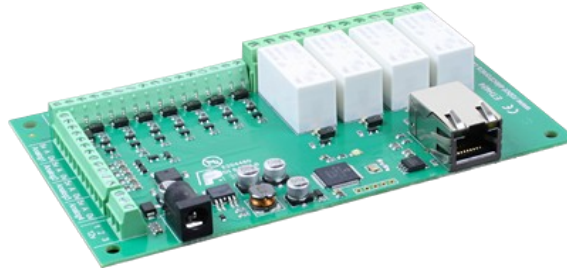


# ETH484 - 4 Relays at 16A, 8 Digital IO and 4 Analogue Inputs

## Technical Documentation



### Overview

The ETH484 provides four volt free contact relay outputs with a current rating of up to 16Amp each, 8 digital I/O and 4 analogue inputs. The module is powered from a 12vdc supply which can be regulated or unregulated. The DC input jack is 2.1mm with positive core polarity, DC supplies are required to supply at least 500mA at 12vdc. The relays are SPCO (Single Pole Change Over) types. The normally open, normally closed and common pins are all available on the screw terminals.

A new feature has been added that allows the digital inputs to remotely control relays or digital outputs on ETH002, ETH008, ETH484 or ETH8020. This offers the opportunity to construct a system where an input can control an output anywhere on the earth provided both locations are connected to the network/internet.

### Operating temperature

-40C to +70C

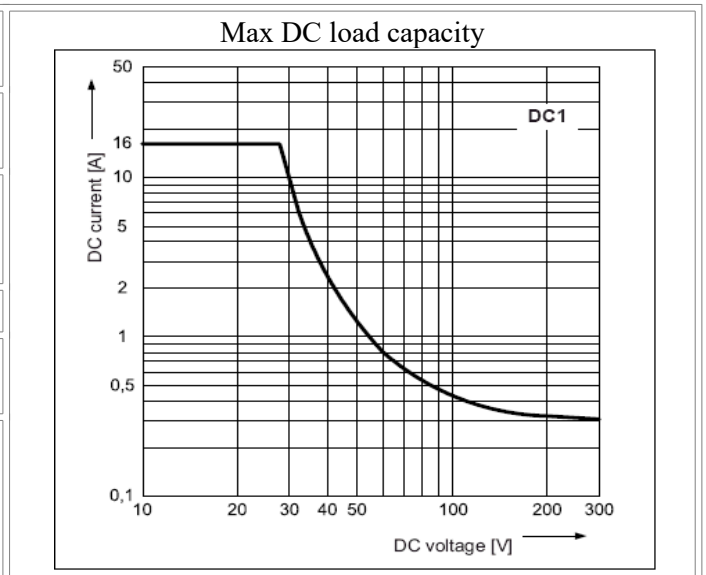
### LED indication

The ETH484 provides a red LED mounted immediately next to each relay to indicate whether it is in a powered state (LED on), there is also two LED's mounted in the Ethernet connector which will flash with Ethernet traffic. Finally there is a green power LED just above the processor.

### Relay power rating

If the contact load voltage and current of the relay are in the region enclosed by the solid and dotted lines in the figure below, the relay can perform stable switching operation. If the relay is used at a voltage or current exceeding this region, the life of the contacts may be significantly shortened.

load type	Typical applications	Rating
AC1	Non inductive or slightly inductive loads	16A @ 250V AC
AC15	Control of electromagnetic load (>72VA)	3A @ 120V AC 1.5A @ 240V AC
AC3	Control of motor	750W
DC1	Non inductive or slightly inductive loads	16A @ 24V DC
DC13	Control of electromagnetic loads	0.22A @ 120V DC 0.1A @ 250V DC



A full datasheet for the relays used on the ETH484 is here: [HF115FD datasheet](#)

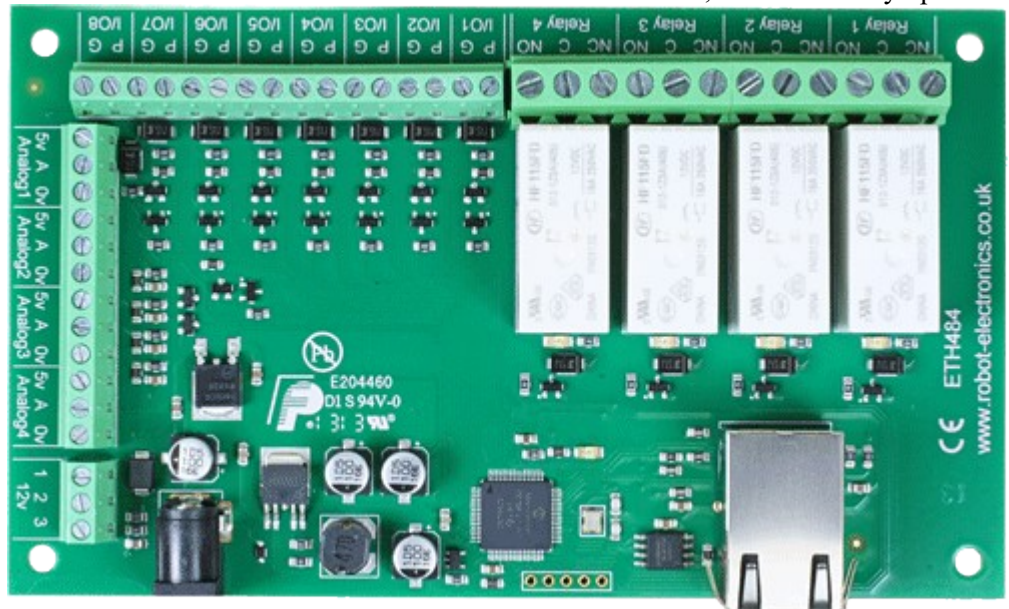
# Connections

Digital I/O  
P = Pin, G = 0v ground

16A VFC (Volt Free Contacts)  
NC = normally connected,  
C = common, NO = normally open

