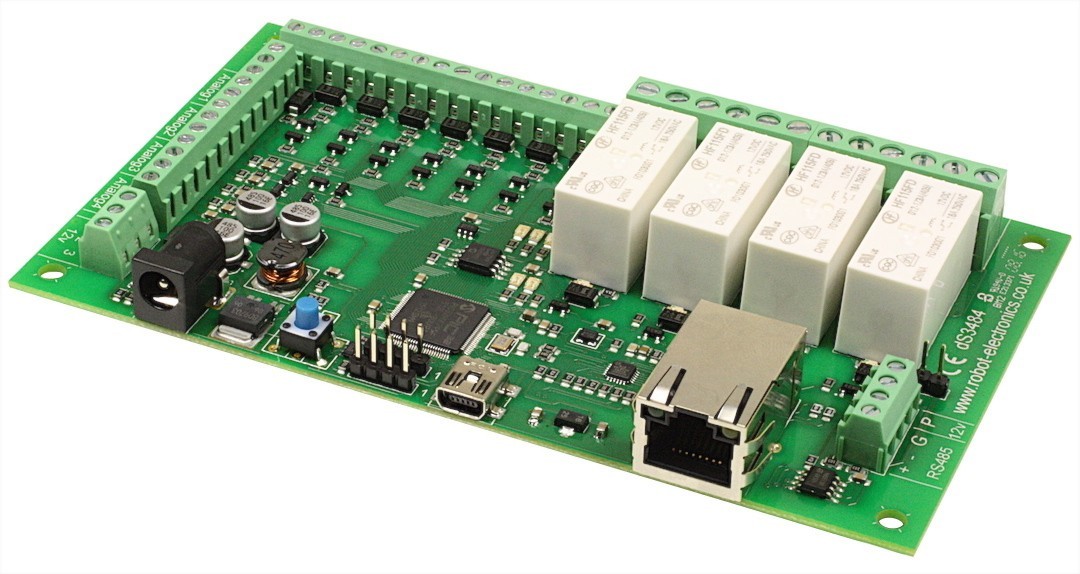
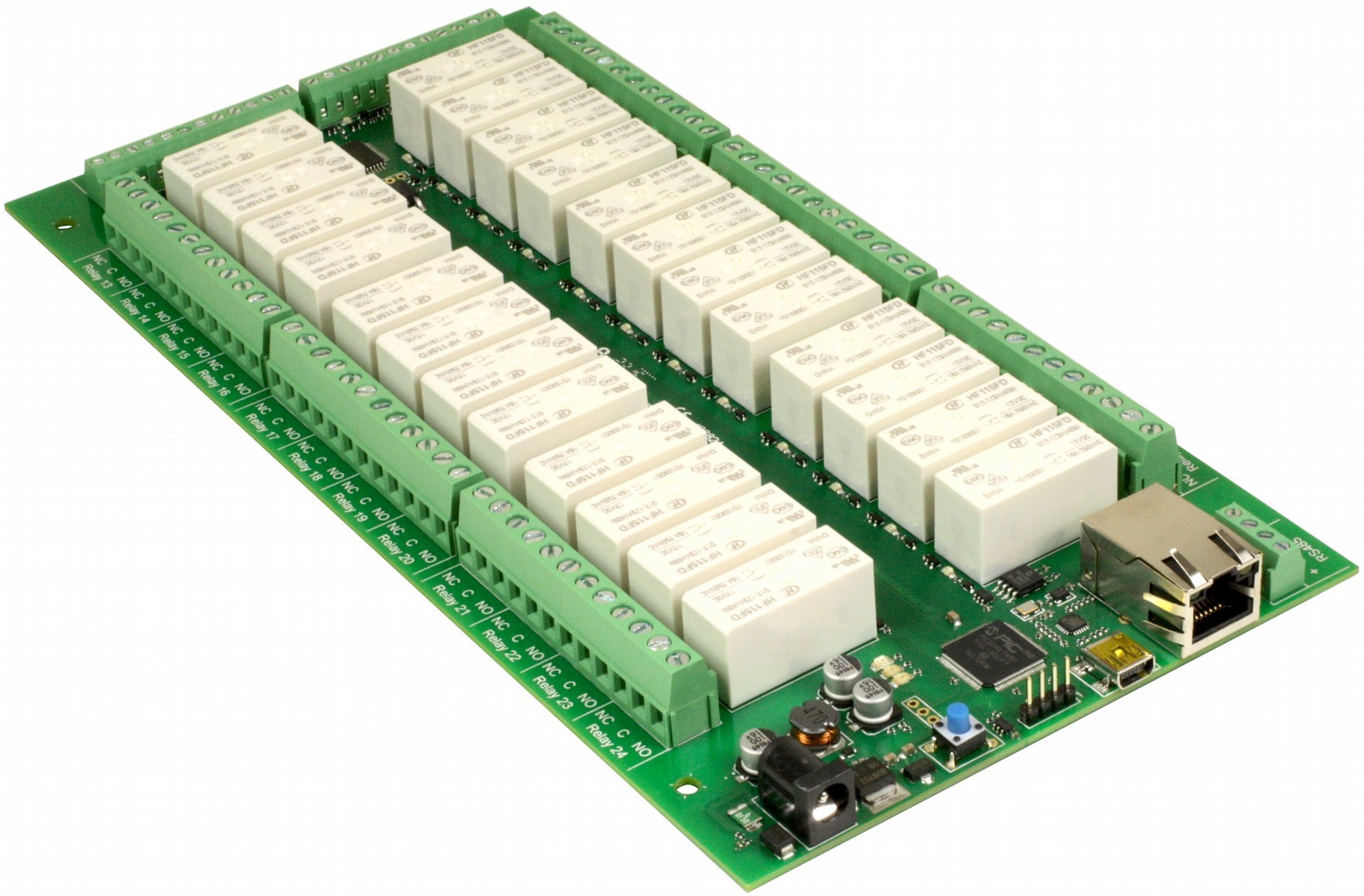
***room-control***

User Manual Version 1.3



# Documentation history

V1.3 First release of manual

Document is based on Devantec Ltd. documentation “dS2824 - v4.04.pdf”  
 It is for private / personal use by Martin Kraemer only!

# A quick look

This documentation covers different robot-electronic modules and a the custom application “room-control” which was developed to provide an easy possiblity to use dSXXXX modules for home automation purposes. While the application was tested only with dS2484 and dS3484 other dScript modules should work as well.

room-control application features:

Controlled graphically by secure webpage or optionally one of Binary or Binary AES command sets over TCP/IP.

Also available when optionally programming in dScript is a TTL level serial port and an RS485 serial port.

dS2484 module:  
Ethernet connected module, 10/100Mb auto negotiated. Relays – 24 x 16Amp 250Vac C/O.

I/O – 8 x flexible I/O's NPN output, Volt free input or 12-bit analogue input. Power – 12VDC 1.5Amp supply required. 2.1mm center positive.

Connections – Screw Terminals for N/O N/C and Common contacts PCB size – 249mm x 123mm

dS3484 module:  
Ethernet connected module, 10/100Mb auto negotiated. Relays – 4 x 16Amp 250Vac C/O.

I/O – 8 x digital I/O's NPN output, Volt free input.

4 x 10-bit analogue input

Power – 12VDC 1Amp supply required. 2.1mm center positive. Connections – ScrewTerminalsfor N/O N/C and Common contacts PCB size – 156mm x84mm

# Introduction

The room-control application provides three built in control methods.

