified axis. Returns the captured position triggered by the latch LTC signal.

Syntax:

```
l16 etm96_get_latch ( HANDLE hEtm,  
                    U8 CardNo,  
                    U8 AxisNo,  
                    l32* LatchData);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>AxisNo:</i>	Axis definition: <table border="1"><thead><tr><th>Axis</th><th>Value</th></tr></thead><tbody><tr><td>Axis0</td><td>AXIS_0 (0x01)</td></tr><tr><td>Axis1</td><td>AXIS_1 (0x02)</td></tr><tr><td>Axis2</td><td>AXIS_2 (0x04)</td></tr><tr><td>Axis3</td><td>AXIS_3 (0x08)</td></tr><tr><td>Axis4</td><td>AXIS_4 (0x10)</td></tr><tr><td>Axis5</td><td>AXIS_5 (0x20)</td></tr></tbody></table>	Axis	Value	Axis0	AXIS_0 (0x01)	Axis1	AXIS_1 (0x02)	Axis2	AXIS_2 (0x04)	Axis3	AXIS_3 (0x08)	Axis4	AXIS_4 (0x10)	Axis5	AXIS_5 (0x20)
Axis	Value														
Axis0	AXIS_0 (0x01)														
Axis1	AXIS_1 (0x02)														
Axis2	AXIS_2 (0x04)														
Axis3	AXIS_3 (0x08)														
Axis4	AXIS_4 (0x10)														
Axis5	AXIS_5 (0x20)														
<i>LatchData:</i>	Pointer to the value of the encoder position counter latch														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

Only when Latch DI activates will the current encoder position be latched.

7 General Purpose IO and Remote FRnet IO Functions

7.1 General Purpose IO

7.1.1 `etm96_set_general_do`

Sets the output state of one general purpose digital output channel.

Syntax:

```
116 etm96_set_general_do ( HANDLE hEtm,  
                           U8 CardNo,  
                           U8 Channel,  
                           U8 OnOff);
```

Parameters:

Name	Description								
<i>hEtm:</i>	TCP/IP handle								
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)								
<i>Channel:</i>	DO channel number: <table border="1"><thead><tr><th>Channel</th><th>Value</th></tr></thead><tbody><tr><td>DO Channel 0</td><td>(0)</td></tr><tr><td>DO Channel 1</td><td>(1)</td></tr><tr><td>DO Channel 2</td><td>(2)</td></tr></tbody></table>	Channel	Value	DO Channel 0	(0)	DO Channel 1	(1)	DO Channel 2	(2)
Channel	Value								
DO Channel 0	(0)								
DO Channel 1	(1)								
DO Channel 2	(2)								
<i>OnOff:</i>	DO State: <table border="1"><thead><tr><th>State</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	State	Value	OFF	OFF (0x00)	ON	ON (0x01)		
State	Value								
OFF	OFF (0x00)								
ON	ON (0x01)								

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.1.2 etm96_get_general_do

Reads the current output state of the specified general purpose DO channel.

Syntax:

```
int etm96_get_general_do ( HANDLE hEtm,  
                          U8 CardNo,  
                          U8 Channel,  
                          U8 *OnOff);
```

Parameters:

Name	Description								
<i>hEtm:</i>	TCP/IP handle								
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)								
<i>Channel:</i>	DO channel number: <table border="1"><thead><tr><th>Channel</th><th>Value</th></tr></thead><tbody><tr><td>DO Channel 0</td><td>(0)</td></tr><tr><td>DO Channel 1</td><td>(1)</td></tr><tr><td>DO Channel 2</td><td>(2)</td></tr></tbody></table>	Channel	Value	DO Channel 0	(0)	DO Channel 1	(1)	DO Channel 2	(2)
Channel	Value								
DO Channel 0	(0)								
DO Channel 1	(1)								
DO Channel 2	(2)								
<i>OnOff:</i>	Pointer to the current DO output state: <table border="1"><thead><tr><th>State</th><th>Value</th></tr></thead><tbody><tr><td>OFF</td><td>OFF (0x00)</td></tr><tr><td>ON</td><td>ON (0x01)</td></tr></tbody></table>	State	Value	OFF	OFF (0x00)	ON	ON (0x01)		
State	Value								
OFF	OFF (0x00)								
ON	ON (0x01)								

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.1.3 etm96_set_all_general_do

Sets all the output signal state of all three general-purpose output channels.

Syntax:

```
I16 etm96_set_all_general_do ( HANDLE hEtm,  
                               U8 CardNo,  
                               U16 MultiGdoStatus);
```

Parameters:

Name	Description										
<i>hEtm:</i>	TCP/IP handle										
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)										
<i>MultiGdoStatus:</i>	Each bit represents a DO channel number. <table border="1"><thead><tr><th>Bit</th><th>Channel No</th></tr></thead><tbody><tr><td>Bit 0</td><td>DO Channel 0</td></tr><tr><td>Bit 1</td><td>DO Channel 1</td></tr><tr><td>Bit 2</td><td>DO Channel 2</td></tr><tr><td>Bit 3~15</td><td>reserved</td></tr></tbody></table> <p>If the bit is 0: indicates that the corresponding signal is OFF If the bit is 1: indicates that the corresponding signal is ON</p>	Bit	Channel No	Bit 0	DO Channel 0	Bit 1	DO Channel 1	Bit 2	DO Channel 2	Bit 3~15	reserved
Bit	Channel No										
Bit 0	DO Channel 0										
Bit 1	DO Channel 1										
Bit 2	DO Channel 2										
Bit 3~15	reserved										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.1.4 etm96_get_all_general_do

Reads the current output state of all three general purpose DO channels.

Syntax:

```
I16 etm96_get_all_general_do ( HANDLE hEtm,  
                               U8 CardNo,  
                               U16* MultiGdoStatus);
```

Parameters:

Name	Description										
<i>hEtm:</i>	TCP/IP handle										
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)										
<i>MultiGdoStatus:</i>	Pointer to 16 bit variable. Each bit represents a DO channel number. <table border="1"><thead><tr><th>Bit</th><th>Channel No</th></tr></thead><tbody><tr><td>Bit 0</td><td>DO Channel 0</td></tr><tr><td>Bit 1</td><td>DO Channel 1</td></tr><tr><td>Bit 2</td><td>DO Channel 2</td></tr><tr><td>Bit 3~15</td><td>reserved</td></tr></tbody></table> If the bit is 0: indicates that the corresponding signal is OFF If the bit is 1: indicates that the corresponding signal is ON	Bit	Channel No	Bit 0	DO Channel 0	Bit 1	DO Channel 1	Bit 2	DO Channel 2	Bit 3~15	reserved
Bit	Channel No										
Bit 0	DO Channel 0										
Bit 1	DO Channel 1										
Bit 2	DO Channel 2										
Bit 3~15	reserved										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.1.5 etm96_get_general_di

Gets the input state of the specified general purpose DI channel.

Syntax:

```

I16 etm96_get_general_di ( HANDLE hEtm,
                          U8 CardNo,
                          U8 Channel,
                          U8 *OnOff);
    
```

Parameters:

Name	Description																										
<i>hEtm:</i>	TCP/IP handle																										
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																										
<i>Channel:</i>	DI channel number: <table border="1" data-bbox="539 808 1107 1266"> <thead> <tr> <th>Channel</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>DI Channel 0</td><td>(0)</td></tr> <tr><td>DI Channel 1</td><td>(1)</td></tr> <tr><td>DI Channel 2</td><td>(2)</td></tr> <tr><td>DI Channel 3</td><td>(3)</td></tr> <tr><td>DI Channel 4</td><td>(4)</td></tr> <tr><td>DI Channel 5</td><td>(5)</td></tr> <tr><td>DI Channel 6</td><td>(6)</td></tr> <tr><td>DI Channel 7</td><td>(7)</td></tr> <tr><td>DI Channel 8</td><td>(8)</td></tr> <tr><td>DI Channel 9</td><td>(9)</td></tr> <tr><td>DI Channel 10</td><td>(10)</td></tr> <tr><td>DI Channel 11</td><td>(11)</td></tr> </tbody> </table>	Channel	Value	DI Channel 0	(0)	DI Channel 1	(1)	DI Channel 2	(2)	DI Channel 3	(3)	DI Channel 4	(4)	DI Channel 5	(5)	DI Channel 6	(6)	DI Channel 7	(7)	DI Channel 8	(8)	DI Channel 9	(9)	DI Channel 10	(10)	DI Channel 11	(11)
Channel	Value																										
DI Channel 0	(0)																										
DI Channel 1	(1)																										
DI Channel 2	(2)																										
DI Channel 3	(3)																										
DI Channel 4	(4)																										
DI Channel 5	(5)																										
DI Channel 6	(6)																										
DI Channel 7	(7)																										
DI Channel 8	(8)																										
DI Channel 9	(9)																										
DI Channel 10	(10)																										
DI Channel 11	(11)																										
<i>OnOff:</i>	Pointer to a DI state: <table border="1" data-bbox="539 1308 1144 1415"> <thead> <tr> <th>State</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>OFF</td> <td>OFF (0x00)</td> </tr> <tr> <td>ON</td> <td>ON (0x01)</td> </tr> </tbody> </table>	State	Value	OFF	OFF (0x00)	ON	ON (0x01)																				
State	Value																										
OFF	OFF (0x00)																										
ON	ON (0x01)																										

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.1.6 etm96_get_all_general_di

Gets the input state of all general purpose DIs.

Syntax:

```
I16 etm96_get_all_general_di ( HANDLE hEtm,  
                               U8 CardNo,  
                               U16* MultiGdiStatus);
```

Parameters:

Name	Description														
<i>hEtm:</i>	TCP/IP handle														
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)														
<i>MultiGdiStatus:</i>	Pointer to 16 bit variable. Each bit represents a DI channel. <table border="1"><thead><tr><th>Bit</th><th>Channel No</th></tr></thead><tbody><tr><td>Bit 0</td><td>DI Channel 0</td></tr><tr><td>Bit 1</td><td>DI Channel 1</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Bit 10</td><td>DI Channel 10</td></tr><tr><td>Bit 11</td><td>DI Channel 11</td></tr><tr><td>Bit 12~15</td><td>reserved</td></tr></tbody></table> <p>If the bit is 0: indicates that the corresponding signal is OFF If the bit is 1: indicates that the corresponding signal is ON</p>	Bit	Channel No	Bit 0	DI Channel 0	Bit 1	DI Channel 1	Bit 10	DI Channel 10	Bit 11	DI Channel 11	Bit 12~15	reserved
Bit	Channel No														
Bit 0	DI Channel 0														
Bit 1	DI Channel 1														
...	...														
Bit 10	DI Channel 10														
Bit 11	DI Channel 11														
Bit 12~15	reserved														

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.2 Remote FRnet IO

7.2.1 etm96_set_frnet_group_do

Sets the output state of a remote FRnet DO module. Each FRnet DO module has 16 DO channels.

Syntax:

```
U16 etm96_set_frnet_group_do ( HANDLE hEtm,  
                                U8 CardNo,  
                                U16 GroupNo,  
                                U16 DOStatus);
```

Parameters:

Name	Description																		
<i>hEtm</i> :	TCP/IP handle																		
<i>CardNo</i> :	1 (The current ET-M8196F supports only one I-8196F module)																		
<i>GroupNo</i> :	Address of the remote FRnet DO module: <table border="1"><thead><tr><th>Address</th><th>Value</th></tr></thead><tbody><tr><td>DO Group 0</td><td>(0)</td></tr><tr><td>DO Group 1</td><td>(1)</td></tr><tr><td>DO Group 2</td><td>(2)</td></tr><tr><td>DO Group 3</td><td>(3)</td></tr><tr><td>DO Group 4</td><td>(4)</td></tr><tr><td>DO Group 5</td><td>(5)</td></tr><tr><td>DO Group 6</td><td>(6)</td></tr><tr><td>DO Group 7</td><td>(7)</td></tr></tbody></table>	Address	Value	DO Group 0	(0)	DO Group 1	(1)	DO Group 2	(2)	DO Group 3	(3)	DO Group 4	(4)	DO Group 5	(5)	DO Group 6	(6)	DO Group 7	(7)
Address	Value																		
DO Group 0	(0)																		
DO Group 1	(1)																		
DO Group 2	(2)																		
DO Group 3	(3)																		
DO Group 4	(4)																		
DO Group 5	(5)																		
DO Group 6	(6)																		
DO Group 7	(7)																		
<i>DOStatus</i> :	A 16 bit parameter where each bit represents the output state of a DO channel. <table border="1"><thead><tr><th>Bit</th><th>Channel Number</th></tr></thead><tbody><tr><td>Bit 0</td><td>CH_0</td></tr><tr><td>Bit 1</td><td>CH_1</td></tr><tr><td>...</td><td>...</td></tr><tr><td>Bit 14</td><td>CH_14</td></tr><tr><td>Bit 15</td><td>CH_15</td></tr></tbody></table> <p>If the bit is 0: indicates that the corresponding signal is OFF If the bit is 1: indicates that the corresponding signal is ON</p>	Bit	Channel Number	Bit 0	CH_0	Bit 1	CH_1	Bit 14	CH_14	Bit 15	CH_15						
Bit	Channel Number																		
Bit 0	CH_0																		
Bit 1	CH_1																		
...	...																		
Bit 14	CH_14																		
Bit 15	CH_15																		

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.2.2 etm96_get_frnet_group_do

Gets the output state of a remote FRnet DO module. Each FRnet DO module has 16 DO channels.

Syntax:

```
U16 etm96_get_frnet_group_do ( HANDLE hEtm,
                                U8 CardNo,
                                U16 GroupNo,
                                U16 * DOStatus);
```

Parameters:

Name	Description																		
<i>hEtm:</i>	TCP/IP handle																		
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																		
<i>GroupNo:</i>	Address of the remote FRnet DO module: <table border="1" data-bbox="539 840 1133 1161"> <thead> <tr> <th>Address</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>DO Group 0</td><td>(0)</td></tr> <tr><td>DO Group 1</td><td>(1)</td></tr> <tr><td>DO Group 2</td><td>(2)</td></tr> <tr><td>DO Group 3</td><td>(3)</td></tr> <tr><td>DO Group 4</td><td>(4)</td></tr> <tr><td>DO Group 5</td><td>(5)</td></tr> <tr><td>DO Group 6</td><td>(6)</td></tr> <tr><td>DO Group 7</td><td>(7)</td></tr> </tbody> </table>	Address	Value	DO Group 0	(0)	DO Group 1	(1)	DO Group 2	(2)	DO Group 3	(3)	DO Group 4	(4)	DO Group 5	(5)	DO Group 6	(6)	DO Group 7	(7)
Address	Value																		
DO Group 0	(0)																		
DO Group 1	(1)																		
DO Group 2	(2)																		
DO Group 3	(3)																		
DO Group 4	(4)																		
DO Group 5	(5)																		
DO Group 6	(6)																		
DO Group 7	(7)																		
<i>DOStatus:</i>	A pointer to a 16 bit parameter. Each bit represents the output state of a DO channel. <table border="1" data-bbox="539 1241 1003 1457"> <thead> <tr> <th>Bit</th> <th>Channel Number</th> </tr> </thead> <tbody> <tr><td>Bit 0</td><td>CH_0</td></tr> <tr><td>Bit 1</td><td>CH_1</td></tr> <tr><td>...</td><td>...</td></tr> <tr><td>Bit 14</td><td>CH_14</td></tr> <tr><td>Bit 15</td><td>CH_15</td></tr> </tbody> </table> If the bit is 0: indicates that the corresponding signal is OFF If the bit is 1: indicates that the corresponding signal is ON	Bit	Channel Number	Bit 0	CH_0	Bit 1	CH_1	Bit 14	CH_14	Bit 15	CH_15						
Bit	Channel Number																		
Bit 0	CH_0																		
Bit 1	CH_1																		
...	...																		
Bit 14	CH_14																		
Bit 15	CH_15																		

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.2.3 etm96_get_frnet_group_do_all

Gets the output state of a remote FRnet DO module. Each FRnet DO module has 16 DO channels.

Syntax:

```
U16 etm96_get_frnet_group_do_all ( HANDLE hEtm,  
U8 CardNo,  
U16 *ArrDOStatus);
```

Parameters:

Name	Description																		
<i>hEtm:</i>	TCP/IP handle																		
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																		
<i>ArrDOStatus:</i>	Pointer to an array with 8 elements: U16 ArrDOStatus[8] ; Each array element contains the status of all 16 DO channels of one FRnet module. <table border="1" data-bbox="537 909 1164 1291"><thead><tr><th>Array Index</th><th>Status</th></tr></thead><tbody><tr><td>[0]</td><td>Group 0 CH_0 ~ CH_15</td></tr><tr><td>[1]</td><td>Group 1 CH_0 ~ CH_15</td></tr><tr><td>[2]</td><td>Group 2 CH_0 ~ CH_15</td></tr><tr><td>[3]</td><td>Group 3 CH_0 ~ CH_15</td></tr><tr><td>[4]</td><td>Group 4 CH_0 ~ CH_15</td></tr><tr><td>[5]</td><td>Group 5 CH_0 ~ CH_15</td></tr><tr><td>[6]</td><td>Group 6 CH_0 ~ CH_15</td></tr><tr><td>[7]</td><td>Group 7 CH_0 ~ CH_15</td></tr></tbody></table>	Array Index	Status	[0]	Group 0 CH_0 ~ CH_15	[1]	Group 1 CH_0 ~ CH_15	[2]	Group 2 CH_0 ~ CH_15	[3]	Group 3 CH_0 ~ CH_15	[4]	Group 4 CH_0 ~ CH_15	[5]	Group 5 CH_0 ~ CH_15	[6]	Group 6 CH_0 ~ CH_15	[7]	Group 7 CH_0 ~ CH_15
Array Index	Status																		
[0]	Group 0 CH_0 ~ CH_15																		
[1]	Group 1 CH_0 ~ CH_15																		
[2]	Group 2 CH_0 ~ CH_15																		
[3]	Group 3 CH_0 ~ CH_15																		
[4]	Group 4 CH_0 ~ CH_15																		
[5]	Group 5 CH_0 ~ CH_15																		
[6]	Group 6 CH_0 ~ CH_15																		
[7]	Group 7 CH_0 ~ CH_15																		

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.2.4 etm96_scan_frnet_di

Scans the FRnet network for remote FRnet DI modules and returns which DI modules are connected and active. A maximum number of 8 FRnet DI modules can be used in a FRnet network.

Syntax:

```
I16 etm96_scan_frnet_di ( HANDLE hEtm,  
                          U8 CardNo,  
                          U16 *DIModules);
```

Parameters:

Name	Description																				
<i>hEtm:</i>	TCP/IP handle																				
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																				
<i>DIModules:</i>	Pointer to a16 bit variable. Each bit represents an FRnet DI module. A set bit indicates that the remote FRnet DI module is connected and active. <table border="1"><thead><tr><th>Bit</th><th>DI Module Address</th></tr></thead><tbody><tr><td>Bit 0</td><td>Group 8</td></tr><tr><td>Bit 1</td><td>Group 9</td></tr><tr><td>Bit 2</td><td>Group 10</td></tr><tr><td>Bit 3</td><td>Group 11</td></tr><tr><td>Bit 4</td><td>Group 12</td></tr><tr><td>Bit 5</td><td>Group 13</td></tr><tr><td>Bit 6</td><td>Group 14</td></tr><tr><td>Bit 7</td><td>Group 15</td></tr><tr><td>Bit 8 ~ 15</td><td>reserved</td></tr></tbody></table>	Bit	DI Module Address	Bit 0	Group 8	Bit 1	Group 9	Bit 2	Group 10	Bit 3	Group 11	Bit 4	Group 12	Bit 5	Group 13	Bit 6	Group 14	Bit 7	Group 15	Bit 8 ~ 15	reserved
Bit	DI Module Address																				
Bit 0	Group 8																				
Bit 1	Group 9																				
Bit 2	Group 10																				
Bit 3	Group 11																				
Bit 4	Group 12																				
Bit 5	Group 13																				
Bit 6	Group 14																				
Bit 7	Group 15																				
Bit 8 ~ 15	reserved																				

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

7.2.5 etm96_get_frnet_group_di

Reads 16 channel states of a remote FRnet DI module. Each FRnet DI module has 16 DI channels.

Syntax:

```
U16 etm96_get_frnet_group_di ( HANDLE hEtm,
                               U8 CardNo,
                               U16 GroupNo,
                               U16 * DIStatus);
```

Parameters:

Name	Description																		
<i>hEtm:</i>	TCP/IP handle																		
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																		
<i>GroupNo:</i>	Address of the remote FRnet DI module: <table border="1" data-bbox="539 840 1325 1161"> <thead> <tr> <th>Address</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>DI Group 8</td> <td>(8)</td> </tr> <tr> <td>DI Group 9</td> <td>(9)</td> </tr> <tr> <td>DI Group 10</td> <td>(10)</td> </tr> <tr> <td>DI Group 11</td> <td>(11)</td> </tr> <tr> <td>DI Group 12</td> <td>(12)</td> </tr> <tr> <td>DI Group 13</td> <td>(13)</td> </tr> <tr> <td>DI Group 14</td> <td>(14)</td> </tr> <tr> <td>DI Group 15</td> <td>(15)</td> </tr> </tbody> </table>	Address	Value	DI Group 8	(8)	DI Group 9	(9)	DI Group 10	(10)	DI Group 11	(11)	DI Group 12	(12)	DI Group 13	(13)	DI Group 14	(14)	DI Group 15	(15)
Address	Value																		
DI Group 8	(8)																		
DI Group 9	(9)																		
DI Group 10	(10)																		
DI Group 11	(11)																		
DI Group 12	(12)																		
DI Group 13	(13)																		
DI Group 14	(14)																		
DI Group 15	(15)																		
<i>DIStatus:</i>	A pointer to a 16 bit parameter. Each bit represents the input state of a DI channel. <table border="1" data-bbox="539 1312 1002 1526"> <thead> <tr> <th>Bit</th> <th>Channel Number</th> </tr> </thead> <tbody> <tr> <td>Bit 0</td> <td>CH_0</td> </tr> <tr> <td>Bit 1</td> <td>CH_1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>Bit 14</td> <td>CH_14</td> </tr> <tr> <td>Bit 15</td> <td>CH_15</td> </tr> </tbody> </table> If the bit is 0: indicates that the corresponding signal is OFF If the bit is 1: indicates that the corresponding signal is ON	Bit	Channel Number	Bit 0	CH_0	Bit 1	CH_1	Bit 14	CH_14	Bit 15	CH_15						
Bit	Channel Number																		
Bit 0	CH_0																		
Bit 1	CH_1																		
...	...																		
Bit 14	CH_14																		
Bit 15	CH_15																		

Return:

- 0: Success
- Others: Error (refer to error documentation)

7.2.6 etm96_get_frnet_group_di_all

Reads 16 channel states of a remote FRnet DI module. Each FRnet DI module has 16 DI channels.

Syntax:

```
U16 etm96_get_frnet_group_di_all ( HANDLE hEtm,  
U8 CardNo,  
U16 * ArrDIStatus);
```

Parameters:

Name	Description																		
<i>hEtm:</i>	TCP/IP handle																		
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)																		
<i>ArrDIStatus:</i>	Pointer to an array with 8 elements: U16 ArrDiStatus[8] ; Each array element contains the status of all 16 DI channels of one FRnet module. <table border="1" data-bbox="539 911 1289 1291"><thead><tr><th>Array Index</th><th>Status</th></tr></thead><tbody><tr><td>[0]</td><td>Group 8 CH_0 ~ CH_15</td></tr><tr><td>[1]</td><td>Group 9 CH_0 ~ CH_15</td></tr><tr><td>[2]</td><td>Group 10 CH_0 ~ CH_15</td></tr><tr><td>[3]</td><td>Group 11 CH_0 ~ CH_15</td></tr><tr><td>[4]</td><td>Group 12 CH_0 ~ CH_15</td></tr><tr><td>[5]</td><td>Group 13 CH_0 ~ CH_15</td></tr><tr><td>[6]</td><td>Group 14 CH_0 ~ CH_15</td></tr><tr><td>[7]</td><td>Group 15 CH_0 ~ CH_15</td></tr></tbody></table>	Array Index	Status	[0]	Group 8 CH_0 ~ CH_15	[1]	Group 9 CH_0 ~ CH_15	[2]	Group 10 CH_0 ~ CH_15	[3]	Group 11 CH_0 ~ CH_15	[4]	Group 12 CH_0 ~ CH_15	[5]	Group 13 CH_0 ~ CH_15	[6]	Group 14 CH_0 ~ CH_15	[7]	Group 15 CH_0 ~ CH_15
Array Index	Status																		
[0]	Group 8 CH_0 ~ CH_15																		
[1]	Group 9 CH_0 ~ CH_15																		
[2]	Group 10 CH_0 ~ CH_15																		
[3]	Group 11 CH_0 ~ CH_15																		
[4]	Group 12 CH_0 ~ CH_15																		
[5]	Group 13 CH_0 ~ CH_15																		
[6]	Group 14 CH_0 ~ CH_15																		
[7]	Group 15 CH_0 ~ CH_15																		

Return:

- 0: Success
- Others: Error (refer to error documentation)

8 Hardware and Software Version Functions

8.1 ET-M8196F Version

8.1.1 `etm96_get_firmware_version`

Gets the firmware version number of the ET-M8196F.

Syntax:

```
U16 etm96_get_firmware_version ( HANDLE hEtm,  
                                U8 CardNo,  
                                U16* Ver);
```

Parameters:

Name	Description
<i>hEtm:</i>	TCP/IP handle
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)
<i>Ver:</i>	Firmware version Format: 0xXXYY XX: Major version number YY: Minor version number Example: 0x0100 indicates the version number is 1.0

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

8.2 I-8196F Version

8.2.1 etm96_get_i8196f_version

Read the firmware and hardware version of the I-8196F.

Syntax:

```
I16 etm96_get_i8196f_version ( HANDLE hEtm,  
                               U8 CardNo,  
                               U16* FWVer,  
                               U16* DSPVer,  
                               U16* PCBVer,  
                               U16* PLDVer,  
                               U16* FPGAVer );
```

Parameters:

Name	Description
<i>hEtm:</i>	TCP/IP handle
<i>CardNo:</i>	1 (The current ET-M8196F supports only one I-8196F module)
<i>FWVer:</i>	Pointer to firmware version
<i>DSPVer:</i>	Pointer to DSP firmware version
<i>PCBVer:</i>	Pointer to PCB version
<i>PLDVer:</i>	Pointer to PLD version
<i>FPGAVer:</i>	Pointer to FPGA version

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

8.3 DLL Version

8.3.1 etm96_get_dll_pathW

Gets the file path location of the loaded "ET-M8196F.dll" in Unicode format.

Syntax:

```
I16 etm96_get_dll_pathW (  wchar_t* pFilePath,  
                          U32 nSize      );
```

Parameters:

Name	Description
<i>pFilePath</i> :	File path location
<i>nSize</i> :	The data length of the "pFilePath"

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

[VC++]