Analogue inputs  
5v = 5v dc output  
A = Input (0-3.3v)  
0v = 0v ground

12v dc output  
Direct from 2.1mm jack



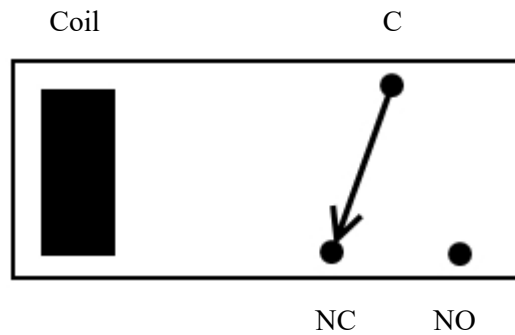
12v dc 2.1mm jack (+ve core)

RJ45 Ethernet

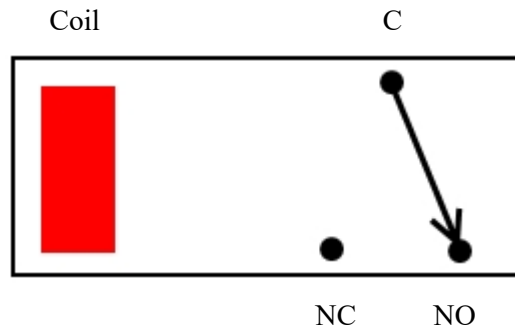
## Power relays

Four 16A volt free contact relays are provided for switching a common input between a normally closed output and a normally open output. The relay coil is powered by the 12vdc incoming supply on user command.

Relay in passive state



Relay in powered state



## Digital IO

Our Ethernet modules could potentially have many types of outputs. The ETH008 only has one type - Relays. The ETH484 has both Relay outputs and NPN Open Collector Transistor outputs. Activating a relay means turning the relay on. Likewise activating an output means turning the transistor on. This will cause it to sink current to 0v ground. If you had an LED connected from the output to 12v (via a resistor of course) it would light up. Other modules (not this one) could have PNP Open Collector Transistor outputs. These types will source current from the supply when active. So here's the point: Active does not mean a high voltage comes out. It means that the output has been activated. That could result in the output sinking or sourcing current, depending on its type. The ETH484 outputs will sink current (up to 100mA) when active.

The same principle applies to the ETH484's inputs. These are designed to allow you to directly connect a VFC (Volt Free Contact). This could be from other relay contacts, thermostat contacts, alarm contacts etc. When the contacts are closed the input will read as active. In fact anything that pulls the input pin down to 0v will read as active. Do not think of the I/O in terms of a high or low voltage output. Think of it in terms of Active (or on, something is actively driving the I/O), or inactive (or off, nothing is driving the I/O). It's a subtle point but one you need to be clear on.

<b>Examples</b>	
<p><b>Example input - connecting a switch</b></p> <p>Connecting a simple switch could not be easier, just wire the switch between a pin (P) and ground (G). When the switch closes the input will become active.</p>	
<p><b>Example output - connect a relay</b></p> <p>The ETH484 allows you to connect your own 12V relays (500mA max), the first coil pin of the relay is wired to the 12V supply terminals on the board, the other is wired to the output pin (P). When the output pin becomes active it is driven down to 0V ground, the relay will have 12V across the terminals and switch so COM is connected to NO.</p>	

<p><b>Representative Digital I/O Schematic</b></p>	
--	--

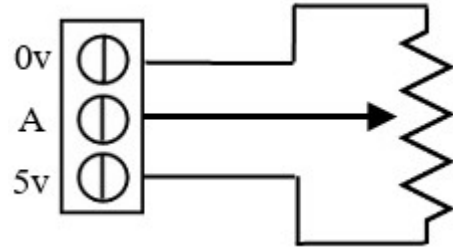
## Analogue Inputs

Four independent analogue input channels are provided for sampling voltages up to 3.3V. Each channel is also filtered with a 10k resistor and a 100n capacitor to stabilise high frequency jitter, there is also a pull down resistor so the port will read around 0 when nothing is connected. 5V inputs can be used, although the 3.3V to 5V region will merely read full scale.

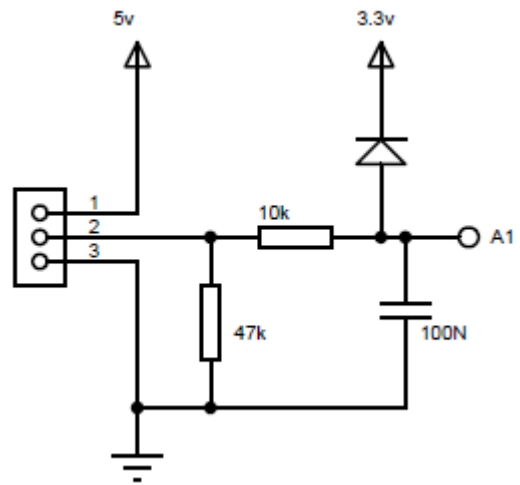
### Examples

#### Example input - connecting a potentiometer

Connecting a potentiometer should be the simplest of tasks, either end of the pot should be wired to the 5v and 0v respectively, the output pin of the pot is then wired into the analogue (A) pin. Please note the reading from the conversion will reach maximum at 3.3V.



