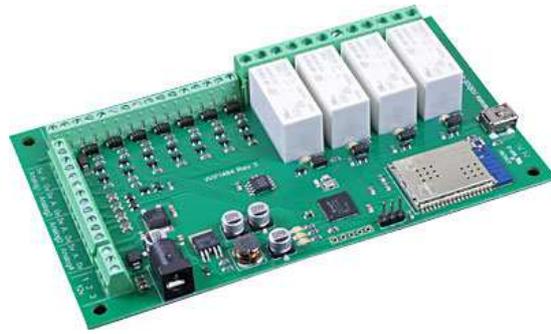


WiFi484 - 4 Relays at 16A, 8 Digital IO and 4 Analogue Inputs

Technical Documentation



Overview

The WiFi484 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. Digital inputs are able to be configured to remotely control outputs on all of our new ethernet relay boards (both wired and WIFI), the older ETH-RLY02 and ETH-RLY16 are not compatible however. 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 WiFi484 provides an LED to indicate each of the relay states and three for board status, these are:

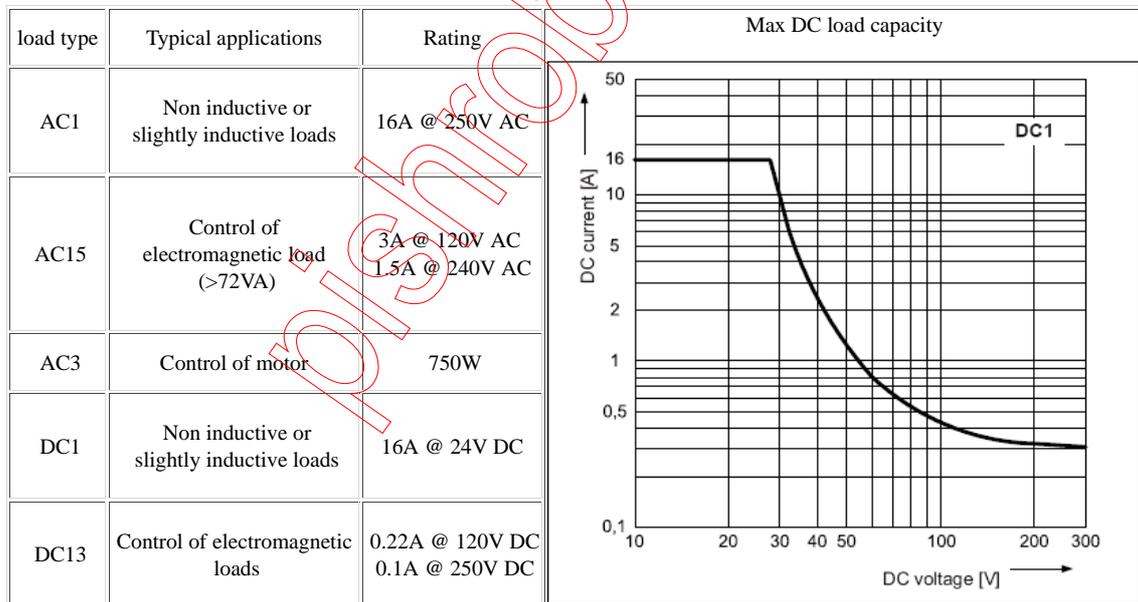
Board power - red LED

WiFi connected - yellow LED

USB connected - green LED

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.



A full datasheet for the relays used on the WiFi484 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

All settings are saved to memory so be careful to remember the username and password! Default password settings are shown in the picture below. There is also 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="00:1E:C0:14:8C:5B"/>
Host Name	<input type="text" value="WIFI484"/>
<input checked="" type="checkbox"/> HTTP authentication	
Username:	<input type="text" value="admin"/>
Password:	<input type="text" value="password"/>
Port	<input type="text" value="17494"/>
<input type="checkbox"/> Enable DHCP	
IP Address	<input type="text" value="192.168.0.94"/>
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 all of our new ethernet relay boards (both wired and WIFI), the older ETH-RLY02 and ETH-RLY16 are not compatible however. The mapped inputs offer 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		<input type="checkbox"/> Enable Input5	
Address of target board	<input type="text" value="192.168.0.1"/>	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		<input type="checkbox"/> Enable Input6	
Address of target board	<input type="text" value="192.168.0.1"/>	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		<input type="checkbox"/> Enable Input7	
Address of target board	<input type="text" value="192.168.0.1"/>	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		<input type="checkbox"/> Enable Input8	
Address of target board	<input type="text" value="192.168.0.1"/>	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"/>
<input type="button" value="Save Config"/>			

Mapping inputs to custom devices

Following customer requests for obtaining input states without the need for polling the WIFI484, this can be achieved with the existing input mapping function.

