Winford Engineering ETH32 Protocol Reference

Table of Contents

Winford Engineering ETH32 Protocol Reference

Overview

This document provides information on the protocol used to control and communicate with the ETH32 device. In most cases, it is not necessary to read or understand the information in this document since the API provided by Winford Engineering can be used to control the device without any knowledge of the details provided in this document. The API takes care of all communication with the device and provides you, the programmer, with a higher level interface to the ETH32 functionality. However, this information is provided for use in special circumstances such as when platforms or lanuages not supported by the API are used.

Connection

All commands and responses to and from the device flow through a single TCP connection as a stream of bytes. TCP is used rather than UDP since its retransmission and verification schemes provide for reliability and integrity of data.

The ETH32 device never initiates a TCP connection, but rather, it listens on port *7152* for incoming connections. Once an application has established a connection with the device's port 7152, there is no limit on the number of commands that can be sent over that connection. It is recommended that the application remain connected until it is finished controlling the device, even if that involves periods of inactivity.

General Structure

Each and every command/response block to or from the device, without exception, is a fixed length of *five bytes* long (referred to as a *command block* throughout this document). Using a fixed-length block makes command delimiting and buffering simple and efficient in that a lookup table is not necessary to know how long a particular command is - they are all the same length. Hence it is extremely important that commands are padded out to a full five bytes. Even if a command only has two meaningful bytes, it will not be processed by the device until all five bytes are received.

Note that there are no restrictions on the number of command blocks that may arrive in a single TCP/IP packet, other than the limit of available buffer space. In fact, there are no assumptions made regarding relationship between TCP/IP packets and command blocks. For example, a TCP/IP packet may contain only the first two bytes of a command block, but as long as the remaining three arrive eventually, the command will be processed as normal.

All command blocks to or from the device begin with a 1-byte command code identifying the purpose of the command, such as setting a bit on an output port or reading the value of an input port. The meaning of the rest of the data in the command block depends on the particular command, as depicted by the following diagram:

Byte 0	Command Code
Byte 1	Command-specific data
Byte 2	Command-specific data
Byte 3	Command-specific data
Byte 4	Command-specific data

Communications Summary

There are several scenarios for when and why communication occurs between the application and the device. The following list summarizes the communications in both directions:

- No-reply Commands - Commands sent from the application to the ETH32 device, to which no reply is returned. For example, no reply is sent for a command to set a port’s value.
- Queries - Commands sent from the application to the device, which invoke a reply from the device. For example, a command to read the value of a port results in the device sending a reply block with the port value.
- Replies - Data blocks sent from the device to the application in direct response to a query received from the application.
- Notifications - Sent from the device to the application when certain criteria is met. This includes event notifications as well as periodic "heartbeat" packets.

Port Numbers

For the most part, this protocol uses the same port numbers that are used elsewhere in ETH32 documentation. One exception is that the two LED’s are each treated as separate ports (6 and 7) in this protocol, allowing the normal input and output commands to be used. For all input and output commands, port numbers are summarized as follows:

Port	Description
0	8-bit digital port, on Connector A
1	8-bit digital port, on Connector A
2	8-bit digital port, on Connector A
3	8-bit digital / 8-channel analog port, on Connector B
4	1-bit digital port, on Connector B
5	1-bit digital port, on Connector B
6	LED0 light on front panel
7	LED1 light on front panel

The other exception is that for the purposes of events, analog event banks 0 and 1 are indicated by specifying 4 and 5 in the command block. Since the 1-bit ports 4 and 5 do not support events, there is no conflict. All event ports are listed in the description table for the Enable Event Notifications command.

No-reply Commands

Set Port Value

This command writes the given value to the specified port register. If the port is in output mode, this will set the voltage on the port pins. In input mode, this sets the value of the internal pull-up resistors, where a 1-bit in the given value enables the pull-up resistor on the corresponding bit of the specified port. For example, a value of 0x04 would enable the pull-up on bit 2 of the port, and disable the pull-up on bits 0,1,3-7.

Byte	Purpose	Value
0	Command	2
1	Port	Port Number: 0, 1, ...
2	Value	New port value: 0-255
3	Reserved	
4	Reserved	

Back to Table of Contents

Set Port Direction

This command sets the direction, input or output, of the given port by setting the direction register. The direction of each bit of the port can be set individually, meaning that some bits of the port can be inputs at the same time that other bits on the same port are outputs. A 1-bit in the direction register causes the corresponding line to be put into output mode. For example, a value of 0xF0 would put bits 0-3 into input mode and bits 4-7 into output mode.

This command has three modes for modifying the direction register:

- Copy - The given value is simply copied into the direction register, overwriting the previous value.
- Or - The given value is OR'ed with the current register value. This allows bits to be selectively set without modifying other bits.
- And - The given value is AND'ed with the curreng register value. This allows bits to be selectively cleared without modifying other bits.

Byte	Purpose	Value
0	Command	6
1	Port	Port Number: 0, 1, ...
2	Value	Value used to set/modify the direction register
3	Mode	0 - Copy value to direction register 1 - Or value with current direction register value 2 - And value with current direction register value
4	Reserved	

Back to Table of Contents

Set Analog-to-Digital Convertor State

This command enables or disables the Analog to Digital Convertor (ADC) on the ETH32 device. This command must be issued to enable the ADC before any meaningful analog values can be read.