#### Representative Analogue Schematic



# Network connection and Http access

## IP Addresses & DHCP Servers

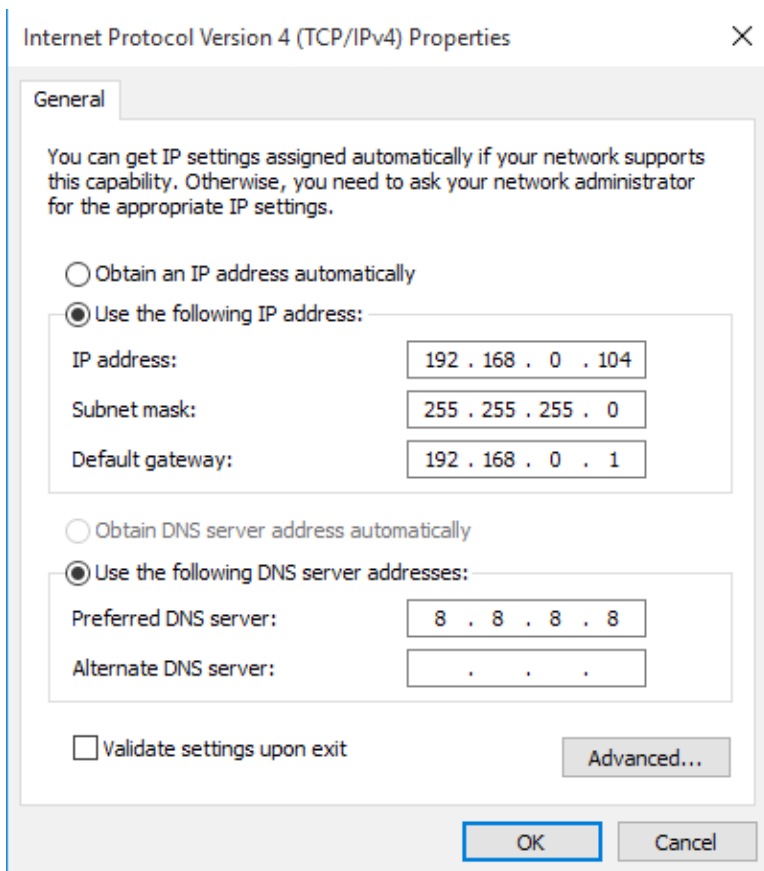
The easiest way to use the ETH484 is to connect it to a network with a DHCP server. In this case the ETH484 will have its IP address assigned automatically by the DHCP server.

If there is no DHCP server on the network, then a fixed IP address of 192.168.0.200 is used. To control the ETH484 using this fixed IP address your computer MUST be on the same subnet.

The next step is to set your computers IP address to 192.168.0.x where x is in the range of 1 to 255 but not 200 (the ETH484 is there!) or any other used IP addresses on the network.

The subnet mask dictates what IP addresses the PC can communicate with, we set this to 255.255.255.0 so the PC can talk to any module with an IP address of 192.168.0.x

The default gateway is likely to be the IP address that the internet connection is located at and the DNS server can be either your router address or Google provide a DNS service at 8.8.8.8..

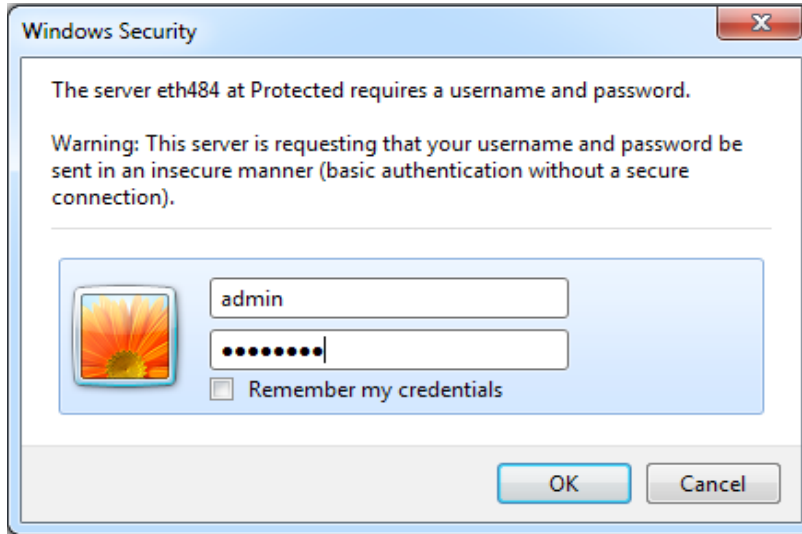


This image is for a PC running Windows 10. To get to the TCP/IP properties screen, go to: Control Panel->Network and Sharing Center->Local Area Connection->Properties. Select Internet Protocol Version 4(TCP/IPv4) from the scroll box and click Properties. That gets you to the dialog box shown left.

## Accessing the ETH484 from your browser

Having plugged in your 12vdc power supply and Ethernet connection the board will power up and request an IP address from the DHCP server, if no server is present then the board will default to IP address 192.168.0.200.

Now please start up your web browser and type `Http://eth484` into the address bar, please note this only works in windows. You will be prompted for a password as shown below:



The default login is:  
Username: admin  
Password: password

The ability to change these settings is shown in the configuration section

You should now see the following web page:

### ETH484 Test Application

Below you will see the current status of the ETH484 board, Relay1 and input/output1 are on the right hand side. Click the bullets to toggle the relays or turn change the output states on the board.

The status is updated in real time. You will see this page automatically update when changing states with commands over TCP/IP.

Firmware V1      Module Id 20  
Hardware V1      input voltage 12v

[Technical documentation](#)

[Configuration](#)

[Mapped inputs](#)



Analogue 1 1008      Analogue 3 1016  
Analogue 2 1012      Analogue 4 1020

Copyright © 2013 Devantech Ltd.