Primary control method is graphically by using its built in secure website. Secondary control may be one of:

Binary - Command set using binary codes

BinaryAES - Command set using binary codes secured by AES key

And if you should wish to modify the supplied application – you can.

Behind the scenes there is dScript, a powerful multi-threaded operating system and programming language. The supplied firmware is written in dScript as well as HTML, CSS and Javascript on the webpages and the full source is in the separately down-loadable dScript support package.

You do not need to use dScript at all, but its great to know its there - just in case you do.

—————————  
The dS2824 is an Ethernet connected relay module featuring 24 channels of 16Amp 250Vac relays. Each relay has both normally open (NO) and normally closed (NC) as well as the common available on three terminals.

In addition to the relays, the dS2824 has 8 flexible I/O channels which may be individually configured to be:

1. NPN transistor output.
2. Active low input with pull-up (allows direct connection of volt freecontacts).
3. Active low input without pull ups (easy to connect logic levelinputs).
4. Analogue 12-bit inputs using a 4.096v reference (0v-4.096v inputrange).
5. Analogue 12-bit inputs using a 5v reference (0v-5v inputrange).

The dS2824 requires a 12v 1.5Amp power supply such as this one:<http://www.robot-electronics.co.uk/universal-12vdc.html>

—————————  
The dS3484 is an Ethernet connected relay module featuring 4 channels of 16Amp 250Vac relays. Each relay has both normally open (NO) and normally closed (NC) as well as the common available on three terminals.

In addition to the relays, the dS3484 has 8 digital I/O channels and 4 x 10-bit analogue input channels (0v-3.3v input range).

The dS3484 requires a 12v 1Amp power supply such as this one:<http://www.robot-electronics.co.uk/universal-12vdc.html>

# Getting started

Start by plugging in the Ethernet cable to connect the module to your network, and the 12v jack plug from your adapter.

As soon as you switch on might notice the red LED blinking 6 times (needing 3 seconds of time). If this happens it indicates the current application version was loaded the first time and all configuration values are set to their defaults (init() function is executed). If the red LED does not flash this is no error - it just means no new application version was loaded and the last state of configuration values is still active.

After this the blue LED will flash X times. This indicates the major version of the application.  
This is followed by the green LED flashing x times, which indicates the minor version of the application loaded.

Switch on and the first thing you will note is that the blue LED will flash 3 times. This indicates that the control firmware is loaded on the module. (*If the blue leddoes not flash you will need to load in the control firmware. Don't worry, this is very easy to do. Just go to the chapter on installing the firmware and follow the instructions there*).

The application firmware is now loaded and you can reach the application page using the devices IP address. To find the IP address either look at your dhcp-server/router or use nmap.  
An example nmap command looks like:

**nmap 192.168.0.0/24 -p 80,502,17123**

This provides an output of any device on the network and information if port 80 (default http), 502 (default modbus) and 17123 (default Binary/BinaryAES/ASCII) are open. A result looks like:

**Nmap scan report for 192.168.0.9**

**Host is up (0.086s latency).**

**PORT STATE SERVICE**

**80/tcp open http**

**502/tcp filtered mbap**

**17123/tcp open unknown**

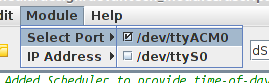
Using any PC open your browser and into the address bar (not the search bar) type:

[http://*192.168.0.9*/index.htm](http://dS2824/index.htm)

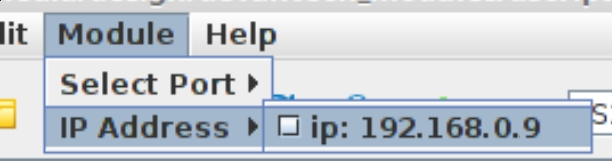
You should now see the application webpage and you can virtually use the inputs and control lights / shutters and others.

## Locating the IP Address

If you are not using a windows PC then you will need to find the IP address of the module. The simplest method is connect the module (Ethernet, USB thenlastly,Power) and load up the dScripteditor.



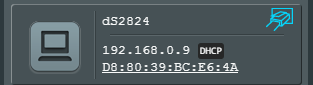
Go to Module→Select Port and make sure the correct serial port is selected as illustrated above.



Now go to Module→IP Address and the current IP address is displayed.

Make sure the Ethernet is connected before you apply power. If you plug the Ethernet cable in after the power, the module will already have booted with a default IP address.

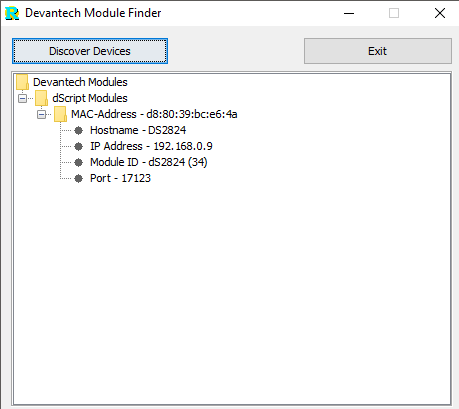
Alternatively, if you have a DHCP server on your network (your router is normally the DHCP server) then the module will get its IP address from that. Log on to your router and navigate to the LAN client list.



Above is the entry from an ASUS RT-N66U router. So now you type: 192.168.0.9/index.htm

into your browser address bar and you should see the application webpage shown on the previous page.jk

If you prefer, you can download a java program that will run on Windows, MAC or Linux, and will list all of our modules that are connected to your LAN.[DevantechModuleFinder.jar](http://www.robot-electronics.co.uk/files/DevantechModuleFinder.jar)



If you do not have a DHCP server the module will use a default IP address of 192.168.0.123 so make sure your PC is on the same subnet of 255.255.255.0 and its IP address is 192.168.0.xxx

# Configuring the module / application

There are a set of configuration pages to get the module operating as you want it. These pages are all \_configx.htm, (that's a leading underscore character).

ie.

\_config.htm

\_config2.htm

Anything that starts with \_config is considered a special name for configuration pages and can only be seen if you have the the USB cable plugged in and connected to your PC.

Why only if the USB cable is plugged in?

Its an additional security measure. After you have configured and deployed the module, you really don't want these configuration pages available for others to change. So with the USB cable disconnected the \_config pages are not available. If you try to access them you just get served a “not authorised” page instead.