Byte	Purpose	Value
0	Command	8
1	Port	3 (only applies to port 3)
2	State	0 - digital only, ADC disabled 1 - ADC enabled
3	Reserved	
4	Reserved	

Back to Table of Contents

Enable Event Notifications

This command is used to enable reception of specific events on the current connection. When an event fires on the ETH32 device, the notification is only sent out to connections that have explicitely requested notification of that particular event using this command. Events that are enabled by one connection do not affect the event notifications that another connection will receive. This command is used for digital events, analog events, and counter events.

The following events can be enabled with this command:

- Digital Events - The Event Type field may be 0-3, corresponding to digital ports 0-3. The Bitmask field specifies which bit(s) of the port should fire events.
- Analog Events - The Event Type field may be 4 or 5 to specify analog event bank 0 or 1, respectively. The Bitmask field specifies which analog channel(s) within the event bank should fire events. The actual parameters of the event are defined using the Set Analog Event Definition command.
- Counter Rollovers - The Event Type must be 6. The Bitmask field specifies which counters should fire rollover events. Setting bit 0 enables rollover events from Counter 0, while bit 1 enables Counter 1 events. The actual parameters of where the counter rolls over the event are defined using the Set Counter Rollover Threshold command.
- Counter Thresholds - The Event Type must be 7. The Bitmask field specifies which counters should fire threshold events. Setting bit 0 enables events from Counter 0, while bit 1 enables Counter 1 events. The actual threshold of the event is defined using the Set Counter Event Threshold command.

Please note that the given Enable Bitmask is ORed with the currently enabled event notifications and hence this does not disable any notifications that were previously enabled. For example, specifying a value of 0 for the Enable Bitmask would effectively do nothing. The Disable Event Notifications command is used to disable event notifications.

Byte	Purpose	Value
0	Command	10
1	Event Type	0 - Port 0 1 - Port 1 2 - Port 2 3 - Port 3 4 - Analog Event Bank 0 5 - Analog Event Bank 1 6 - Counters Rollovers 7 - Counter Thresholds
2	Enable Bitmask	Any 1-bit enables event notifications for the corresponding port bit
3	Reserved	
4	Reserved	

Back to Table of Contents

Disable Event Notifications

This command turns off event notifications from the specified event. Any bits that are 0 in the given Disable Bitmask will turn off event notifications on the corresponding bit/channel/counter. Note that the given Disable Bitmask is ANDed with the currently enabled event notifications and hence this command cannot be used to enable any event notifications. For example, specifying a Disable Bitmask of 0xFF (255) would effectively do nothing.

Byte	Purpose	Value
0	Command	11
1	Event Type	0 - Port 0 1 - Port 1 2 - Port 2 3 - Port 3 4 - Analog Event Bank 0 5 - Analog Event Bank 1 6 - Counter Rollovers 7 - Counter Thresholds
2	Disable Bitmask	Any 0-bit disables event notifications for the corresponding port bit
3	Reserved	
4	Reserved	

Back to Table of Contents

Set Analog Event Definition

This command defines the thresholds for a single analog channel in the specified analog event bank. The thresholds that are defined determine what analog readings will cause the event to fire. The thresholds allow the event logic to assign a current state (high or low) to the channel. The channel will be considered high if the reading is at or above the given hi-mark and will be considered low if at or below the given lo-mark. Whenever the state of the channel changes (low to high or high to low), an event will be fired. When the analog reading is between the lo-mark and hi-mark, it will retain its previous value. This allows hysteresis to be built into the event so that a fluctuating signal will not cause an event to continuously fire. The thresholds are specified in 8-bit resolution, and thus they will be compared with the eight Most Significant Bits of the analog readings to determine when an event should be fired. The given hi-mark must be greater than the lo-mark.

Normally, the "initial state" (high or low) of the analog event definition is determined by the current level of the analog reading at the time this command is received. However, if the current analog reading is between the lo-mark and hi-mark, an initial state cannot be accurately assigned. To deal with this, a spare bit in the channel field of this command is used to specify a default state when the initial state cannot be determined. In all other situations (when the reading at the time of event definition is <= lo-mark or >= hi-mark) this bit will simply be ignored.

Please note that defining the thresholds with this command does not enable the current connection to actually receive the event notifications when they occur. These must be enabled separately using the Enable Event Notifications command. Also note that the analog event thresholds are common to all connections. Changing the thresholds will affect other connections if they are utilizing that particular event.

Because the ETH32 device has two analog event banks, two events can be defined for each logical analog channel. Applications can utilize both event banks independently to implement a number of different event notification schemes.

Byte	Purpose	Value
0	Command	14
1	Bank and Channel	Bit 7: 0 for low default state, 1 for high default state Bit 3: 0 for Bank 0, 1 for Bank 1 Bits 0-2: Channel number (0-7)
2	Lo-mark	Low threshold
3	Hi-mark	High threshold
4	Reserved	

Back to Table of Contents

Set Port Bits

This command turns on (sets high) the specified bits on the specified port. This is similar to the Set Port Value command except that this command allows for less disruption of the port value if the current value is not known. Any bits that are 1 in the given bitmask will set the corresponding bit in the output register to 1. Bits that are 0 in the given bitmask will have no effect. For example, specifying a bitmask of 0 would effectively do nothing.