This web page will allow you to switch the relays on and off by clicking the relay buttons (the red/gray circles). It also contains a links to this technical documentation page, the network configuration and mapped inputs.

## Configuration

By clicking the configuration link it's possible to configure the ETH484 IP address and subnet mask together with the ability to set a password for entry to control screens. Gateway address and DNS address is configurable and is used with mapped inputs which are described in section below. The configuration page also offers the option to set a password that will be required to change any of the relay states or digital outputs using TCP IP commands, this is explained in the TCP/IP password section. All settings are saved to memory so be careful to remember the username and password! Default password settings are shown in the picture below. Version 4 now offers a latched outputs option, this will automatically save any permanent output changes (not pulsed) and restore them following power loss.

# Board Configuration

This section allows the configuration of the board's network settings.

**CAUTION:** Incorrect settings may cause the board to lose network connectivity.

MAC Address	<input type="text" value="00:04:A3:A7:FF:D2"/>
Host Name	<input type="text" value="ETH484"/>
<input checked="" type="checkbox"/> HTTP authentication	
Username:	<input type="text" value="admin"/>
Password:	<input type="text" value="password"/>
Port	<input type="text" value="17496"/>
<input type="checkbox"/> Enable DHCP	
IP Address	<input type="text" value="192.168.0.96"/>
Subnet Mask	<input type="text" value="255.255.255.0"/>
Gateway address	<input type="text" value="192.168.0.1"/>
DNS address	<input type="text" value="192.168.0.1"/>
<input type="checkbox"/> TCP/IP Password	<input type="text" value="password"/>
<input type="checkbox"/> Latched outputs	
<input type="button" value="Save Config"/>	

## Mapped inputs

Digital inputs are able to be configured to remotely control outputs on ETH008,ETH484 or ETH8020, this offers simple linking and versatile usage. An input in one country can control a output in another, or across a small network. There are eight independent inputs that can be mapped to eight different relays (on the same or different boards). The "Address of target board" field accepts an IP address or hostname (which will be converted to an IP address by the DNS server supplied in the board configuration above). If the module is on the local network then you can use the assigned IP address, if the target is over the internet then you need to supply the gateway in the configuration (internet source IP like your router) and the "Address of target board" is the IP address of the targets internet connection (to point at the router). Accessing the target via a router is dealt with in the section "**Access from the Internet**".

## Input mapping configuration

You can map an input to switch a relay over on another board using the configuration below.

<input type="checkbox"/> Enable Input1	Address of target board	<input type="text" value="192.168.0.1"/>	<input type="checkbox"/> Enable Input5	Address of target board	<input type="text" value="192.168.0.1"/>
	Port	<input type="text" value="17494"/>		Port	<input type="text" value="17494"/>
	Output number	<input type="text" value="1"/>		Output number	<input type="text" value="5"/>
	TCP Password: (if required)	<input type="text" value="password"/>		TCP Password: (if required)	<input type="text" value="password"/>
<input type="checkbox"/> Enable Input2	Address of target board	<input type="text" value="192.168.0.1"/>	<input type="checkbox"/> Enable Input6	Address of target board	<input type="text" value="192.168.0.1"/>
	Port	<input type="text" value="17494"/>		Port	<input type="text" value="17494"/>
	Output number	<input type="text" value="2"/>		Output number	<input type="text" value="6"/>
	TCP Password: (if required)	<input type="text" value="password"/>		TCP Password: (if required)	<input type="text" value="password"/>
<input type="checkbox"/> Enable Input3	Address of target board	<input type="text" value="192.168.0.1"/>	<input type="checkbox"/> Enable Input7	Address of target board	<input type="text" value="192.168.0.1"/>
	Port	<input type="text" value="17494"/>		Port	<input type="text" value="17494"/>
	Output number	<input type="text" value="3"/>		Output number	<input type="text" value="7"/>
	TCP Password: (if required)	<input type="text" value="password"/>		TCP Password: (if required)	<input type="text" value="password"/>
<input type="checkbox"/> Enable Input4	Address of target board	<input type="text" value="192.168.0.1"/>	<input type="checkbox"/> Enable Input8	Address of target board	<input type="text" value="192.168.0.1"/>
	Port	<input type="text" value="17494"/>		Port	<input type="text" value="17494"/>
	Output number	<input type="text" value="4"/>		Output number	<input type="text" value="8"/>
	TCP Password: (if required)	<input type="text" value="password"/>		TCP Password: (if required)	<input type="text" value="password"/>



## Mapping inputs to custom devices

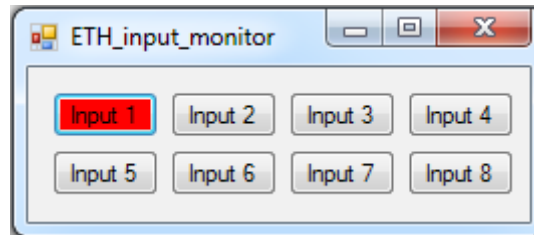
Sometimes it is useful to have a message sent on input change without the need for polling, you can use the input mapping for sending messages to an IP address. The transactions are described below along with an example of the custom device needing a password of “apple” and setting output number 1 active:

- The ETH484 sends a TCP packet with 0x79 (password entry) in the first byte, then the following bytes will be the password supplied above.  
So it would send 6 bytes: 0x79, 'a','p','p','l','e'
- To acknowledge a password match, respond with 1, else send 2  
You reply with 1 byte 0x01
- The ETH484 sends then sends digital active (0x20) or digital inactive (0x21) followed by the output number  
So it would send 2 bytes 0x20, 0x01
- Reply with a 0 for success, else send 1  
You reply with 1 byte 0x00

Note that the complete sequence must be followed, even if the password fails. If no password is supplied in the input mapping configuration the sequence will be followed with the absence of "a','p','p','l','e' bytes.