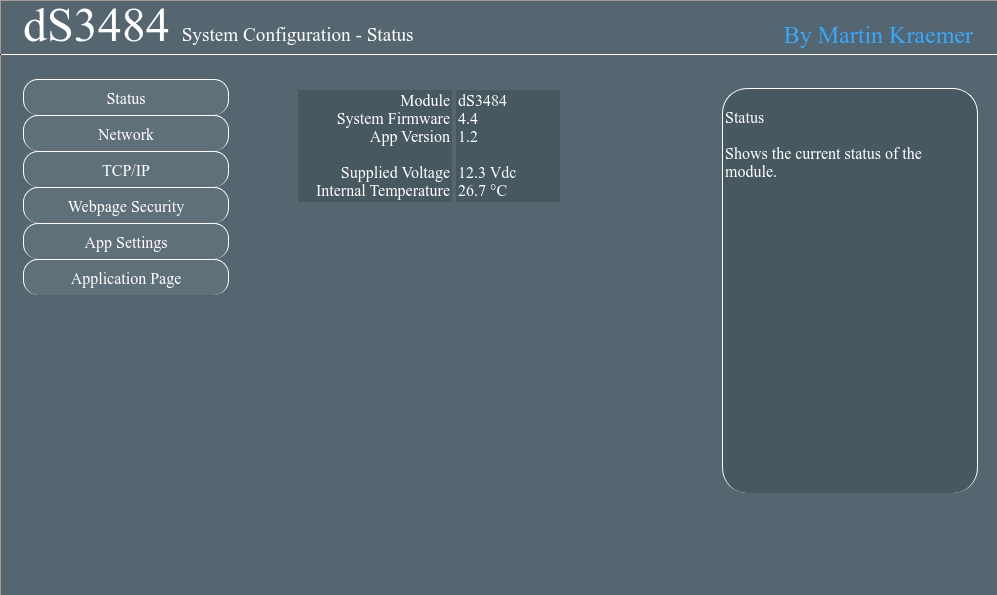
So for now, you do want to look over the config pages. If you have a Win10, Linux or MAC PC, you can go right ahead and plug in the USB cable. These machines will install their own USB drivers. We won't be sending anything to the board, its just the presence of the USB connection that enables the config pages to be served. If you have a Win7 or Win8 PC you will need to install the drivers. (*go to the chapter on installing the firmware and follow theinstructions there*).

With the USB cable connected, browse to: 192.168.0.9/\_config.htm

(substituting your IP address)

## Status page

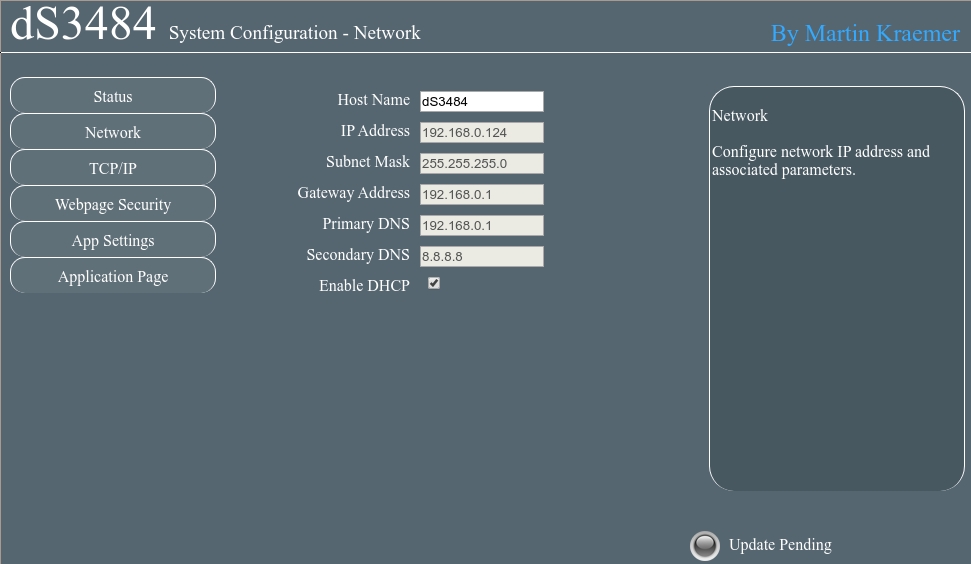
You should now see the following page:



This status page shows you the system, module and application firmware revisions as well as the supplied voltage to the board and its internal temperature.

If you hover your mouse cursor over the menu buttons on the left, the help panel will give you an overview of each one.

## Network page



Notice that everything below the Host Name is greyed out and can't be changed. This is because the “Enable DHCP” box is checked and all the greyed out fields are supplied by the DHCP server. Although a quick way to get you connected, we really do not recommend this as the DHCP server can assign a different IP each time you power up. If you want to control this module from the internet while you are away from the premises then you will be setting up port forwarding on your router which requires a fixed IP address.

So lets do that first.

Uncheck the DHCP box and you can then set all the other fields. Notice that the Red “Update Pending” light comes on. It indicates there are changes which have not yet been written to the flash memory. It will go off again 5 seconds after you stop changing anything.

Choose an IP address for the module, something outside of the DHCP settings on your router so it will not assign anything to that address.

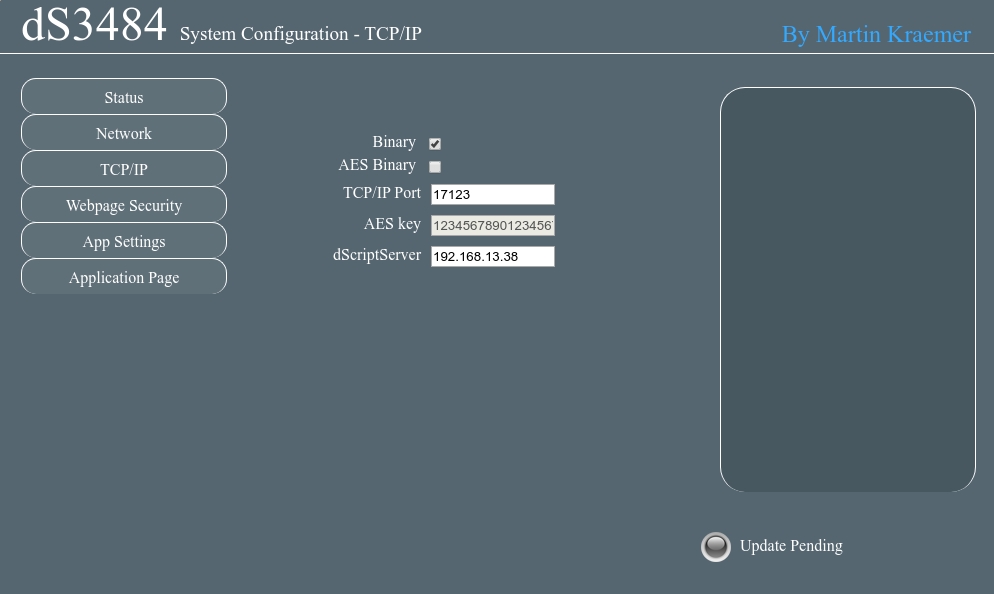
The subnet mask, Gateway and DNS can all be left as the defaults.

Network changes only take effect after the next re-boot, so wait until the “Update Pending” light goes out and give the reset button on the module a quick press. The Green LED will light and the start-up LED flashing is shown. You will now find the module at your new IP address. If the Red led comes on after you press the reset button, its because you pressed it for too long (and entered bootloader mode). Just have another go with the reset button for a bit less time.

Your browser won't know you have changed the IP address so it will still be showing the old, now dead page. Make sure you change to your new IP address and load the page again.

## TCP/IP page

The TCP/IP tab allows you to select one of three command sets to control the module.These are independentof,and separate to the HTML webpagecontrol.



Clicking on one of the four check boxes will select that command set. Only one commandset may be selected.You can disable all TCP/IP command sets by clicking on an already selected box.