Byte	Purpose	Value
0	Command	15
1	Port	Port Number: 0, 1, ...
2	Bitmask	Any 1-bit turns on the corresponding output register bit
3	Reserved	
4	Reserved	

Back to Table of Contents

Clear Port Bits

This command turns off (sets to low) the specified bits high on the specified port. This is similar to the Set Port Value command except that this command allows for less disruption of the port value if the current value is not known. Any bits that are 0 in the given bitmask will set the corresponding bit in the output register to 0. Bits that are 1 in the given bitmask will have no effect. For example, specifying a bitmask of 0xFF (255) would effectively do nothing.

Byte	Purpose	Value
0	Command	16
1	Port	Port Number: 0, 1, ...
2	Bitmask	Any 0-bit turns off the corresponding output register bit
3	Reserved	
4	Reserved	

Back to Table of Contents

Set Analog Voltage Reference

This command instructs the Analog to Digital Convertor to select the specified source as the reference voltage for conversions. The reference voltage determines the voltage level that will give the highest possible analog reading value. There are three possible configurations as specified below.

Byte	Purpose	Value
0	Command	18
1	Reference	0 - External voltage reference (user supplied) 1 - Internal AVCC (5V) 2 - Reserved for future use 3 - Internal 2.56V Reference
2	Reserved	
3	Reserved	
4	Reserved	

Back to Table of Contents

Set Analog Channel Assignment

This command allows the source of each of the eight analog channels to be modified. The default on powerup is for the eight analog channels to be assigned the eight physical pins of Port 3 respectively and to read in single-ended mode (value with respect to ground). While the typical main use of this command will be to access the various differential analog readings that are available, it can also be used to completely reorganize and reassign the mapping from physical pins to analog channel number.

For example, this command could be used to configure Channel 1 to also read pin 0 at the same time that Channel 0 also remains configured to read pin 0. One advantage to such a configuration would be that the extra analog event definitions provided by Channel 1 would allow for more complex analog event monitoring on the value of pin 0.

The value specified to select a channel assignment is a 5-bit value used internally as the the Analog Multiplexer selection register on the Atmel ATMEGA64 microcontroller. Please see the full ETH32 product documentation or the ATMEGA64 datasheet (refer to the MUX 0-4 bits of the ADMUX register) for a reference of possible options.

Byte	Purpose	Value
0	Command	20
1	Channel	Analog Channel to configure (0-7)
2	Source	Multiplexer value used to select the source of the analog reading for this channel
3	Reserved	
4	Reserved	

Back to Table of Contents

Reset Port Configuration

This command resets the default configuration of all ports and their values without actually doing a reset of the entire board. All configurations of port direction, value, analog voltage reference, analog channel assignments, digital events, analog events, counter configuration and events, and PWM channels will be reset to their defaults. The Analog to Digital Convertor will also be disabled after this command. This command does not reset or modify any of the TCP/IP configuration data that is stored in the device, such as IP address. All active TCP/IP connections to the board are maintained and do not need to be reestablished after the command.

Byte	Purpose	Value
0	Command	26
1	Reserved	
2	Reserved	
3	Reserved	
4	Reserved	

Back to Table of Contents

Pulse Bit

This command outputs a burst of pulses on the port and bit specified. The bit must already be configured as output mode for this command to have effect. The Edge parameter specifies the type of pulse that is generated. Depending on this parameter, a single pulse is defined as follows:

- Falling Edge - Bit is set low, then set high
- Rising Edge - Bit is set high, then set low

The falling edge mode would typically be used on a bit that is initially high (and likewise rising edge with low), but this is not required. If a single falling edge pulse is performed on a bit that is already low, the pulse will end up simply setting the bit high. The reverse applies to a rising edge pulse where the bit is already high.

Byte	Purpose	Value
0	Command	28
1	Port	Port Number: 0, 1, ...
2	Bit	Bit Number: 0-7
3	Edge	0 - Falling Edge 1 - Rising Edge
4	Count	Number of pulses to perform

Back to Table of Contents

Set Counter State

This command enables and disables the counters of the ETH32 device and configures which input signal edge (rising or falling) will increment the counter value. This command does not affect the current counter value in any way. In other words, a counter that is disabled and then enabled again will retain its value. The power-on value of all counters is zero.

Byte	Purpose	Value
0	Command	30
1	Counter	Counter to configure (0 or 1)
2	State	0 - Counter disabled 1 - Increment on falling edge 2 - Increment on rising edge
3	Reserved	
4	Reserved	

Back to Table of Contents

Write Counter Value

This command allows the counter to be preloaded with the given value. This may be useful in initial setup of the counter. Note that if you have a counter event threshhold enabled and you use this command to set the counter value equal to the threshold, the counter event will NOT fire as it normally would on the next counter increment. This case only applies when the counter is set exactly equal to the threshhold.

Byte	Purpose	Value
0	Command	32
1	Counter	Counter to write (0 or 1)
2	High Byte	High byte of counter value (ignored on 8-bit counters)
3	Low Byte	Low byte of counter value
4	Reserved	

Back to Table of Contents

Set Counter Event Threshold

This command defines a counter threshold that will cause an event to fire as the counter value passes the threshold. On the ETH32 device, only Counter 0 supports these thresholds (although both counters support rollover thresholds).