```
I16 ret = 0;  
wchar_t pFilePath[MAX_PATH];  
U32 nSize = MAX_PATH;  
  
ret = etm96_get_dll_pathW( pFilePath, nSize);  
if(ret == 0) // ETM96_ERR_NO_ERROR  
{ MessageBox(pFilePath);}  
else  
{ MessageBox(L"Get DLL Path Error !!!");}
```

8.3.2 etm96_get_dll_pathA

Gets the file path location of the loaded "ET-M8196F.dll" in ANSI format.

Syntax:

```
I16 etm96_get_dll_pathA ( char* pFilePath,  
                        U32 nSize );
```

Parameters:

Name	Description
<i>pFilePath</i> :	File path location
<i>nSize</i> :	The data length of the "pFilePath"

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

[VC++]

```
I16 ret = 0;  
char pFilePath[MAX_PATH];  
U32 nSize = MAX_PATH;  
  
ret = etm96_get_dll_pathA( pFilePath, nSize);  
  
if(ret == 0) // ETM96_ERR_NO_ERROR  
{ MessageBox(pFilePath);}  
else  
{ MessageBox("Get DLL Path Error !!!");}
```

8.3.3 etm96_get_dll_version

Gets the DLL version.

Syntax:

```
I16 etm96_get_dll_version ( U16* Ver1,  
                           U16* Ver2,  
                           U16* Ver3,  
                           U16* Ver4);
```

Parameters:

Name	Description
Ver1:	<ul style="list-style-type: none">• 0032: DLL for the 32-bit version• 0064: DLL for the 64-bit version
Ver2:	Indicates the year of the DLL release date 00YY: the last two digits indicate the year of release Example: 0015 indicates that the DLL was released in 2015
Ver3:	MMDD: the month and day of the release date Example: 1021 indicates that the DLL was released on October the 21.
Ver4:	Function version number; General version is 0

Return:

- 0: Success
- Others: Error (refer to error documentation)

Remarks:

[VC++]

```
I16 ret = 0;  
U16 Ver[4] = {0};  
CString str;  
  
ret = etm96_get_dll_version( &Ver[0], &Ver[1], &Ver[2], &Ver[3]);  
  