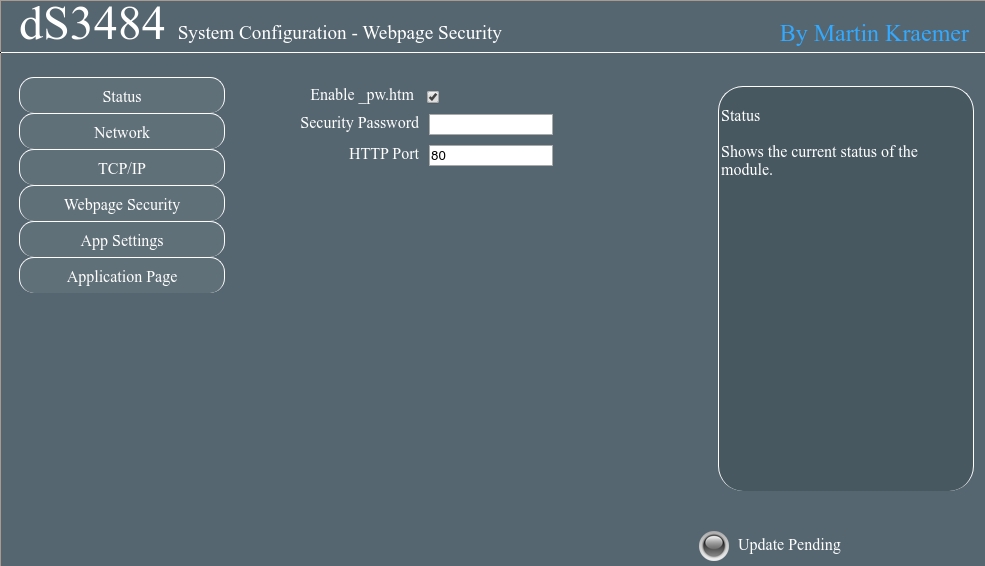
Currently only Binary and AES Binary mode are available. In future ASCII and Modbus might come (maybe MQTT, too). AES Binary adds encryption. Note the AES key MUST be 32 bytes long. A full description of the commands is in the “Command Sets” section later in this manual.

The TCP/IP Port is the listening port. Default here is 17123 - for modbus never the less you should use the officially reserved port 502.

The “dScriptServer” is a server provided via python modul. This server can receive consolidated information about multiple dScirpt modules to allow integration e.g. into home automation systems (read more about this within dScriptServer

## Webpage security

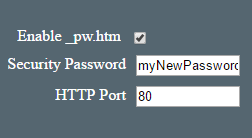
Allows you to prevent unauthorized personnel from accessing the application web page or using it to control the module.



Leaving the Security Password blank will disable it and allow everyone to access the application page to control the module.

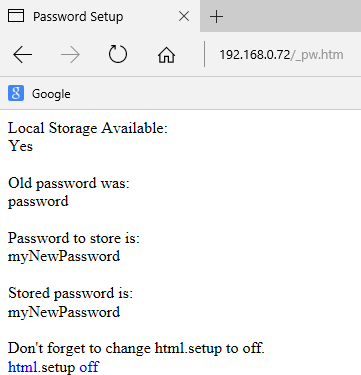
To enable password protection enter a password into the Password box. You can use any characters from the ASCII character set from 0x20 to 0x7E except “ (0x22). It may be up to 200 characters long and you don't have to memorize it, so make it a long one with plenty of uppercase, lowercase, numbers and symbols.

Accessing a special web page, \_pw.htm, will install the password on your browser. To do this make sure the Enable \_pw.htm box is checked.



When the “Update Pending” light goes out, re-boot the module and go to yourIP/\_pw.htm

You will see something like this:



The password is now loaded on your browser. Do the same for any further browsers you want to enable. When you have done unchecked the “Enable \_pw.htm” box to prevent anyone else from loading the password. When the “Update Pending” light goes out re-boot the module again.

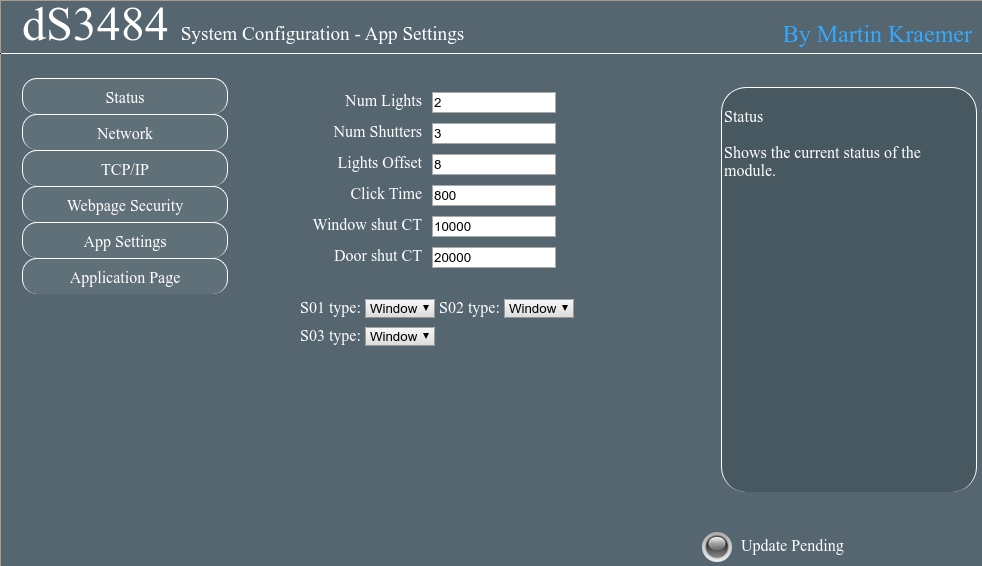
The default port used by html web pages is 80. You can change this if required. If you do so then you will need to include the port number in the address.

If you change the port to 2345 then the webpage will be at: <YourIP>:2345/index.htm

For example: 192.168.0.9:2345/index.htm

## App Settings

This tab allows you to configure the provided room-control application and is not part of the default devantec configuration pages. It is reachable via <YourIP>/\_config99.htm.



The “Click Time” input box defines the number of milliseconds (seconds \* 1000) you have to perform a multi-click input. So if you define a value of e.g. 500 this will give you 0.5 seconds to click multiple times on the same input. The usual value you should calculate for a click are between 200-250 milliseconds. Please note that with increasing this value the delay between click & action will increase, too.

Within the “Num Lights” input you can define the number of lights connected to the board. Each light needs exactly one relay so a maximum of 32 lights is possible (using all virtual & physical relays).

“Num Shutters” input defines the number of shutters connected to the board. One shutter needs two relays so a maximum of 16 shutters is possible (using all virtual & physical relays). According to the number of shutters you have defined at the very bottom of the page you will see a different number of drop down boxes named “SXX type:” which allow you to define the type of of any shutter individually.