If you would like the inputs to be mapped to a custom device then we have a simple command structure to achieve this, the WIFI484 will send the commands in blue, your device will respond with commands in yellow.

A TCP packet with 0x79 (password entry) in the first byte, then the following bytes will be the password supplied above

To acknowledge a password match, respond with 1, else send 2

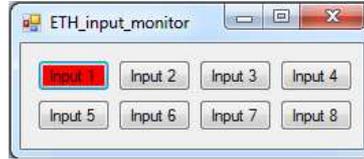
Digital active (0x20) or Digital inactive (0x21) followed by the output number

Reply with a 0 for success, else send 1

Note that the complete sequence must be followed, even if the password fails.

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 WIFI484 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 Visual C# express project with source files [here](#). Visual studio express is provided free from Microsoft: <http://www.microsoft.com/exPress/download/>

Email

An email can be sent in response to a change of digital input state, there can be two different messages for active and inactive transitions. The email option can be turned off, send on active, send on inactive or send on both. Each input can send emails to different addresses if required.

Support is provided for SSL so gmail can be used, however TLS that is required for hotmail is unsupported. Gmail is a free service though, so you can set up a dedicated account if need be.

Email

Outgoing email server connection details:

Server name	smtp.gmail.com
Port	587
Username:	myemail@gmail.com
Password	password
Connection Security	SSL

Message setup

Input 1	Disabled
Email sending	Disabled
Email address	someone@address.com
From	Name: WIFI484
	Email address: from@me.com
Subject	Alarm status
Message for active transition	ALARM condition has been raised
Message for inactive transition	ALARM condition has been cancelled

INP1 INP2 INP3 INP4 INP5 INP6 INP7 INP8 SAVE AND EXIT

Factory Reset

Should it be necessary to reset the WIFI484 to its shipped condition then a link is provided for factory reset. The link should be moved from its normal position to the reset position before board power up. The red power LED should then flash as the settings are reset, please wait until the LED finishes flashing and do not remove power during this period.

Firmware Updates

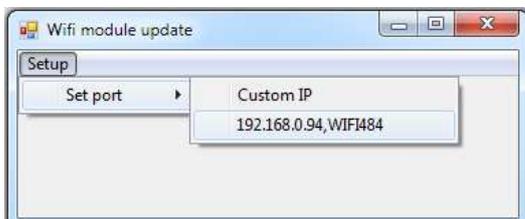
The firmware is fully updateable by re-flashing the board using our custom windows program. The updates currently available are:

[V2 - 14/07/14](#) - ASCII messages added in tcp, security now on all http pages, http set output commands added

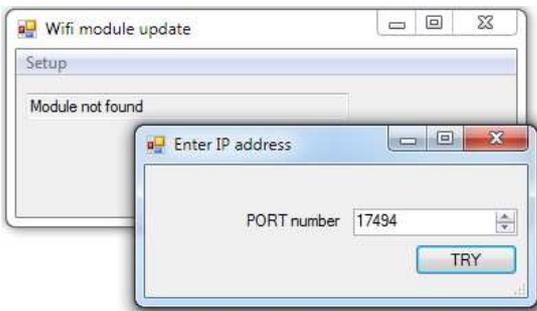
[V3 - 22/07/14](#) - Fixed bug with email messages not being saved.

Using the [Wifi 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.