if (ret == 0) // ETM96_ERR_NO_ERROR  
{str.Format("DLL Version: %d.%d.%d.%d", Ver[0], Ver[1], Ver[2],  
Ver[3]);}  
else  
{str.Format("Get DLL Version Error");}  
MessageBox(str);
```

9 Appendix

9.1 Variable data type definition

Type	Bytes	VC++ / BCB	C#	VB.NET	VB6	Delphi
F64	8	double	double	Double	Double	Double
F32	4	float	float	Single	Single	Single
U32	4	unsigned long	uint	UInteger	Long	LongWord
I32	4	long	int	Integer	Long	LongInt
U16	2	unsigned short	ushort	UShort	Integer	Word
I16	2	short	short	Short	Integer	Smallint
U8	1	unsigned char	byte	Byte	Byte	Byte
		char*	string	String	String	WideString/ WideChar
		wchar_t*	string	String	String	AnsiString/ AnsiChar
		HANDLE	IntPtr	IntPtr	Long	Thandle

Table 4: Variable data type definition

9.2 Function Error Code

Communication Errors		
Error Code	Name	Description
0	ETM96_ERR_NO_ERROR	No error occurred during the function call
-5001	ETM96_ERR_NOT_CONNECTED	PC is not connected to the ET-M8196F. First call "etm96_connect()" to establish an Ethernet connection to the remote ET-M8196F.
-5002	ETM96_ERR_SOCKET_ERROR	Communication error
-5003	ETM96_ERR_INVALID_HANDLE	Incorrect handle. Use the handle returned by "etm96_connect()" to call the ET-M8196F motion commands.
-5004	ETM96_ERR_INVALID_CARD	Invalid card number. Currently the ET-M8196F only supports one I-8196F module, therefore select 1 for the card parameter.
-5005	ETM96_ERR_INVALID_AXIS	Invalid axis number.
-5006	ETM96_ERR_INVALID_GROUP	Invalid FRnet group number.
-5007	ETM96_ERR_INVALID_CHANNEL	Invalid I/O channel number.
-5008	ETM96_ERR_WRONG_SPEED	Incorrect speed setting. Make sure that the initial and end speed is smaller than the driving speed.
-5009	ETM96_ERR_WRONG_PARAMETER	Incorrect parameter setting.
-5010	ETM96_ERR_NOT_ENOUGH_MEMORY	The amount of continuous interpolation commands stored in one TCP/IP packet exceeds the maximum reserved memory of 1300 bytes.
-5011	ETM96_ERR_NO_DATA	It was attempted to send an empty TCP/IP packet. The TCP/IP packet for storing continuous interpolation commands is empty. Call the "etm96_get_conti_packet_size()" to determine the number of bytes written to the TCP/IP packet.
-5012	ETM96_ERR_GET_DLL_INFO_FAIL	Failed to get the ET_M8196F.dll version. Make sure the file exist.
-5100	ETM96_ERR_MB_UNDEFINED_ERROR	Modbus TCP packet transmission error occurred. Make sure that the 1. Ethernet communication is normal 2. ET-M8196F firmware version supports this function
-5101	ETM96_ERR_MB_ILLEGAL_FUNCTION	
-5102	ETM96_ERR_MB_ILLEGAL_DATA_ADDRESS	
-5103	ETM96_ERR_MB_ILLEGAL_DATA_VALUE	
-5104	ETM96_ERR_MB_SLAVE_DEVICE_FAILURE	
-5105	ETM96_ERR_MB_ACKNOWLEDGE	
-5106	ETM96_ERR_MB_SLAVE_DEVICE_BUSY	
-5111	ETM96_ERR_MB_WRONG_CARD_NO	
-5112	ETM96_ERR_MB_TIME_OUT	
-5113	ETM96_ERR_MB_INCORRECT_TID	
-5114	ETM96_ERR_MB_INCORRECT_FC	
-5120	ETM96_ERR_RESPONSE_WRONG_DATA	
-5121	ETM96_ERR_CMD_NOT_SUPPORTED	

-5122	ETM96_ERR_REGISTER_QTY	
-5123	ETM96_ERR_VALUE_EXCEED_RANGE	
-5200	ETM96_ERR_INTERNAL_ERROR	Internal error. Contact our technical service personnel for assistance.
-5210	ETM96_ERR_SP_NOT_OPEN_PORT	Serial port communication error.
-5211	ETM96_ERR_SP_OPEN_PORT_FAIL	
-5212	ETM96_ERR_SP_FILE_NOT_EXIST	
-5213	ETM96_ERR_SP_DOWNLOAD_FAIL	
-5214	ETM96_ERR_SP_SEND_CMD_FAIL	

I-8196F Errors		
Error Code	Name	Description
1	I8196_ERR_ERROR	Reserved
2	I8196_ERR_CARD_NOT_FOUND	No I-8196F card detected in the slot.
3	I8196_ERR_INVALID_SLOT_NO	Specified slot is not being supported.
4	I8196_ERR_INVALID_PARA	An invalid parameter value has been passed to a function.
5	I8196_ERR_INVALID_DRIVING_SPEED	An invalid value has been passed to the driving speed parameter
6	I8196_ERR_INVALID_START_SPEED	An invalid value has been passed to the start speed parameter
7	I8196_ERR_INVALID_END_SPEED	An invalid value has been passed to the end speed parameter
8	I8196_ERR_INVALID_MPG_SPEED	Reserved
9	I8196_ERR_INVALID_MOVE_DIRECTION	An invalid value has been passed to the “move direction” parameter
10	I8196_ERR_AXES_NOT_MATCH	Internal error
11	I8196_ERR_GROUP_INVALID_DIMENSION	Internal error
12	I8196_ERR_INVALID_GROUP_AXES	Internal error
13	I8196_ERR_GROUP_NOT_CONFIGURED	Internal error
14	I8196_ERR_GROUP_OUT_OF_RANGE	Internal error
15	I8196_ERR_INCORRECT_GROUP_ASSIGNED	Internal error
16	I8196_ERR_GROUP_SAME_AXIS	Internal error
17	I8196_ERR_GROUP_RELEASE_REQUIRED	Internal error
18	I8196_ERR_MOTION_NOT_FINISHED	<ul style="list-style-type: none"> Some parameters cannot be set if a motion command is still executing If the axis is in not in a continuous interpolation mode then the next motion command cannot be sent while the previous motion command is still executing.
19	I8196_ERR_CARD_NOT_REGISTERED	The I-8196F card has not been registered and initialized yet.