## Input monitoring example and source code

We have an example of mapping the inputs to a PC, it operates on the default port of 17494, the default password of "password" (although both are easily changed in the source code) and requires the ETH484 input mapping to be pointed at the host PC IP address. As it's having to constantly listen it's a multi thread program



The input monitor program is available as Visual C# express ready built installation files [here](#), or as project with source files [here](#). Visual studio express is provided free from Microsoft: <http://www.microsoft.com/exPress/download/>

## HTML commands DOAx and DOIx (V5+ firmware needed)

A customer requested feature, allowing the digital outputs to be switched by the Http get function such as used in some voice over ip phones (VOIP). You can use the Http get function to write to the io.cgi file with the following syntax:

***192.168.0.200/io.cgi?DOA2=10***

This would use the default address (192.168.0.200) and make output 2 active for 1 second.

Another example would be to set output 1 inactive for 10 seconds:

***192.168.0.200/io.cgi?DOI1=10***

You can test these functions by typing them directly into the address bar of most internet browsers. Also be aware that you may need to disable Http authentication in the Http configuration if your control device does not support it.

## TCP access and commands

The command set designed to provide consistent expansion and new features, they are sent over TCP/IP on port 17494 (0x4456). This is the default port, it can be changed in the configuration settings.

Five connections are allowed at any one time, these are independently protected but all using the same password as defined in the board configuration.

Command		Action
dec	hex	
16	0x10	Get Module Info - returns 3 bytes. Module Id (20 for ETH484), Hardware version, Firmware version.
32	0x20	Digital Active - follow with 1-4 to set relay on, or 9-16 for digital I/O then a time for pulsed output from 1-255 (100ms resolution) or 0 for permanent. Board will return 0 for success, 1 for failure
33	0x21	Digital Inactive - follow with 1-4 to turn relay off, or 9-16 for digital I/O then a time for pulsed output from 1-255 (100ms resolution) or 0 for permanent. Board will return 0 for success, 1 for failure
35	0x23	Digital Set Outputs - the first byte will set all relays states, All on = 255 (xxxx1111) All off = 0, 2nd byte sets digital outputs. Board will return 0 for success, 1 for failure
36	0x24	Digital Get Outputs - returns 2 bytes, the first corresponds with relays being powered and the 2nd corresponds with active digital outputs
37	0x25	Digital Get Inputs - returns 2 bytes, the 1st is always 0 as the relays are not inputs, the 2nd bytes bits correspond with the digital io, a high bit meaning input is active (driven low)
50	0x32	Get Analogue Voltage - follow with 1-4 for channel and ETH484 will respond with 2 bytes to form an integer (high byte 1st)
58	0x3A	ASCII text commands (V4+) - allows a text string to switch outputs, see section below
119	0x77	Get Serial Number - Returns the unique 6 byte MAC address of the module.
120	0x78	Get Volts - returns relay supply voltage as byte, 125 being 12.5V DC
121	0x79	Password Entry - see TCP/IP password, board will return 1 for success or 2 for failure
122	0x7A	Get Unlock Time - see section below
123	0x7B	Log Out - immediately re-enables TCP/IP password protection, board will return 0 for success

### Digital Active/Inactive Commands

These are 3 byte commands, The first byte is the command, 32 (active means on) or 33 (inactive).

Second byte is the output number, 1-4 for the relays, or 9 - 16 for digital outputs (marked I/O1 to I/O8 on the board).

Third byte is the on time. Set this to zero for non pulsed mode, or 1-255 for a pulse in 100mS intervals (100mS to 25.5 seconds).

For example:

**0x20** - turn the relay on command

**0x03** - relay 3

**0x32** (50) - 5 seconds (50 \* 100ms)

Board will return 0 for success, 1 for failure.

Note - All bytes in a command must be sent in one TCP/IP packet . Digital outputs pull the output pin down to 0v when active and pull up to 12v via a 10k resistor when inactive.

### Analogue input read

Four analogue inputs of 0v-3.3V at 10-bit resolution are provided, the conversion is performed when you send the "Get Analogue Voltage" command 0x32 and the channel number. The resulting 16 bit integer will be transmitted back in two bytes (high byte first), combine these for the result. The 10-bit conversion will be in the lower 10 bits of the 16 bit integer with the upper 6 bits being 0. Note that although the analogue input voltage is 0-3.3v, there is a 5v supply

available on the terminals. This is generally more useful as a supply for your own circuitry than 3.3v.

## TCP/IP Password

If this option is enabled in the Http configuration page then a password will be required to be entered before relay states can be changed. In the following example the password was set to "apple":

*0x79* - 1st byte in frame sent to ETH484 to indicate password entry  
*'a' (0x61)* - 2nd byte in frame (ASCII hex equivalent in brackets <http://www.asciitable.com/>)  
*'p' (0x70)* - 3rd byte in frame  
*'p' (0x70)* - 4th byte in frame  
*'l' (0x6C)* - 5th byte in frame  
*'e' (0x65)* - 6th byte in frame

These 6 bytes are then transmitted in the same transaction to the ETH484 and if the password is correct then 1 will be transmitted back, a failure will send 2.

The board will now accept changes from the device that entered the password. If communication becomes idle for more than 30 seconds then the password protection is re-enabled. There is also a log-out command of 0x7B to enable the protection immediately.

## Get Unlock Time

Returns TCP/IP password protection status:

*0* - password protection is enabled and password entry is required before changes can be made  
*1-30* - seconds until TCP/IP password protection is re-enabled. All authorised commands set the timer back to 30 seconds (including this one).  
*255* - TCP/IP password is not enabled.