An event is fired as a result of the counter surpassing the threshold, not meeting it. For example, with a threshold of 9, the counter's value would increment from 8 to 9 without firing the event, but it would fire as the counter incremented from 9 to 10.

The valid range for a counter event threshold is from 0 to the maximum possible counter value (65535 on 16-bit counters). The powerup default threshold is 0. The threshold has no other side-effects on the counter - it does not reset the counter to 0 like the rollover threshold.

Please note that defining an event threshold with this command does not enable the current connection to actually receive the event notifications when they occur. These must be enabled separately using the Enable Event Notifications command. Also note that event thresholds are common to all connections. Changing the thresholds will affect other connections if they are utilizing that particular counter event.

Byte	Purpose	Value
0	Command	34
1	Counter	Counter (must be 0 on the ETH32)
2	High Byte	High byte of counter event threshold (ignored on 8-bit counter)
3	Low Byte	Low byte of counter event threshold
4	Reserved	

Back to Table of Contents

Set Counter Rollover Threshold

This command defines the maximum permissible value for a counter. After the counter reaches the given value, the next count will cause the counter to be reset to 0 and fire a rollover event. For example, with a rollover threshold set to 0x35, the counter value will progress as follows: ..., 0x33, 0x34, 0x35, 0, 1, ... Because the comparisons and reset are done directly in hardware, no counts are missed during a rollover.

The valid range of the rollover threshold is from 0 to the maximum value of the counter (255 for 8-bit, 65535 for 16-bit). The powerup default rollover threshold is 255 for 8-bit counters and 65535 for 16-bit counters.

There is one special case involving rollover thresholds. When the counter value is manually set to EXACTLY the threshold value using the Write Counter Value command, the rollover will NOT occur and the rollover event will NOT fire on the next timer increment. Instead, the timer will increment past the threshold value. The event will not fire until the counter value has wrapped around and again exceeds the threshold. For example, suppose the rollover threshold is set to 10 on an 8-bit counter and the Write Counter Value command is used to set the counter value to 10. As the input line pulses, the counter value would increment as follows: 11, 12, ..., 255, 0, 1, ..., 10, 0, 1, ..., 10, 0, ...

Please note that defining a rollover threshold with this command does not enable the current connection to actually receive the rollover event notifications when they occur. These must be enabled separately using the Enable Event Notifications command. Also note that rollover thresholds are common to all connections. Changing the thresholds will affect other connections if they are utilizing that particular counter.

Byte	Purpose	Value
0	Command	36
1	Counter	Counter (0 or 1)
2	High Byte	High byte of counter rollover threshold (ignored on 8-bit counter)
3	Low Byte	Low byte of counter rollover threshold
4	Reserved	

Back to Table of Contents

Set PWM Clock State

This command enables or disables the PWM clock from counting. When the PWM clock is disabled the PWM outputs will be idle (not pulsing). The PWM clock may be enabled or disabled independently of whether the individual PWM channel outputs are enabled or disabled.

Byte	Purpose	Value
0	Command	38
1	Clock State	0 - PWM clock is disabled 1 - PWM clock is enabled
2	Reserved	
3	Reserved	
4	Reserved	

Back to Table of Contents

Set PWM Base Period

This command configures the main PWM clock to have a cycle period of the given number of counts. This defines the base frequency that will be used for the PWM channels. The base frequency is not individually selectable for each channel, so this command will affect all PWM outputs. Each complete PWM waveform will have a duration of (period + 1) PWM clock cycles. The PWM clock counts at a rate of 2 MHZ. This means, for example, that specifying a period of 99 would result in a frequency of 20 KHZ (2,000,000/(99+1)).

The base period is specified as a 16-bit number that may range from a value of 49 (40 KHZ) to a value of 65,535 (30.5 HZ).

Byte	Purpose	Value
0	Command	40
1	High Byte	High byte of the PWM base period
2	Low Byte	Low byte of the PWM base period
3	Reserved	
4	Reserved	

Back to Table of Contents

Set PWM Channel State

This command configures the state of the specified PWM channel. When a channel is disabled, the I/O pin will function as a normal digital I/O pin. When the channel is enabled, that I/O pin will be overridden and the pin will become the PWM output. However, note that the bit must be put into output mode using the Set Port Direction command. In the normal PWM state, the pin will be high for the given duty cycle period, and low for the rest of the PWM base period. In the inverted PWM state, the pin will be just the opposite - low for the amount of time specified by the duty cycle period, and high for the remainder.

Byte	Purpose	Value
0	Command	42
1	Channel	PWM channel (0 or 1)
2	Channel State	0 - disabled (normal I/O port operation) 1 - normal PWM 2 - inverted PWM
3	Reserved	
4	Reserved	

Back to Table of Contents

Set PWM Channel Duty Cycle Period

This command configures a PWM channel to have the specified duty cycle period. The period specifies the pulse length, specified as PWM clock counts less one. In other words, when the PWM channel state is in normal mode, the PWM output will be high for (period + 1) counts of the PWM clock and low for the remainder of the clock counts in the cycle.