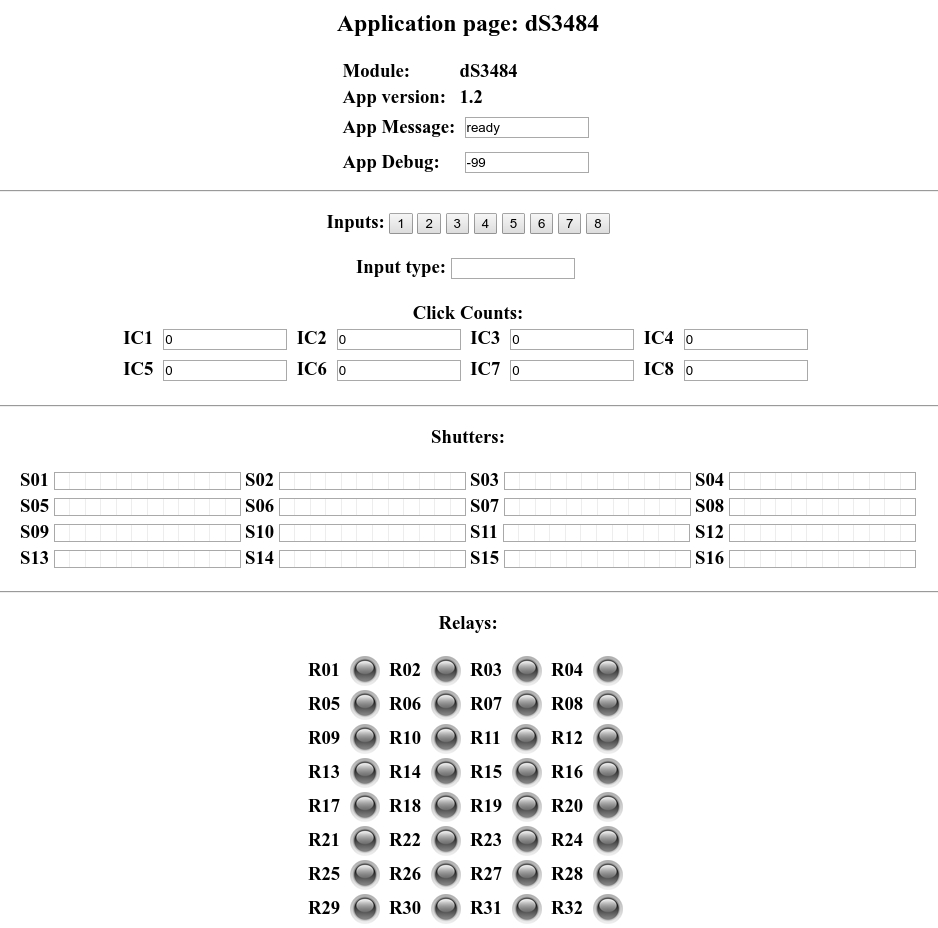
Coming back to the number (Num) of lights and shutters you of course cannot use the same relay for a shutter and a light at the same time. This is why the sum of (Shutters \* 2) + Lights needs to be smaller or equal to the 32 relays available.

The input “Lights Offset” defines at which relay lights start to be connected. To get the idea behind this configuration value you first need to know that shutters always start at relay one. This means if you have e.g. two shutters defined they will use the first four relays (two relays per shutter). If we now continue directly with lights, we do not have the possibility to easily (without reconfiguring the app and with this our input buttons) add new lights or shutters afterwards. This is why the relay id for our first light starts at the highest relay for lights (defined by “Lights Offeset” input) and counts backwards. This means if you e.g. have two lights defined and a lights offset of four, the first light will start at relay number four and the second light at relay number three. Please additionally note that in this example case you can connect a maximum of one shutter since you have two relays left at the beginning.

The values of “Window shut CT” and “Door shut CT” define the closing time (CT) a shutter needs to move from fully open to fully closed or the other way round. To assign the correct values to your shutter based on the fact that they are used on doors or windows, use the according drop down available below. (A additional closing times of “OtherX shut CT” might come in the future to define additional closing times for garage doors etc.)

# The application page

The last tab in the configuration pages takes you directly to the application page so you can quickly see the results of your configuration changes.



## Application page security

The configuration pages (any page name starting with \_config) are only served when the USB port is connected. If the USB port is unplugged then no configuration pages are served.

Instead, you will be served a page saying “You do not have permission to view this page.”

The \_pw.htm page (which contains the javascript that loads the password into your browser) will only be served when the enable \_pw.htm checkbox is checked on the Webpage Security tab.

If the password field is left blank, the application page will always be served – to anyone! Entering a password means the application page will only be served to a browser that has the matching password set. Everyone else will just just be served a simple webpage saying “You do not have permission to view this page.”

To summarize, set a password, load it onto your browser, disable \_pw.htm and un-plug the USB cable. Now you, and only you, can access and control your module.

## Accessing your webpage from the internet

Now you have your webpage up and running on your local network, for example 192.168.0.9, and you can access the webpage and control the module.

You just go to 192.168.0.9/index.htm, and the page is there.

However you can't get directly at that page from your phone when you are away from home. You can't access it on 192.168.0.9 because your network is not publicly accessible, its a private network address. You will have another IP address. This is the one your ISP gave you for your internet connection, and is the public IP address of your router on the internet. If you don't know what it is you can type “my ip” into Google's search bar and it will tell you. This is the IP address you will use to access the modules webpage.

Everything on the internet uses an IP address and a port number.

When you access a webpage in your browser all you enter is the IP address (or more likely a domain name, but its ultimately translated to an IP address). You don't normally have to enter a port number but its still required. Your browser simply uses the default port number, which is 80 for the web, unless otherwise specified our modules also use port 80 for the webpage.

However its a good idea to use a different port number for our boards as this will avoid conflict with any web server you may have on your network.

Pick a number, I'll choose 19321 as our port number. Just make sure its different from any TCP/IP port number you are using. The HTML port is set on the Webpage Security tab.

After you have re-loaded the program you can access the webpage with: 192.168.0.9:19321/index.htm

Note that as we have changed the modules html port we need to tell the browser how to find the page with the new port number. Do that by inserting a ':' character and the port number between the IP address and the page name as shown above.

Assuming your routers internet IP address is 86.87.88.89 (I made that up – replace with your actual IP address) you will access the page from anywhere with address: 86.87.88.89:19321/index.htm

However first you have to set up your router to do that.

It's called “port forwarding” or “virtual server”,but whatever your router calls it, you need to set it up so that all incoming connections on port 19321 are forwarded to port 19321 on local IP address 192.168.0.9.

Unfortunately there are so many routers out there we cannot give details on all of them.You should consult your routers manual or search Google for details on your specific router.

# TCP/IP command sets

There are two TCP/IP command sets on two selectable check boxes, of which one or none may be selected on the TCP/IP config tab. These are Binary and Binary with AES256 encryption (ASCII, Modbus and MQTT might come in future).

## Binary command set

0x30 Get Status 0x31 Set relay 0x32 Set output 0x33 Get Relays 0x34 Get Inputs 0x35 Get Analogue

0x36 Get Counters

0x40 Set Light

0x41 Set Shutter

#### 0x30 (byte 1 = decimal 48) Get Status (1 byte command, returning 8 bytes)

This command returns 8 bytes of status data

ModuleID This will be 34 (0x22) for the dS2824

SystemFirmwareMajor 2 for example

SystemFirmwareMinor 18 for example

ApplicationFirmwareMajor 1 for example ApplicationFirmwareMinor 2 for example

Volts (Power supply volts x 10.) Example 125 is 12.5v

InternalTemperature(highbyte) x10

InternalTemperature(lowbyte) combined to 16 bits, 267 = 26.7 °C

In the above example the returned bytes would be:

0x22 0x02 0x12 0x01 0x02 0x7D 0x01 0x0B

The bytes 7 (0x01) & 8 (0x0B) combined are 0x10B which is 267 decimal, meaning 26.7 °C

#### 0x31 0x02 0x01 0x00 0x00 0x00 0x00 (byte 1 = decimal 49) Set Relay (7 byte command, returning 1 byte)

This command turns a relay on or off or pulses it for a time period and returns an ACK/NACK byte. ACK=0, NACK=non-zero (actually the unknown relay number).