20	I8196_ERR_INVALID_AXIS	An invalid value has been passed to the axis parameter
21	I8196_ERR_AXIS_ERROR	Internal error
22	I8196_ERR_CMD_NO_CLOSED_LOOP_SUPPORT	Reserved
23	I8196_ERR_INTERPOL_NOT_CONFIGURED	A continuous interpolation command has been sent while the axes are not in

		continuous interpolation mode. First call "etm96_set_conti_interp_cfg()" to set the axes in continuous interpolation mode.
24	I8196_ERR_INVALID_HELICAL_MODE	reserved
25	I8196_ERR_INVALID_FRNET_SPEED_SETTING	reserved
26	I8196_ERR_INVALID_FRNET_SA_GROUP_ADDR	Invalid address has been passed to the FRnet DI module address parameter.
27	I8196_ERR_INVALID_FRNET_RA_GROUP_ADDR	Invalid value has been passed to the FRnet DO module address parameter.
28	I8196_ERR_INVALID_MPG_GAIN	Reserved
29	I8196_ERR_MPG_NOT_CONFIGURED	Reserved
30	I8196_ERR_MOTION_IS_COMPLETED	Reserved
31	I8196_ERR_CMD_NOT_FOR_CONTI_INTERPOL	The called function cannot be called when the axes are in continuous interpolation mode. First call "etm96_set_conti_interp_cfg()" to disable the continuous interpolation mode.
32	I8196_ERR_MODE_NOT_FOR_CONTI_INTERPOL	Reserved
33	I8196_ERR_DPRAM_RTC_BLOCKS_AVAILABLE	Internal error
34	I8196_ERR_DPRAM_RTC_BUFFER_FULL	Internal error
35	I8196_EMG_ACTIVATED	Emergency stop has been activated. No new motion commands can be executed while emergency stop is active.
36	I8196_ERR_CONFLICT_CONTI_INTERP_CONFIG	A "non-continuous" interpolation command has been sent while the axes are "continuous interpolation" mode. First call "etm96_set_conti_interp_cfg()" to disable the continuous interpolation mode.
37	I8196_ERR_CONTI_INTERP_INVALID_CONFIG	Reserved
38	I8196_ERR_CONTI_INTERP_INTERRUPTED	Reserved
39	I8196_ERR_CONTI_INTERP_INVALID_START	Reserved
40	I8196_ERR_CONTI_INTERP_FIFO_EMPTY	Reserved
41	I8196_ERR_NOT_IN_CONTI_INTERP_MODE	Reserved
42	I8196_ERR_CONTI_INTERP_START	Reserved
43	I8196_ERR_INVALID_RING_COUNTER	The value passed to the ring counter parameter is not being supported.
44	I8196_ERR_ISR_IS_USED_FOR_CMP_TRIG	Compare trigger function is in use, can't use ring counter mode.
45	I8196_ERR_RTC_WAIT_ABORTED	Internal error
46	I8196_ERR_RTC_TIMEOUT	Internal error
47	I8196_ERR_DSP_RESET	Resetting the DSP failed
48	I8196_ERR_INFO_SET_DDA_NOT_REQUIRED	The DDA has already been set
49	I8196_ERR_EXCEED_MAX_POSTION	The value passed to the set position counter parameter exceeds the valid range. Valid range:-2147483646~ 2147483646
50	I8196_ERR_NOT_SUPPORTED_BY_VRING_MODE	The axes are in ring counter mode. Some functions are not supported in ring counter mode (e.g. "etm96_set_compare_trig_cfg()")
-1001	I8196_ERR_DSP_MOTIONPATH_FULL	DSP response error code
-1002	I8196_ERR_DSP_REMAIN_STILL	

-1003	I8196_ERR_DSP_DIMENSION_OUT_RANGE
-1004	I8196_ERR_DSP_SPEED_VALUE
-1005	I8196_ERR_DSP_SMALLSHAPE_ERROR
-1006	I8196_ERR_DSP_SAWSHAPE_ERROR
-1007	I8196_ERR_DSP_AXIS_INUSE
-1008	I8196_ERR_DSP_NORMALVECTOR
-1009	I8196_ERR_DSP_CIRCLE_RADIUS
-1011	I8196_ERR_DSP_MECH_PARA_VALUE
-1100	I8196_ERR_DSP_MOTIONPATH_ALREADY_FREE
-1201	I8196_ERR_DSP_EXCEED_SPEED_LIMIT
-1202	I8196_ERR_DSP_EXCEED_VRING_LIMIT
-1203	I8196_ERR_DSP_EXCEED_MAX_POSTION
-2001	I8196_ERR_DSP_NO_COMMAND
-2002	I8196_ERR_DSP_CRC_ERROR
-2003	I8196_ERR_DSP_UNKNOWN_COMMAND
-2004	I8196_ERR_DSP_MULTIBLOCK_CMD
-2005	I8196_ERR_DSP_ACC_TYPE
-2006	I8196_ERR_DSP_DEC_TYPE
-2007	I8196_ERR_DSP_CMD_NOT_ALLOWED
-2011	I8196_ERR_DSP_INHIBIT_BY_EMG
-2012	I8196_ERR_DSP_INHIBIT_BY_MPG_EMG
-2013	I8196_ERR_DSP_INHIBIT_BY_PEL
-2014	I8196_ERR_DSP_INHIBIT_BY_MEL
-2015	I8196_ERR_DSP_INHIBIT_BY_ALM
-2016	I8196_ERR_DSP_INHIBIT_BY_RDY
-2101	I8196_ERR_DSP_GROUP_OUT_RANGE
-2102	I8196_ERR_DSP_GROUP_ALREADY_FREE
-2103	I8196_ERR_DSP_GROUP_ASSIGNED
-2104	I8196_ERR_DSP_GROUP_INUSE
-2105	I8196_ERR_DSP_GROUP_NOT_INUSE
-2106	I8196_ERR_DSP_AXIS_ASSIGNED
-2201	I8196_ERR_DSP_CONTROL_MODE
-2301	I8196_ERR_DSP_BUFFER_FULL
-2302	I8196_ERR_DSP_BUFFER_INFO
-2303	I8196_ERR_DSP_BUFFER_TYPE
-2304	I8196_ERR_DSP_BUFFER_SIZE
-2305	I8196_ERR_DSP_BUFFER_INUSE
-2306	I8196_ERR_DSP_BUFFER_NOT_READY
-2401	I8196_ERR_DSP_STOP_BY_P_CHANGE
-2402	I8196_ERR_DSP_CLEAR_STOP
-2410	I8196_ERR_DSP_STOP_BY_GINP
-2430	I8196_ERR_DSP_STOP_BY_AXIS_IO
-2501	I8196_ERR_DSP_NO_AVAILABLE_MACRO
-2502	I8196_ERR_DSP_MACRO_INUSE
-2503	I8196_ERR_DSP_MACRO_EMPTY
-2601	I8196_ERR_DSP_HOMING_IN_PROGRESS
-3601	I8196_ERR_DSP_OUTPUT_SATURATION
-3611	I8196_ERR_DSP_ERR_MSG_BUF_EMPTY
-3612	I8196_ERR_DSP_ERR_MSG_BUF_OVERFLOW
-1	I8196_ERR_DSP_TIMEOUT_ERROR

-2	I8196_ERR_DSP_HW_ID_ERROR	
-3	I8196_ERR_DSP_AXIS_OUT_RANGE	
-4	I8196_ERR_DSP_ADDR_OUT_RANGE	
-5	I8196_ERR_DSP_VALUE_OUT_RANGE	
-6	I8196_ERR_DSP_FPGA_DL_FAILED	
-101	I8196_ERR_DSP_DA_AUTO_UPDATE	
-102	I8196_ERR_DSP_DA_BUSY	
-201	I8196_ERR_DSP_CMP_INUSE	
-32767	I8196_ERR_DSP_SYSTEM_ERROR	
-32768	I8196_ERR_DSP_NOT_IMPLEMENT	

Table 5: Function error code