## ASCII text commands DOA and DOI (V5+ firmware needed)

Following customer request we have added a feature that allows the outputs to be switched using an ASCII string, devices like a Mobotix camera can now switch relays with simple strings.

The string for activating output1 for 5 seconds is formatted using comma separated variables with the following syntax:

***":DOA,1,50,password"***

To break this down ":" (0x3A) at the start of the string indicates that there is an ASCII message to follow, "DOA" is digital output active, "1" is the output number, then "50" for 5 seconds (50 \* 100ms) and finally the TCP password (if applicable). If I wanted to make output 2 inactive for 3 seconds I would use:

***":DOI,2,30,password"***

To break this down ":" (0x3A) at the start of the string indicates that there is an ASCII message to follow, "DOI" is digital output inactive, "2" is the output number, then "30" for 3 seconds (30 \* 100ms) and finally the TCP password (if applicable). Assuming no password is used the previous command would simply be:

***":DOI,2,30 "***

## Access from the Internet

The ETH484 can be controlled over the internet almost as easily as on your local network. Your network will most likely be connected to the internet with a broadband router. This will provide NAT (Network Address Translation) and Firewall services. To access the ETH484 from the internet you will need to open up port 17494 (0x4456) to allow incoming TCP connections. Be careful not to open up any other ports. There are a wide variety of routers and we cannot give details for all of them. If in doubt ask your system administrator for assistance. The following shows how to open up a port on a Netgear DG834 router.

### Add Services

#### Service Definition

Name:	<input type="text" value="ETHRLY484 x"/>
Type:	<input type="text" value="TCP"/>
Start Port:	<input type="text" value="17496"/>
Finish Port:	<input type="text" value="17496"/>

After logging on to your routers setup page, the first thing to do is create a new service. Click on the "Services" menu then "Add Custom Service". Enter a name for the service, select TCP and enter the ETH484's port address for both the start and finish ports. Click "Apply".

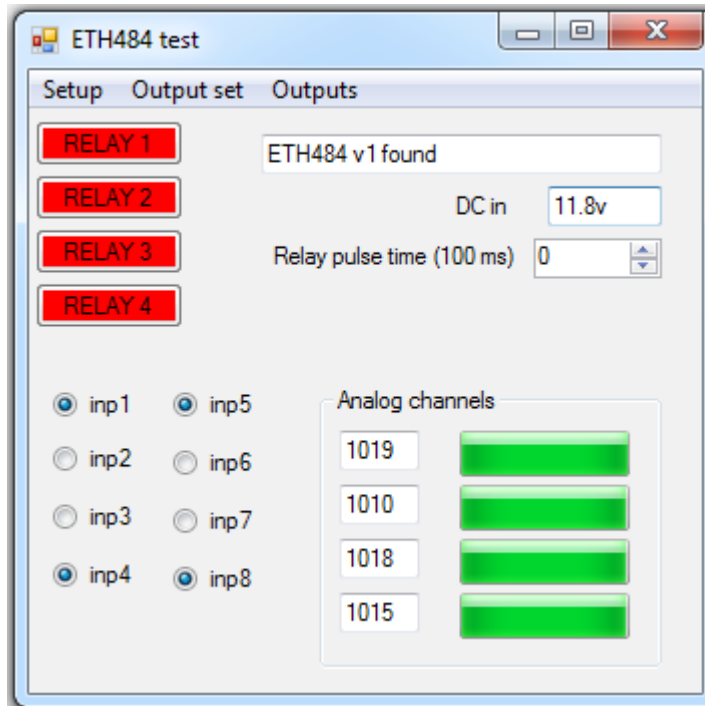
### Inbound Services

Service	<input type="text" value="ETHRLY484(TCP:17496)"/>
Action	<input type="text" value="ALLOW always"/>
Send to LAN Server	<input type="text" value="192"/> . <input type="text" value="168"/> . <input type="text" value="0"/> . <input type="text" value="96"/>
WAN Users	<input type="text" value="Any"/>
start:	<input type="text"/> . <input type="text"/> . <input type="text"/> . <input type="text"/>
finish:	<input type="text"/> . <input type="text"/> . <input type="text"/> . <input type="text"/>
Log	<input type="text" value="Always"/>

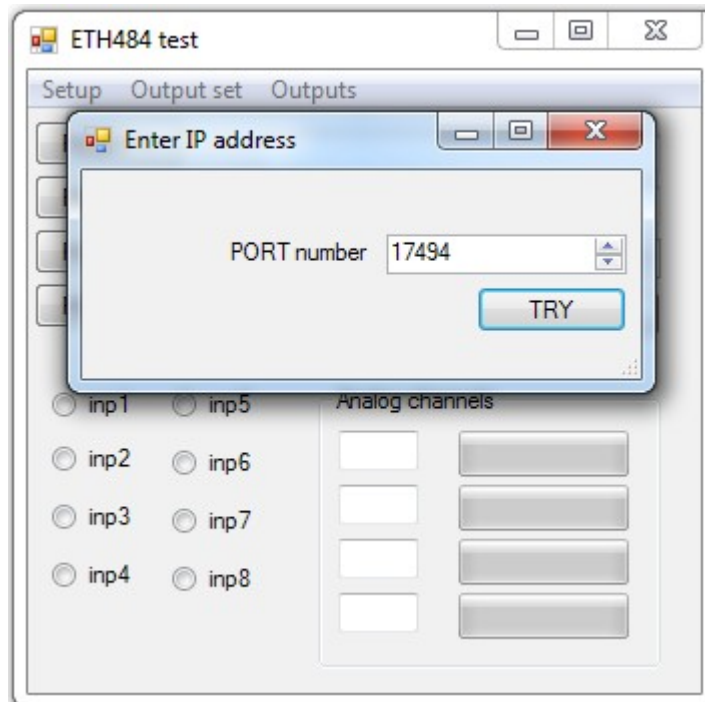
Now go to the "Firewall Rules" menu and click "Add" in the Inbound services section. Select the ETH484 service and ALLOW always. The "Send to LAN Server" IP address is the ETH484's IP address, 192.168.0.96 in the example above but check what it is on your network. Click "Apply" and that's it. The ETH484 is now accessible over the internet. Before you close the routers setup pages, go to the "Router Status" menu and make a note of its ADSL port IP address. This is the routers internet facing IP address.