0x31 The actual command, the rest are parameters. 0x02 Relay number Validnumbers are 1-24(0x01-0x18)

0x01 Turn relay on (0x00 for off). This is ignored when following pulse time is >100. 0x00 }highbyte Pulsetime (this parameter is not supported and always set to 0)

0x00 }midhigh These 4 bytes combined are a 32-bit pulse time for therelay 0x00 }mid low when less than 100 (as it is here 0x00000000) its ignored 0x00 }low byte When >100 this pulses relay on for that number ofmS

To pulse relay 5 on for one second the command is:

0x31 0x05 0x00 0x00 0x00 0x03 0xE8

0x000003E8 (or just 0x3E8) is 1000 decimal, which is 1000mS or 1 second. The relay will turn on and then go off 1 second later.

When sending a relay pulse time, the relay on/off byte is ignored. The relay is always on for the duration of the pulse. Never the less the pulse time parameter is no longer support by this application and will be ignored if set to different than 0.

#### 0x32 0x04 0x01 (byte 1 = decimal 50) Set Output (3 byte command, returning 1 byte)

This command turns an output on or off and returns an ACK/NACK byte. ACK=0, NACK=non- zero (actually the unknown I/O port number).

All I/O's which need to be inputs should have the output turned off. When turned on the NPN transistor can sink up to 100mA.

0x32 The command, the rest are parameters 0x04 the I/O number,4 in this case.

0x01 on (0x01) or off(0x00)

#### 0x33 0x01 (byte 1 = decimal 51) Get Relay (2 byte command – returning 5 bytes)

This command is used to get the states of the relays. The second byte is the relay number, relay 1 in this case.

The first returned byte is the state of the requested relay, 0x00 (off) or 0x01 (on).

The next four bytes pack the states of all 32 relays (virtual and actual relays). Bit 7 of byte 2 is relay 32 through to bit 0 of byte 5 which is relay 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 2 | | | | | | | |  | Byte 3 | | | | | | | |  | Byte 4 | | | | | | | |  | Byte 5 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

If the bit is set the relay is on, off otherwise.

#### 0x34 0x01 (byte 1 = decimal 52) Get Input (2 byte command – returning 2 bytes)

This command is used to get the states of the inputs. The second byte is the input number, input 1 in this case.

The first returned byte is the state of the requested input, 0x00 (inactive) or 0x01 (active) the second byte packs the states of all 8 inputs. Bit 7 is input 8 through to bit 0 which isinput

1. If the bit is high the input isactive.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Byte 2 | | | | | | | |
| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Input | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

#### 0x35 (byte 1 = decimal 53)

**Get Analogue inputs (1 byte command returning 16 bytes)**

This returns all eight possible analogue inputs. 16 bytes are returned, 2 for each analogue input. Inputs bytes that do not exist for corresponding module are filled with 0.

Byte1 byte2 for example:

0x02 0x3E combined to 0x023E, or 574 decimal for input1.

Byte3 byte4

0x01 0xFB combined to 0x01FB, or 507 decimal for input 2. Bytes 5 – 14 follow in a similar way.

Byte 15 byte 16

0x03 0x2C combined to 0x032C, or 812 decimal for input 8. If the input is configured as a digital port then the input will be 0 or 1.

#### 0x36 0x01 (byte 1 = decimal 54)

#### Get Counters (2 byte command – returning 8 bytes)

This command is used only for backwards compatibility. It was used to get counters.

The second byte is the counter number, counter 1 in this case.

The first 4 bytes returned is the current counter value. This 32-bit (4 bytes) value is returned high byte first. The second group of 4 bytes returned is the capture register for this counter, also a 32-bit (4 byte) value returned high byte first.

Now the command will return always 0 since no counters in that way exist anymore.

#### 0x40 0x02 0x01 (byte 1 = decimal 64) Set Light (3 byte command, returning 1 byte)

This command turns a light on, off or toggles it and returns an ACK/NACK byte.

ACK=0, NACK=non- zero (actually the unknown light number).

0x40 The command, the rest are parameters 0x02 the light number - 2 in this case.

0x01 on (0x01) or off (0x00), any other value will toggle

#### 0x41 0x03 0x32 (byte 1 = decimal 65) Set Shutter (3 byte command, returning 1 byte)

This command opens/closes or sets a shutter to a specific state and returns an ACK/NACK byte.

ACK=0, NACK=non- zero (actually the unknown light number).

0x40 The command, the rest are parameters 0x03 the shutter number - 3 in this case.

0x32 closed/open state of the shutter (50% in this case ) - fully open (0x64), fully closed (0x00), any other value (between 0 and 100) setting directly the closing state

#### 0x42 0x01 0x01 (byte 1 = decimal 66) Set Socket (3 byte command, returning 1 byte)

This command turns a light on, off or toggles it and returns an ACK/NACK byte.

ACK=0, NACK=non- zero (actually the unknown light number).

0x40 The command, the rest are parameters 0x01 the socket number - 1 in this case.

0x01 on (0x01) or off (0x00), any other value will toggle

#### 0x50 (byte 1 = decimal 80) Get Config (1 byte command, returning 8 bytes)

This command returns 8 bytes of configuration data

ModuleID This will be 34 (0x22) for the dS2824

TCP/IP Port (highbyte) 0x42 (local tcp/ip server)

TCP/IP Port (lowbyte) combined to 16 bits, 0x42 + 0xE3 = 0x42E3 = 17123

TCP/IP “Protocol” 3 ( 1=Modbus; 2=ASCII; 3=Binary ; 4=BinaryAES )

Count of pysical relays 8 for example

Count of lights 2 for example

Count of shutters 1 for example

Count of power sockets 5 for example

In the above example the returned bytes would be:

0x22 0x42 0x42 0x03 0x08 0x02 0x01 0x05

#### 0x51 0x03 (byte 1 = decimal 81) Get Light (2 byte command – returning 1 byte)

This command is used to get the state of a light. The second byte is the light id, Light 3 in this case.

The returning byte can be either 0 (= off) or 1 (=on).

#### 0x52 0x01 (byte 1 = decimal 82) Get Shutter (2 byte command – returning 2 bytes)

This command is used to get the states of a shutter. The second byte is the shutter id, Shutter 1 in this case.

The first returned byte is the state of the requested shutter in percentage, values from 0x00 (0=Fully closed) to 0x64 (100=Fully open) are possible. The second byte returns the current movement of the shutter 0x00 (0=stopped), 0x01 (1=opening) and 0x02 (2=closing).

#### 0x53 0x07 (byte 1 = decimal 83) Get Socket (2 byte command – returning 1 byte)

This command is used to get the state of a socket. The second byte is the socket id, socket 7 in this case.

The returning byte can be either 0 (= off) or 1 (=on).   
(We assume that sockets are connected via NC instead of NO to relays. Based on this Sockets are inverted with their state. That means if the relay is on, the socket is off. If the relay is off, the socket is on)

## AES binary command set

The AES Binary commands are the same as the Binary commands described above. The only difference is that they are AES encrypted and always 16 bytes in length. The first bytes are the same as described in the Binary command set. The last 4 bytes is the Nonce (a random number) and the bytes in the middle are undefined. The module will decrypt the command, generate the response and encrypt it before returning it to you.

Your program that controls the module will need to encrypt the commands and then decrypt the response. We use AES256 CBC encryption, hence the requirement for a 256 bit (or 32 byte) key. To complete the security we use a random IV generated by a cryptographically secure random number generator (ISAAC).

To control the module you will need to send the commands with AES encryption. To help you with this we have examples in C#, Java and Python. The C# and Java applications are complete and may be used or modified as you wish.