Any 16-bit value can be specified for the period, from 0 to 65535. Note that if a duty cycle period is given that is greater than or equal to the current PWM base period, the output will be a solid high (in normal mode) or a solid low (in inverted mode). If a duty cycle period of 0 is given, the output will not be solid, but rather it will have a short spike during each period of the PWM clock.

Byte	Purpose	Value
0	Command	44
1	Channel	PWM channel (0 or 1)
2	High Byte	High byte of the duty cycle period
3	Low Byte	Low byte of the duty cycle period
4	Reserved	

Back to Table of Contents

Set One EEPROM Byte

This command stores a single byte into a specific location in the user-accessible EEPROM memory. Storing data into EEPROM is a relatively slow process. See the product user manual for specific timing information.

There are 256 bytes of user-accessible EEPROM, so the Address parameter to this command can be from 0 to 255 to select any of the 256 bytes.

Byte	Purpose	Value
0	Command	47
1	Address	EEPROM location into which the value should be stored.
2	Value	The value to store into EEPROM
3	Reserved	
4	Reserved	

Back to Table of Contents

Set Three EEPROM Bytes

This command stores three consecutive bytes into a specific location in the user-accessible EEPROM memory. Storing data into EEPROM is a relatively slow process. See the product user manual for specific timing information.

There are 256 bytes of user-accessible EEPROM, so the addresses to those bytes ranges from 0 to 255. If the Address parameter to this command is specified as 255, only one byte can be stored, so the First byte will be stored to that location. If the Address is specified as 254, the First byte will be stored to address 254, and the Second byte will be stored to address 255. Otherwise all three bytes will be stored as normal.

Byte	Purpose	Value
0	Command	48
1	Address	EEPROM starting location into which the values should be stored.
2	First byte	The first value to store into EEPROM (at Address)
3	Second byte	The second value to store into EEPROM (at Address+1)
4	Third byte	The third value to store into EEPROM (at Address+2)

Back to Table of Contents

Queries and Replies

The following command codes are used to retrieve information from the device. The process is very simple: A command block is sent indicating the specifics of the information desired (such as port 2's value). The device then sends back a command block with the data that was requested.

There are two conditions that apply to all query/reply pairs:

- Replies always use the same command code as the query that they respond to.
- A 1-byte sequence number field is included in all query/reply blocks to assist the application in correlating a reply with a specific query. The application may include any arbitrary sequence number in the query it sends and the device will simply return that sequence number with its response. In other words, whatever you choose to send, the device will return back to you.

Therefore, the command block structure of queries/replies is just slightly more specific:

Byte 0	Command Code
Byte 1	Sequence Number
Byte 2	Command-specific data
Byte 3	Command-specific data
Byte 4	Command-specific data

Ping

This command simply invokes a response from the board to verify that there is still connectivity. It has no effect on the device.

Query Format:

Byte	Purpose	Value
0	Command	1
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	1
1	Sequence Number	Returns the value specified in the query
2	Reserved	
3	Reserved	
4	Reserved	

Back to Table of Contents

Read Input Value

This command reads the current value of the specified port.

Query Format:

Byte	Purpose	Value
0	Command	3
1	Sequence Number	Arbitrary number used to pair query and reply
2	Port	Port Number: 0, 1, ...
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	3
1	Sequence Number	Returns the value specified in the query
2	Port	Returns the port number specified in the query
3	Value	The current value of the specified port
4	Reserved	

Back to Table of Contents

Read Output Register

This command reads and returns the current value of the output register for the specified port.

Query Format:

Byte	Purpose	Value
0	Command	4
1	Sequence Number	Arbitrary number used to pair query and reply
2	Port	Port Number: 0, 1, ...
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	4
1	Sequence Number	Returns the value specified in the query
2	Port	Returns the port number specified in the query
3	Value	Current value of the port’s output register
4	Reserved	

Back to Table of Contents

Get Port Direction

This command returns the current direction configuration of the specified port. A 0-bit in the returned direction register indicates that the bit is in input mode, while a 1-bit indicates the bit is in output mode.

Query Format:

Byte	Purpose	Value
0	Command	5
1	Sequence Number	Arbitrary number used to pair query and reply
2	Port	Port Number: 0, 1, ...
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	5
1	Sequence Number	Returns the value specified in the query
2	Port	Returns the port number specified in the query
3	Direction Register	0-bit indicates input mode, 1-bit indicates output mode
4	Reserved	

Back to Table of Contents

Get Analog-to-Digital Convertor State

This command returns the current status (enabled or disabled) of the Analog to Digital Convertor. This command can be used on any port but it is only useful for Port 3 since no other port besides Port 3 has an ADC.

Query Format:

Byte	Purpose	Value
0	Command	7
1	Sequence Number	Arbitrary number used to pair query and reply
2	Port	Port Number: 0, 1, ...
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	7
1	Sequence Number	Returns the value specified in the query
2	Port	Returns the port number specified in the query
3	Analog Status	0 - ADC disabled, digital only 1 - ADC enabled
4	Reserved	

Back to Table of Contents

Read Analog Channel

This command returns the current value of the specified analog channel. The returned values are only meaningful when the Analog to Digital Convertor is enabled. The analog readings are 10-bit values.