## TCP Test program and example source code

To get the ETH484 up and running in the minimum amount of time we have put together an [example program](#) and also full [C# source code](#) for this program. You may examine this code or use it as a starting point for your own application.



Visual studio express is provided free from Microsoft: <http://www.microsoft.com/exPress/download/>



The test program is able to control an ETH484 on the local network or over the internet using your router port forwarding, an example of the setup procedure is described on the page above. To point the example program at your router go to menu Setup, Set Comport and click "Custom IP", then enter the Router IP and desired port.

## Android & iPhone Apps

We have a free app IO network available for Android and iPhone to remotely control your relays, download from Google Play or iTunes. Search for "Devantech" and you will find the app.

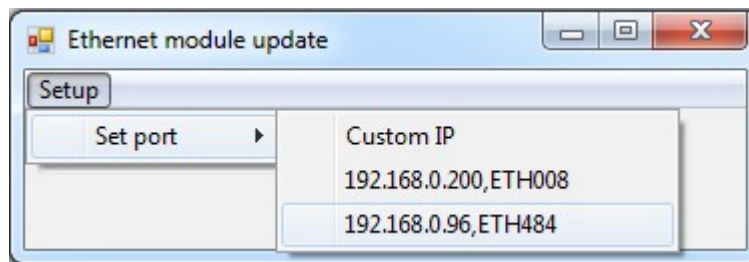
# Firmware Updates

The firmware can be fully updated by re-flashing the board using our custom windows program. The updates currently available are:

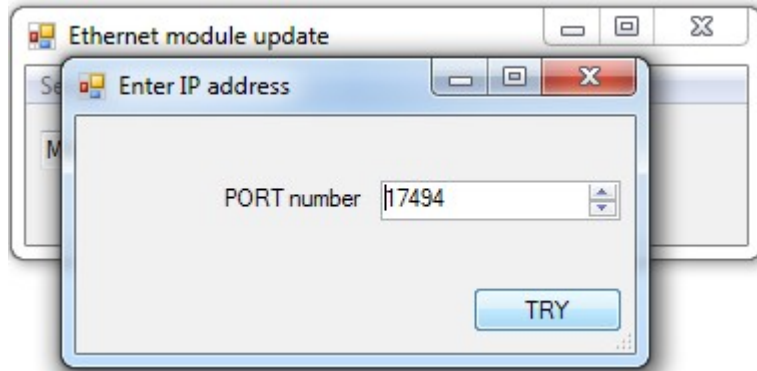
Version	Date of update	Reason for update
<a href="#">V2</a>	22/10/13	ETH484 will now cease attempts to change an output state on target board if the password is incorrect. It will recommence attempts following the input state changing
<a href="#">V3</a>	10/01/14	Inputs can now be mapped to outputs on the same board
<a href="#">V4</a>	13/02/14	DNS and Gateway string lengths corrected in Http config page, new latching function added to config and board name now appears correctly on NBNS (WINS) meaning routers display board name
<a href="#">V5</a>	11/07/14	ASCII messages added in tcp, security now on all Http pages, Http set output commands added
<a href="#">V6</a>	13/08/14	Fixed ASCII command buffer overruns
<a href="#">V7</a>	25/02/15	Corrected issue with Http password length of 12 characters
<a href="#">V8</a>	12/02/16	Voltage input reading via tcp corrected
<a href="#">V9</a>	01/12/16	Added numbering to inputs, outputs and relays on html page
<a href="#">V10</a>	03/04/17	Mapped inputs socket closure correction
<a href="#">V11</a>	07/02/19	CGI file now returns full board status

Using the [Ethernet module update](#):

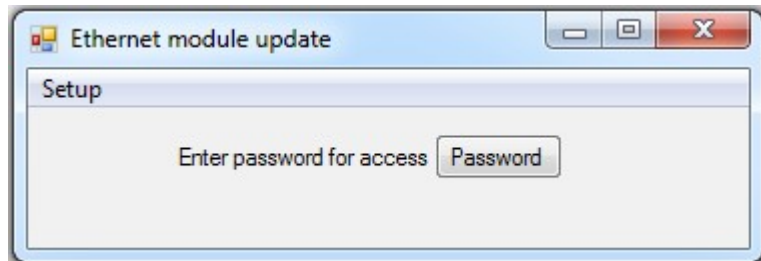
First you need to connect to the board, the program will scan the local network and provide any boards in a list that it can see.



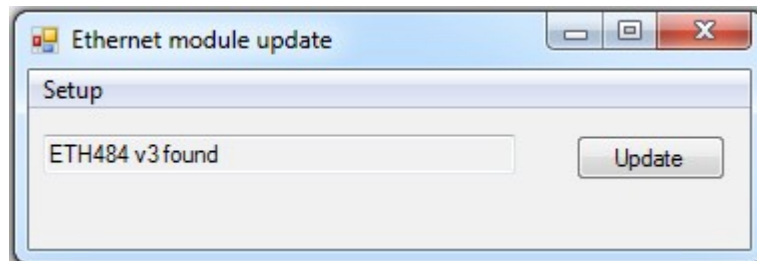
Next you will need to specify what port the board is located on, this will be 17494 unless you have changed it in the settings.