To prevent re-play (or Playback) attacks the command packet includes a Nonce. This takes the form of a 32-bit (4 byte) random number in positions 12, 13, 14 & 15 of the 16 byte data packet. For example when you send a Get Status command (0x30) you will get a 16 byte block returned. The first 8 bytes (0-7) will be as defined for the binary commands. Bytes 8-11 are unused. Bytes 12-15 contain the Nounce.

Commands with generate and send you a Nounce are:

0x30 – Get Status 0x31 – Set Relay 0x32 – Set Output

Commands which require a Nounce to be sent by you are:

0x31 – Set Relay 0x32 – Set Output

A Nounce is only ever used once, you must always used the most recently issued Nounce.

The following example shows how the Nounce provided by the module is used in the next Set Relay or Set Ouput command.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Get Status | 0x30 | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx |
| Response | 0x22 | 0x02 | 0x12 | 0x01 | 0x02 | 0x7D | 0x01 | 0x0B | xx | xx | xx | xx | 0x89 | 0xAB | 0xCD | 0xEF |
| Set Relay | 0x31 | 0x02 | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 | xx | xx | xx | xx | xx | 0x89 | 0xAB | 0xCD | 0xEF |
| Response | 0x00 | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | 0x1A | 0x2B | 0x3C | 0x4D |
| Set Relay | 0x31 | 0x03 | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 | xx | xx | xx | xx | xx | 0x1A | 0x2B | 0x3C | 0x4D |
| Response | 0x00 | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | 0xF1 | 0xE2 | 0xD3 | 0xC4 |
| Set Output | 0x32 | 0x04 | 0x01 | xx | xx | xx | xx | xx | xx | xx | xx | xx | 0xF1 | 0xE2 | 0xD3 | 0xC4 |
| Response | 0x00 | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | 0x5C | 0x47 | 0x9B | 0xED |

If the Nounce you send with the command does not match the last one sent to you, then the Relay (or Output) will not be changed.

No other commands either require or provide a Nounce.

# The dScriptServer

BLABLABALBABLA <TO-DO>

# Loading the application firmware

If your module does not perform the initial LED flashing which shows the application version on power-up, you will need to update the system firmware and load the application program. These instructions should also be followed if you want to update your firmware to the latest version or even revert to an earlier version.

You need to download the dScript programming environment from here:<http://www.robot-electronics.co.uk/dscript.html>

#### PC requirements

Windows 7 or later, Linux or MAC OS X USB port to program the module.

The dScript IDE is supplied as a zip file that can be download and unzipped into a temporary folder, inside the temporary folder will be four folders:

Installation USBdriver Examples Documentation

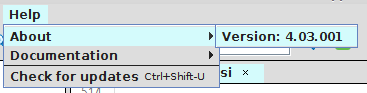
GototheInstallationdirectoryandclick“setup”toinstallthedScriptIDE,ifyouhavealready installed a previous version you will need to uninstall it before installing the newone.

The USBdriver folder contains the USB com port driver for the modules.

Copy the Examples directory to a convenient location on yourcomputer,it contains both dScript source code examples and associated web pages, one of which is the application firmware you will need.

In this order:

* 1. Start from thisposition:
     1. dScript Editor closed down.
     2. dS2824 not connected orpowered.
  2. Power-up thedS2824.
  3. Hold down the reset button for a couple of seconds until the red LED comes on.This indicates the module is in boot-loadermode.
  4. Connect the USB lead to the PC. If windows wants adriver,point it to the USB driver folder and install the driver fromthere.
  5. Run the dScripteditor.Look in Help→About and check you have the latest version of dScript. In this case4.03



If you have an earlier version you should uninstall it and install the new version from the installation folder, then start these instructions from the beginning. Older versions will not work with the new application.

* 1. Now look in the Module panel, you shouldsee:



v1.01 indicates it’s the boot-loader that is running.

* 1. Load the project: File→Openproject

\dScriptPublish-4-03\Examples\app-dS2824-v4-03\app-dS2824-v4-03.dsj

* 1. Click the build button (white triangle on green button). This will update thesystem firmware and load theapplication.

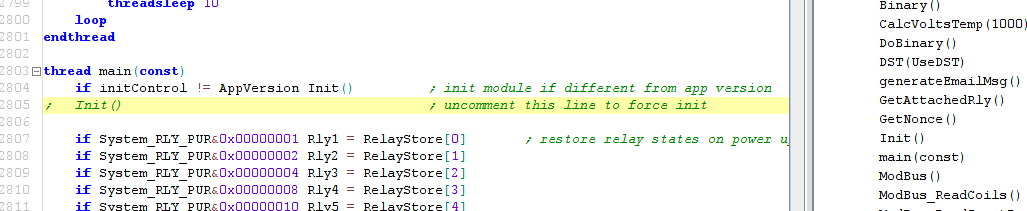


* 1. When done, the new version will bedisplayed:

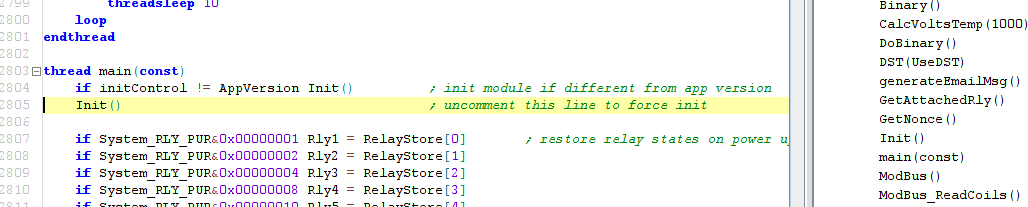
## Erasing old configuration settings

Uploading the firmware, as described in the previous section, will not erase all the configuration values. If you need to clear these, do the following:

Load up the app-dS2824-v4-03 application in the editor, but before you upload it to the dS2824 you need to make a small change. Locate the thread "main" (click the word in the right panel is quickest). Just below this is a commented out line "init()".



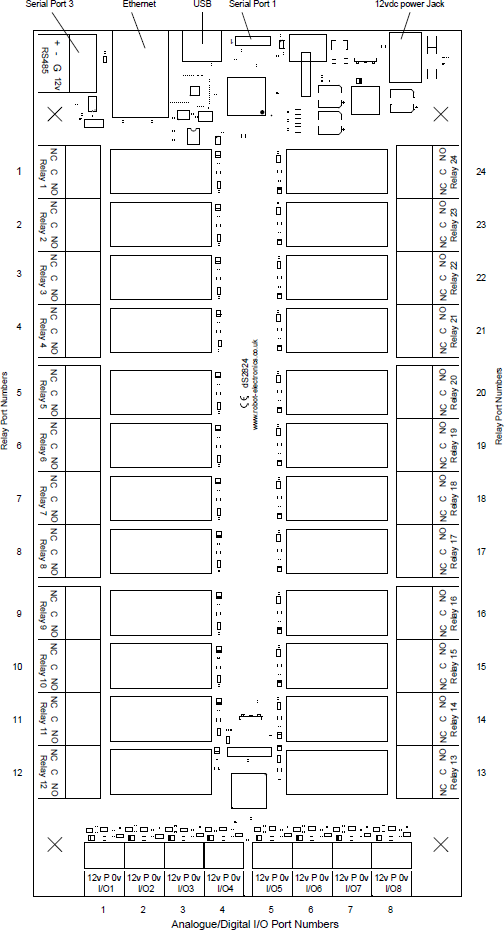
Uncomment this by removing the semicolon.