Query Format:

Byte	Purpose	Value
0	Command	9
1	Sequence Number	Arbitrary number used to pair query and reply
2	Channel	Channel number: 0-7
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	9
1	Sequence Number	Returns the value specified in the query
2	Channel	Returns the channel number specified in the query
3	Value (8 MSBs)	The eight Most Significant Bits of the current analog reading. If only 8-bit precision is needed, this value may be used directly.
4	Value (2 LSBs)	The two Least Significant Bits of the current analog reading. These two bits are left-aligned within this byte, meaning they are in bits 6 and 7. Bits 0-5 are unused.

Back to Table of Contents

Get Analog Event Definition

This command returns the analog event threshold definitions for the specified analog channel in the specified event bank. Please see the Set Analog Event Definition command description for an explanation of what the thresholds mean.

Query Format:

Byte	Purpose	Value
0	Command	12
1	Sequence Number	Arbitrary number used to pair query and reply
2	Bank and Channel	Bit 3: 0 for Bank 0, 1 for Bank 1 Bits 0-2: Channel number (0-7)
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	12
1	Sequence Number	Returns the value specified in the query
2	Bank and Channel	Returns the bank and channel numbers specified in the query
3	Lo-mark	Current low threshold
4	Hi-mark	Current high threshold

Back to Table of Contents

Get Analog Voltage Reference

This command returns the currently configured voltage reference setting for the Analog to Digital Convertor.

Query Format:

Byte	Purpose	Value
0	Command	17
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	17
1	Sequence Number	Returns the value specified in the query
2	Reference	Current reference setting. See the Set Analog Voltage Reference command for explanation of the possible values.
3	Reserved	
4	Reserved	

Back to Table of Contents

Get Analog Channel Assignment

This command returns the current multiplexer assignment for the specified analog channel.

Query Format:

Byte	Purpose	Value
0	Command	19
1	Sequence Number	Arbitrary number used to pair query and reply
2	Channel	Channel number: 0-7
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	19
1	Sequence Number	Returns the value specified in the query
2	Channel	Returns the channel number specified in the query
3	Source	Current 5-bit multiplexer value used to select the source of the analog reading for this channel
4	Reserved	

Back to Table of Contents

Get Serial Number Batch

This command returns the batch portion of the device’s serial number. The complete device serial number is made up of a batch ID and a unit ID, both of which are 16-bit numbers.

Query Format:

Byte	Purpose	Value
0	Command	21
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	21
1	Sequence Number	Returns the value specified in the query
2	Batch	High byte of the 16-bit batch ID
3	Batch	Low byte of the 16-bit batch ID
4	Reserved	

Back to Table of Contents

Get Serial Number Unit

This command returns the unit portion of the device’s serial number. The complete device serial number is made up of a batch ID and a unit ID, both of which are 16-bit numbers.

Query Format:

Byte	Purpose	Value
0	Command	22
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	22
1	Sequence Number	Returns the value specified in the query
2	Unit	High byte of the 16-bit unit ID
3	Unit	Low byte of the 16-bit unit ID
4	Reserved	

Back to Table of Contents

Get Product ID

This command returns a product identifier for the device. This can distinguish between product types when this same protocol is used to communicate with more than just an ETH32 device. The product ID is an 8-bit value. The ETH32 product ID is 105.

Query Format:

Byte	Purpose	Value
0	Command	23
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	23
1	Sequence Number	Returns the value specified in the query
2	Product ID	105
3	Reserved	
4	Reserved	

Back to Table of Contents

Get Firmware Release

This command returns the release number of the firmware programmed into the device. The release number is broken down into major and minor parts. For example, for release 2.001, the major number is 2 and the minor number is 1. Both are specified as 8-bit numbers.

Query Format:

Byte	Purpose	Value
0	Command	24
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	24
1	Sequence Number	Returns the value specified in the query
2	Major	Firmware Major Release number
3	Minor	Firmware Minor Release number
4	Reserved	

Back to Table of Contents

Successive Read

This command is similar to the Read Input Value command except that this command reads the current value of the specified port *repeatedly* until two successive reads have the same value. This can be useful when reading a data bus to ensure that a bogus value isn't returned when the bus is in transition. This command continues to read the specified port until one of the following conditions is met:

- Two successive reads give the same port value. This value is returned.

- The port was read the maximum number of times specified in the command without a match occurring.

Query Format:

Byte	Purpose	Value
0	Command	27
1	Sequence Number	Arbitrary number used to pair query and reply
2	Port	Port Number: 0, 1, ...
3	Max Reads	Maximum number of times to read the port (2-255)
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	27
1	Sequence Number	Returns the value specified in the query
2	Port	Returns the port number specified in the query
3	Times / Status	If successful, the number of times the port was actually read. Otherwise, 0 to indicate no match was ever seen.
4	Value	Port value. Last value read from the port, regardless of whether or not two successive reads ever matched.

Back to Table of Contents

Get Counter State

This command returns the current state of the specified counter.

Query Format:

Byte	Purpose	Value
0	Command	29
1	Sequence Number	Arbitrary number used to pair query and reply
2	Counter	Counter (0 or 1)
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	29
1	Sequence Number	Returns the value specified in the query
2	Counter	Returns the counter specified in the query
3	State	0 - Counter disabled 1 - Increment on falling edge 2 - Increment on rising edge
4	Reserved	

Back to Table of Contents

Read Counter Value

This command reads the current value of the specified counter. The counter value is not affected by reading it.