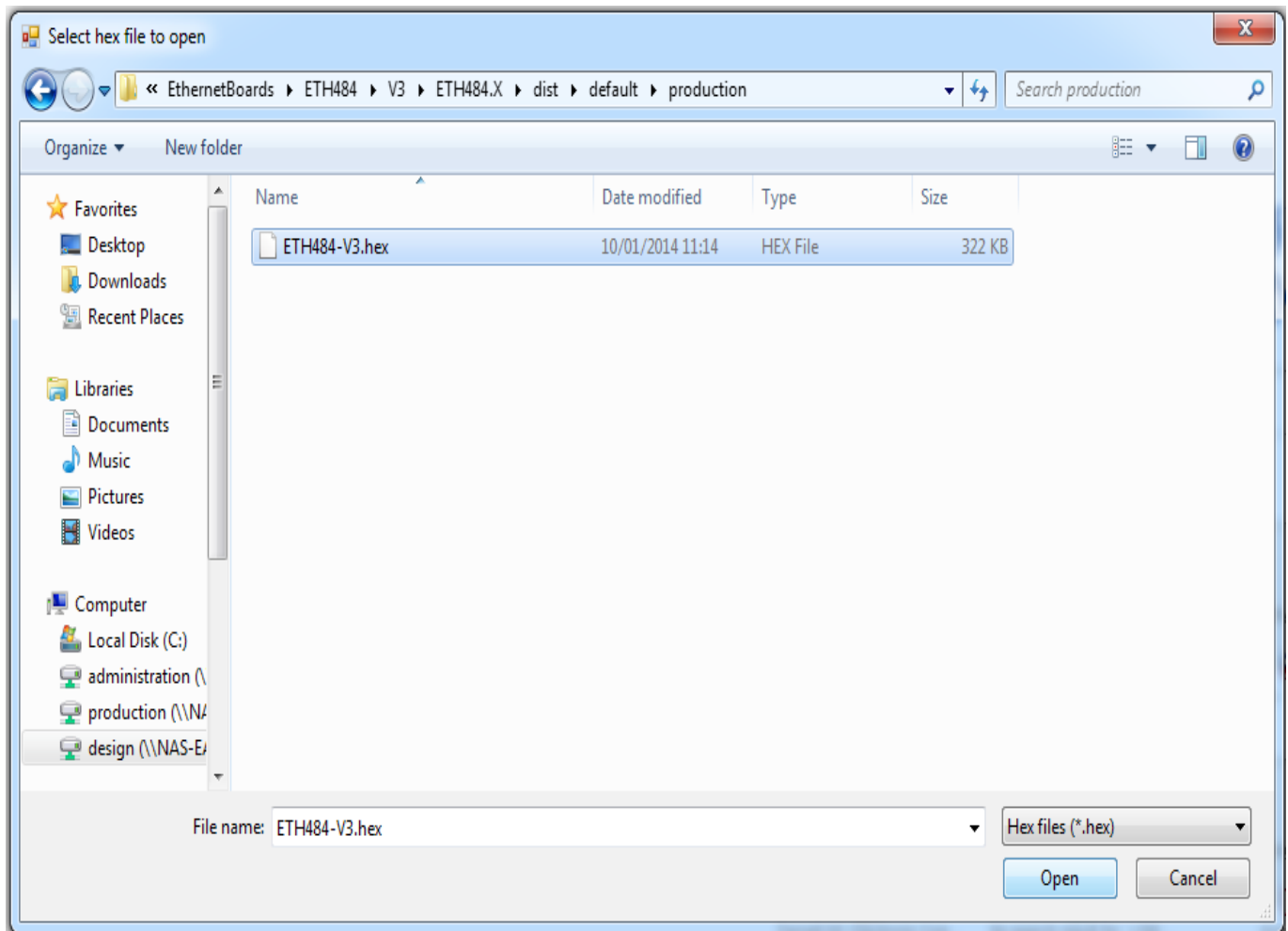


If there is a TCP/IP password the program will require it to be entered by clicking the password button.

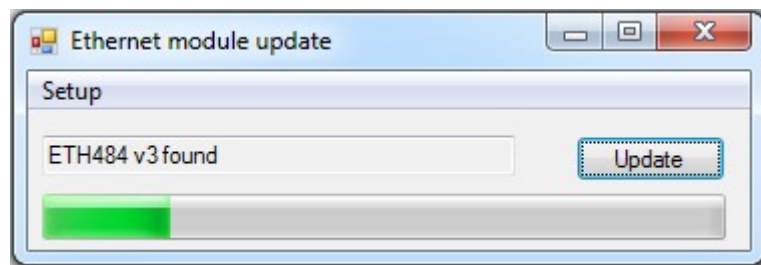


Following a successful connection you will now be able to click the update button to select a file



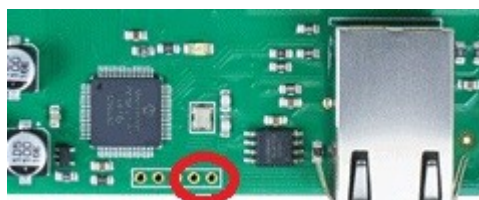


Once the file has been opened the programming sequence will now begin, it's fairly short and following its completion the board will reset (automatically) and the LED will flash quickly while the programming sequence completes. Do not turn the power off in this sequence.



### Factory Reset

Should it be necessary to reset the ETH484 to its shipped condition then the end two contacts of the row of 5 holes near the large chip on the side nearest the Ethernet connection must be shorted together at board power up. The green LED should then flash as the settings are reset, please wait until the LED finishes flashing and do not remove power during this period.





# Board dimensions