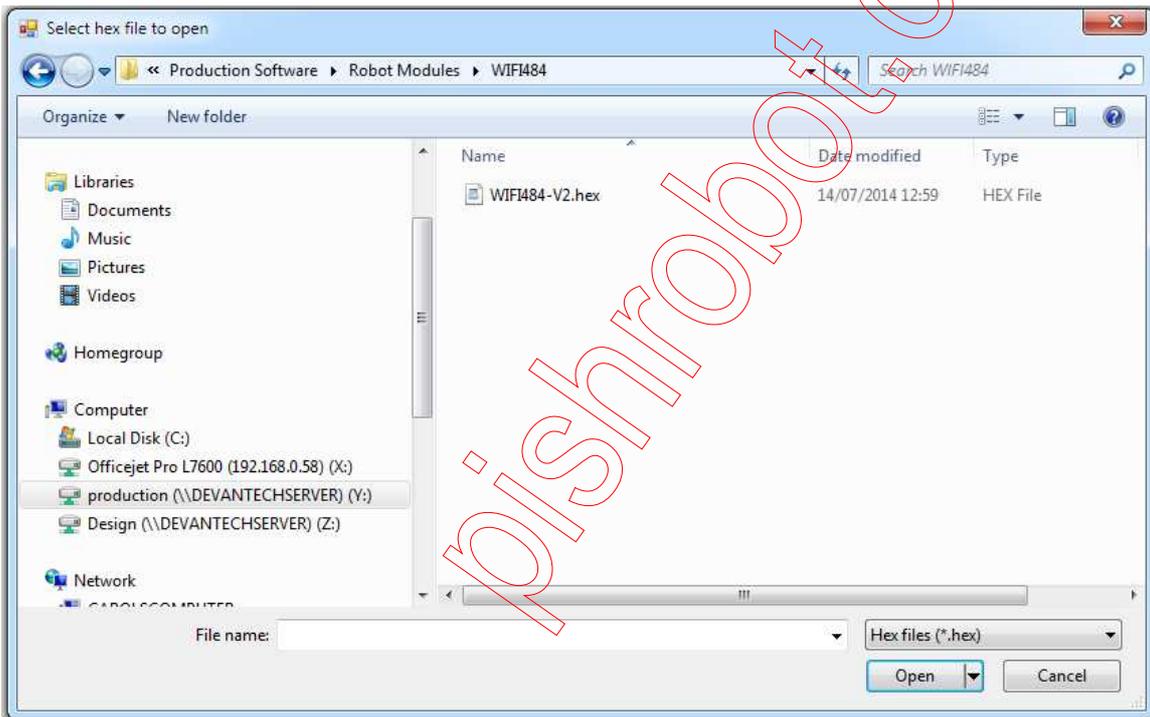
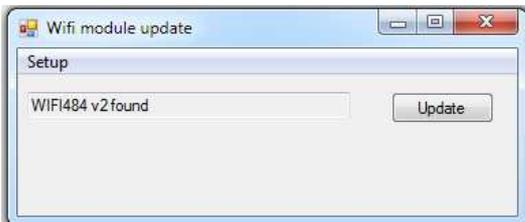
Next you will need to specify what port the board is located on, this will be 17494 unless you have changed it in the settings.



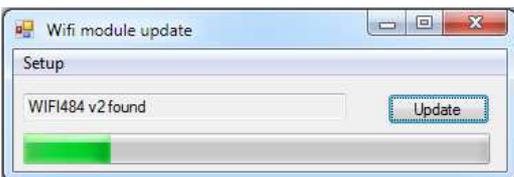
If there is a TCP/IP password the program will require it to be entered by clicking the password button.



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 it's 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.



WIFI484 Command Set

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	10	Get Module Info - returns 3 bytes. Module Id (22 for WIFI484), Hardware version, Firmware version.

32	20	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	21	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	23	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	24	Digital Get Outputs - returns 2 bytes, the first corresponds with relays being powered and the 2nd corresponds with active digital outputs
37	25	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	32	Get Analogue Voltage - follow with 1-4 for channel and WIFI484 will respond with 2 bytes to form an integer (high byte 1st)
58	3A	ASCII text commands (V4+) - allows a text string to switch outputs, see section below
119	77	Get Serial Number - Returns the unique 6 byte MAC address of the module.
120	78	Get Volts - returns relay supply voltage as byte, 125 being 12.5V DC
121	79	Password Entry - see TCP/IP password
122	7A	Get Unlock Time - see section below
123	7B	Log Out - immediately re-enables TCP/IP password protection

Active and Inactive I/O - What do we mean by that?