For 8-bit counters, the high byte of the returned value should be ignored.

Query Format:

Byte	Purpose	Value
0	Command	31
1	Sequence Number	Arbitrary number used to pair query and reply
2	Counter	Counter to read (0 or 1)
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	31
1	Sequence Number	Returns the value specified in the query
2	Counter	Returns the counter specified in the query
3	High Byte	The high byte of the counter value (ignore on 8-bit counters)
4	Low Byte	The low byte of the counter value

Back to Table of Contents

Get Counter Event Threshold

This command returns the counter event threshold defined for the specified counter. Please see the Set Counter Event Threshold command for an explanation of event thresholds.

For 8-bit counters, the high byte should be ignored.

Query Format:

Byte	Purpose	Value
0	Command	33
1	Sequence Number	Arbitrary number used to pair query and reply
2	Counter	Counter whose event threshold should be returned
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	33
1	Sequence Number	Returns the value specified in the query
2	Counter	Returns the counter specified in the query
3	High Byte	The high byte of the counter event threshold (ignore on 8-bit counters)
4	Low Byte	The low byte of the counter event threshold

Back to Table of Contents

Get Counter Rollover Threshold

This command returns the counter rollover threshold defined for the specified counter. Please see the Set Counter Rollover Threshold command for an explanation of rollover thresholds.

For 8-bit counters, the high byte should be ignored.

Query Format:

Byte	Purpose	Value
0	Command	35
1	Sequence Number	Arbitrary number used to pair query and reply
2	Counter	Counter whose rollover threshold should be returned (0 or 1)
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	35
1	Sequence Number	Returns the value specified in the query
2	Counter	Returns the counter specified in the query
3	High Byte	The high byte of the counter rollover threshold (ignore on 8-bit counters)
4	Low Byte	The low byte of the counter rollover threshold

Back to Table of Contents

Get PWM Clock State

This command returns the current state of the PWM clock, which indicates whether the clock is currently counting (enabled) or not (disabled).

Query Format:

Byte	Purpose	Value
0	Command	37
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	37
1	Sequence Number	Returns the value specified in the query
2	Clock State	0 - PWM clock is disabled 1 - PWM clock is enabled
3	Reserved	
4	Reserved	

Back to Table of Contents

Get PWM Base Period

This command returns the currently configured PWM base frequency. See the Set PWM Base Period command for details of the PWM base frequency.

Query Format:

Byte	Purpose	Value
0	Command	39
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reserved	
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	39
1	Sequence Number	Returns the value specified in the query
2	High Byte	High byte of the base period
3	Low Byte	Low byte of the base period
4	Reserved	

Back to Table of Contents

Get PWM Channel State

This command returns the current state of one of the PWM output channels on the device. See the Set PWM Channel State command for details on the various PWM channel states.

Query Format:

Byte	Purpose	Value
0	Command	41
1	Sequence Number	Arbitrary number used to pair query and reply
2	Channel	PWM channel (0 or 1)
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	41
1	Sequence Number	Returns the value specified in the query
2	Channel	Returns the PWM channel
3	Channel State	0 - disabled (normal I/O port operation) 1 - normal PWM 2 - inverted PWM
4	Reserved	

Back to Table of Contents

Get PWM Channel Duty Cycle Period

This command returns the currently configured duty cycle period for the specified PWM channel. See the Set PWM Channel Duty Cycle Period command for more information about the duty cycle period.

Query Format:

Byte	Purpose	Value
0	Command	43
1	Sequence Number	Arbitrary number used to pair query and reply
2	Channel	PWM channel (0 or 1)
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	43
1	Sequence Number	Returns the value specified in the query
2	Channel	Returns the PWM channel
3	High Byte	High byte of the duty cycle period
4	Low Byte	Low byte of the duty cycle period

Back to Table of Contents

Get/Reset Connection Flags

The ETH32 device maintains several flag bits for each individual active TCP/IP connection. The flags indicate conditions that are (or were) present for that connection. Currently, these flags are used to indicate whether any data that needed to be sent had to be discarded due to lack of queue space.

To understand how this is helpful, consider the following example. Suppose that digital events are enabled on port 0, bit 0 for a given connection to the ETH32. Now suppose that port 0, bit 0 begins pulsing rapidly, generating a steady stream of event notifications. Finally, suppose that the connection to the computer is having trouble. Due to the nature of TCP/IP, the event notifications must be retained in the queue of the ETH32 device until a TCP/IP acknowledgement for them has been received from the computer (in case they need to be retransmitted). If the TCP/IP acknowledgements do not come promptly and the events keep occurring, the queue will eventually fill up and the ETH32 device will be forced to simply discard any new event notifications.

Although this scenario is undesirable and should be avoided, if it does happen, it is helpful for the computer application to be able to detect that it happened and that data was lost. The flags keep track of this individually for each TCP/IP connection (that is, a full queue on one connection will not affect flags on another). Once a flag is set, it will remain set until it is manually reset back to zero by setting the appropriate field in this query block. Note that the flags are informational only - they do not affect the behavior of the device.

The flags also indicate what kind of data had to be discarded. More than one of these flags may be set at the same time.

- 0x01 (bit 0): Response to a query command (for example a Read Input Value query).
- 0x02 (bit 1): Digital event notification
- 0x04 (bit 2): Analog event notification
- 0x08 (bit 3): Counter event (rollover or threshold) notification

This command returns a byte containing any combination of these flags. The flags may also be optionally reset to zero (as a side-effect of the query) if the appropriate field in the query is set. In this case, the flags will only be reset to zero if the connection has enough space to queue up the reply data. In other words, the flags will not be lost if the response itself is unable to be queued.

The connection flags for new connections always start out as zero. When a Reset command is received, the flags for the connection it was received on are cleared, but the flags for any other active connections are not affected.

Query Format:

Byte	Purpose	Value
0	Command	45
1	Sequence Number	Arbitrary number used to pair query and reply
2	Reset	0 - Flags will not be affected by this command 1 - Flags will be cleared to zero if the reply is able to be successfully queued.
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	45
1	Sequence Number	Returns the value specified in the query
2	Reset	Returns the value specified in the query
3	Flags	The current flags for this connection. See above for the meaning of each bit.
4	Reserved	

Back to Table of Contents

Get Two EEPROM Bytes

This command retrieves two consecutive bytes from a specific location in the user-accessible EEPROM memory.

There are 256 bytes of user-accessible EEPROM, so the addresses to those bytes ranges from 0 to 255. If the Address parameter to this command is specified as 255, only one byte can be retrieved, so it will be returned as the First byte, and the Second byte will be returned as zero. Otherwise, both bytes will be retrieved from EEPROM and returned as normal.

Query Format:

Byte	Purpose	Value
0	Command	46
1	Sequence Number	Arbitrary number used to pair query and reply
2	Address	The EEPROM location from which values should be retrieved
3	Reserved	
4	Reserved	

Reply Format:

Byte	Purpose	Value
0	Command	46
1	Sequence Number	Returns the value specified in the query
2	Address	Returns the value specified in the query
3	First byte	Returns the value retrieved from EEPROM at Address
4	Second byte	Returns the value retrieved from EEPROM at Address+1

Back to Table of Contents

Notifications

Notifications are command blocks sent from the device to the application when some condition is met. They are not in direct response to any query or command received from the application, although commands can be used to enable or disable certain notifications from occurring. When a notification is received by an application, there is no response necessary from the application.

Digital Event Fired

This command notifies the application that a digital event has occurred. The command block includes details of what port and bits are effected, as well as the new value of the port reading after the event occurred.

Byte	Purpose	Value
0	Command	10
1	Port	Port number that the event occurred on
2	New Value	New value of the port after the event occurred
3	Changed	Bitmask indicating which bits on this port changed value. A 1-bit indicates that the value of the bit changed. This can be used to determine which bit(s) caused the event to fire.
4	Reserved	

Back to Table of Contents

Analog Event Fired

This command notifies the application that an analog event has occurred. An analog event fires whenever an analog channel changes state (low/high) based upon the thresholds defined in its analog event definitions. This command block includes details of what bank and channel event definition caused the event as well as the new value of the analog reading after the event occurred.

Byte	Purpose	Value
0	Command	14
1	Bank, Channel, Level	Bit 7: New level. Indicates whether analog reading went above (1) or below (0) the defined threshold. Bit 3: 0 for Bank 0, 1 for Bank 1 Bits 0-2: Channel number (0-7)
2	Old Value	The eight Most Significant Bits of the old reading just before the event occurred
3	New Value	The eight Most Significant Bits of the new reading that caused the event to fire
4	LSB's	Bits 6-7: Two Least Significant Bits of the New Value Bits 0-1: Two Least Significant Bits of the Old Value

Back to Table of Contents

Heartbeat

This command block is sent periodically from the device to the application. No response is needed from the application. The hearbeat serves two primary purposes:

- It notifies the application that the connection to the device is still working.
- Due to the TCP acknowledgement needed for the hearbeat command block data, it allows the device to verify that the remote system is still alive. This acknowledgement happens at the system TCP/IP stack level and requires no action from the application whatsoever.

A heartbeat command block is typically sent out every 4-5 minutes to all active connections. Note that because of the heartbeat, programmers must remember that they will be receiving data from the device that was not explicitely requested, even when no events are enabled.

Byte	Purpose	Value
0	Command	25
1	Reserved	
2	Reserved	
3	Reserved	
4	Reserved	

Back to Table of Contents

Counter Event Fired

This command notifies the application that an event has fired on a counter. Depending on the value of the Type field, this notification may indicate that the counter has rolled over or that the counter has exceeded its event threshold. The command block includes the number of times this has happened since the last event was fired. In almost all cases, this will be 1, but in the case of a fast-switching signal and/or a very low rollover threshold, the counter may exceed the threshold, be automatically reset, and count all the way up again again before the event can be serviced.

Byte	Purpose	Value
0	Command	34
1	Counter	Counter on which event fired (0 or 1)
2	Type	Event type: 0 for rollover, 1 for event threshold
3	Matches	Number of times this event has occurred since the last notification packet was sent
4	Reserved	

Back to Table of Contents