Our Ethernet modules could potentially have many types of outputs. The ETH008 only has one type - Relays. The WIFI484 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 WIFI484 outputs will sink current (up to 100mA) when active.

The same principle applies to the WIFI484'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.

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 un-timed operation, 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

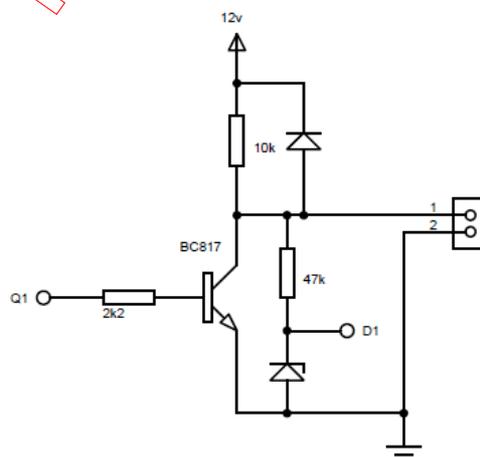
0x32 (50) - 5 seconds (50 * 100ms)

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.

Digital inputs

Digital Inputs can be read with a command of 0x25. The two bytes returned will be encoded with each bit corresponding to whether the input is active or inactive. A high bit means the pin is either being pulled low (active) by the output or has an external device pulling low. To use the pins as inputs, the corresponding output must be made inactive. The first byte returned will always be zero. The 8 inputs are in the second byte.

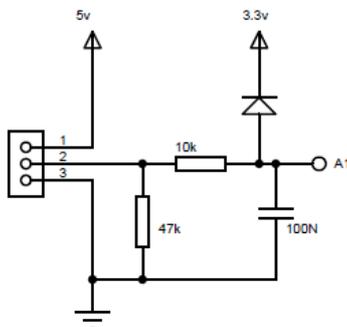


Representative Digital I/O Schematic.

Analogue Inputs

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.



Representative Analogue Input Schematic.

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 WIFI484 to indicate password entry

'a' (0x61) - 2nd byte in frame (ASCII hex equivalent in brackets, full table is available at <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 WIFI484 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 to 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 (V2+ 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 output 1 for 5 seconds is formatted using comma separated variables with the following syntax:

":DOA,1,50,password"

To break this down ":" (hex 3A) 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 (50x100ms) 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 ":" (hex 3A) 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 (30x100ms) and finally the TCP password (if applicable).

Assuming no password is used the previous command would simply be:

":DOI,2,30 "

HTML commands DOAx and DOIx (V2+ firmware needed)

Another 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 10 seconds.

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.

IP Addresses & DHCP Servers

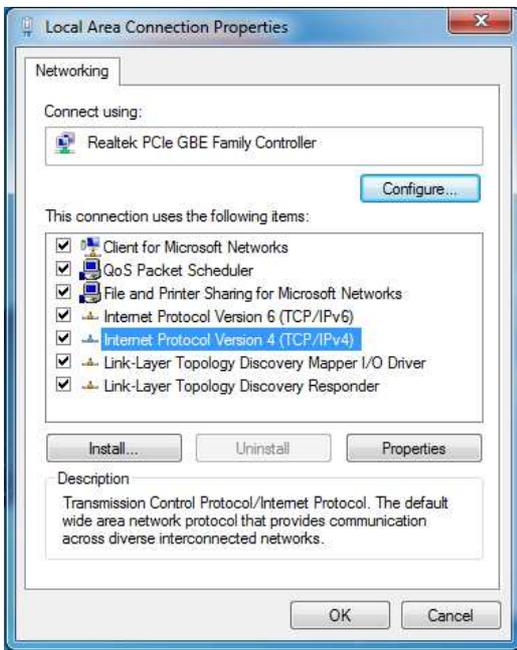
The easiest way to use the WIFI484 is to connect it to a network with a DHCP server. In this case the WIFI484 will have its IP address assigned automatically by the DHCP server.

If there is no DHCP server on the network, then a fixed IP address of 192.168.0.200 is used. To control the WIFI484 using this fixed IP address your computer MUST be on the same subnet.

The next step is to set your computer's IP address to 192.168.0.x where x is in the range of 1 to 255 but not 200 (the WIFI484 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 7.

To get to the TCP/IP properties screen, go to:

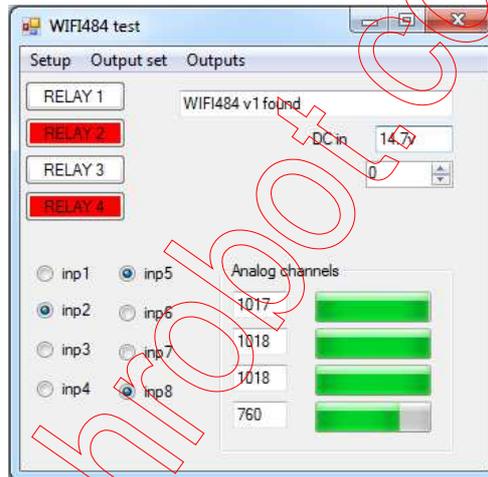
Control Panel->Network and Sharing Center->Local Area Connection->Properties.

Select Internet Protocol Version 4(TCP/IPv4) from the scroll box and click Properties.

That gets you to the dialog box shown left.

Test program and example source code

To get the WIFI484 up and running in the minimum amount of time we have put together an example program to demonstrate the functionality of the module. We provide the full source code for this program. You may examine this code to see how it works or use it as a starting point for your own application.



Visual studio express C# examples

The test program is available as Visual C# express ready built installation files [here](#), or as Visual C# express project with source files [here](#).

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

Access from the Internet

The WIFI484 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 WIFI484 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 WNR2200 router.

ADVANCED Home

Setup Wizard

WPS Wizard

▶ Setup

▶ USB Storage

▶ Security

▶ Administration

▼ Advanced Setup

[Wireless Settings](#)

[Wireless Repeating Function](#)

[Port Forwarding / Port Triggering](#)

[Dynamic DNS](#)

[Static Routes](#)

[Remote Management](#)

[USB Settings](#)

[UPnP](#)

[IPv6](#)

[Traffic Meter](#)

Ports - Custom Services

Cancel

Apply

Service Name:

Protocol:

External Starting Port: (1-65534)

External Ending Port: (1-65534)

Use the same port range for Internal port

Internal Starting Port: (1-65534)

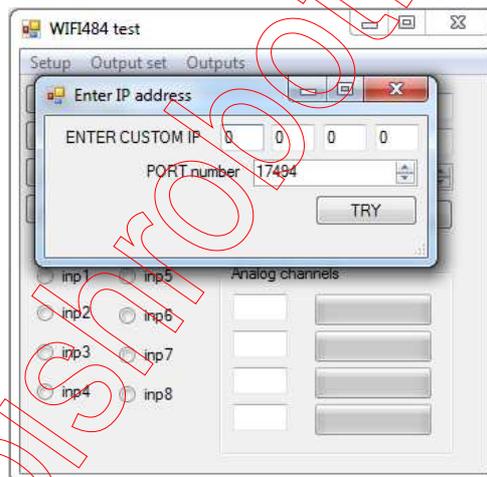
Internal Ending Port:

Internal IP address: . . .

Or select from currently attached devices

	IP Address
<input type="radio"/>	192.168.0.101
<input type="radio"/>	192.168.0.106
<input type="radio"/>	192.168.0.100
<input type="radio"/>	192.168.0.58
<input type="radio"/>	192.168.0.3
<input type="radio"/>	192.168.0.104
<input type="radio"/>	192.168.0.105
<input type="radio"/>	192.168.0.130
<input type="radio"/>	192.168.0.107
<input type="radio"/>	192.168.0.102

When applied you should now be able to talk to the router over the internet at the IP address designated by your internet provider. When a data packet arrives on port 17494 the router will pass the packet on to the local address on the network (192.168.0.93) on port 17494.



To test this you will need a computer that has its own internet connection and is NOT connected to the same network as the WIFI484. Download and run the [test program](#) above and select Custom IP. In the pop-up box enter your routers internet facing IP address. Click on "Try IP" and it will connect you to the WIFI484 just as if it were on your own network.

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.

Board dimensions

pishrobot.com