Now upload the application and it will reset the IP address along with all other variables. Confirm the board is operating, but don't change anything yet.

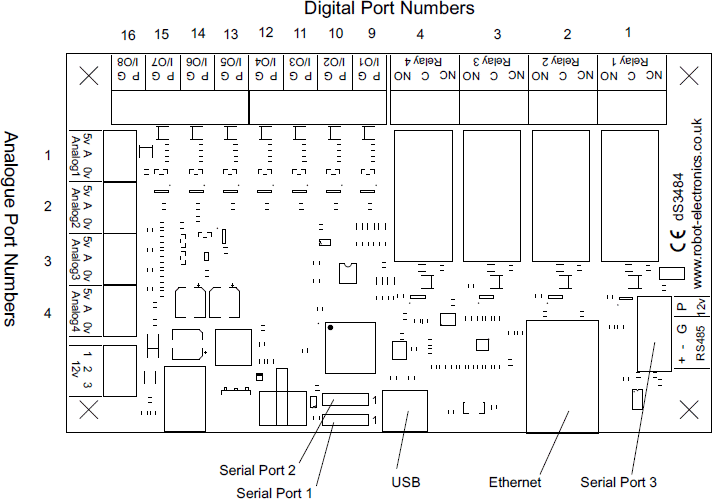
You need to replace the semicolon and upload the application again, otherwise it will continue to reset everything each time you power-up.

# dS2824 hardware



The dS2824 provides twentyfour (24) volt free contact relay outputs with a current rating ofup to 16Amp each, 8 flexible I/O's which can be analogue or digital and 1 serial port (3.3v level). The module is powered from a 12vdc supply which can be regulated or unregulated. The relays are SPCO (Single Pole Change Over) types. The normally open, normally closed and common pins are all available on the screw terminals.

# dS3484 hardware



The dS3484 provides four (4) volt free contact relay outputs with a current rating of up to 16Amp each, 8 digital I/O's which can be an NPN transistor output or accept a volt free input (from relay or switch contacts, etc). 4 analogue inputs (0-3.3v level), 2 serial ports (3.3v level) and an RS485 serial port. The module is powered from a 12vdc supply which can be regulated or unregulated. The relays are SPCO (Single Pole Change Over) types.The normally open, normally closed and common pins are all available on the screw terminals.

## LED indication

The modules provide a red LED mounted immediately next to each relay to indicate whether it is in a powered state (LED on), there are also two LED's mounted in the Ethernet connector which will flash with Ethernet traffic. A row of three LEDs, Blue, Green and Red, are available for status indication. The Red LED lights when the module is in bootloader mode – this is when the IDE is uploading system firmware to the module. The Green LED lights the board begins running user programs. All three LEDs are available and can be controlled as digitalports 33- 35.

digitalport LedBlue 33

digitalport LedGreen 34

digitalport LedRed 35

The “FlashingLeds” example provides a colourful display showing how to use them.

There are also LEDs on the I/O lines. These indicate the I/O status in digital output and VFC input modes only.

## Power supply

The modules require a 12v DC supply capable of supplying a minimum of 1.5A. This is most easily provided by a low cost mains adapter. A suitable universal adapter is available on our website and may be ordered along with the modules. Connection is via the 2.1mm DC jack socket. Positive on the center pin.

## Operating temperature

-40C to +70C

## Power relays

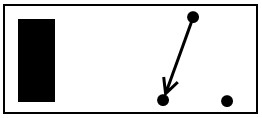
Four 16A volt free contact relays are provided for switching a common input between a normally closed output and a normally open output. The relay coil is powered by the 12vdc incoming supply on user command.

Coil

C

Relay in passive state

NC NO

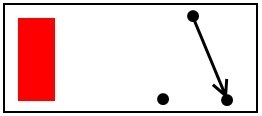


Coil

C

Relay in powered state

NC NO



A full datasheet for the relays used on the dS2824 is here:[HF115FD datasheet](http://www.robot-electronics.co.uk/files/HF115FD.pdf)  
A full datasheet for the relays used on the dS3484 is here:[HF115FD datasheet](http://www.robot-electronics.co.uk/files/HF115FD.pdf)

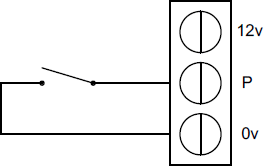
## Analogue/Digital flexible I/O's

The **dS2824** has eight flexible I/O ports, numbered 1-8, which can be your selection of:

1. Digital open collectoroutput.
2. Digital open collector output with passive pullup to12v
3. Digital input(0-12v,2.5v threshold).
4. VFC(VoltFree Contact)input.
5. 12-bit Analogue input, range 0-4.096v or0-5v.

# Connection examples

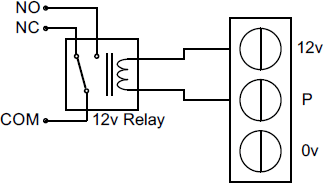
#### Example input - connecting a switch



Connecting a simple switch could not be easier, just wire the switch between a pin (P) and ground (0v). When the switch closes the input will become active.

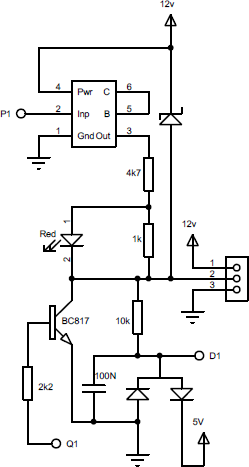
Make sure you use the 1 option in the digitalport declaration to enable the ports passive pull-up

#### Example output - connect a relay



Youcan connect your own 12V relays, 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 (NormallyOpen).

#### Representative flexible I/O schematic



P1 controls the passive pull-up. It is automatically disabled in analogue modes and set by the option in the digitalport declaration in digital modes.

Q1 controls the open collector NPN output. It is set by writing to the port.

D1 is the ports input. It supplies the 12bit ADC in analogue modes and is read as 0 or 1 in digital modes.

## Digital IO

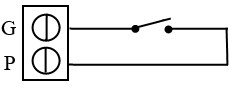
Our Ethernet modules could potentially have many types of outputs. For example the ETH008 only has one type - Relays. The **dS3484** has both Relay outputs and NPN Open Collector Transistor outputs. Activating a relay means turning the relay on. Likewise activating an output means turning the transistor 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 outputs will sink current (up to 100mA max.) when active.

The same principle applies to the 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.

## Connection Examples

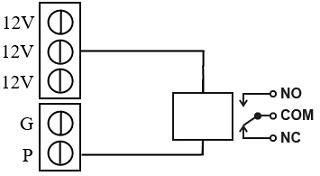
### Example input - connecting a switch



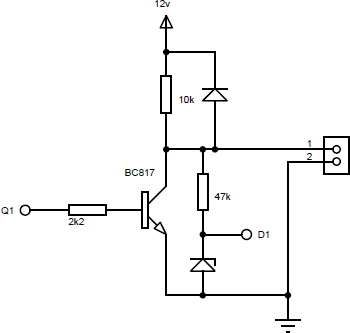
Connecting a simple switch could not be easier, just wire the switch between a pin (P) and ground (G).

When the switch closes the input will become active.

### Example output - connect a relay



Youcan connect your own 12V relays, 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.



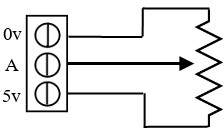
### Representative Digital I/O Schematic

**Analogue inputs**

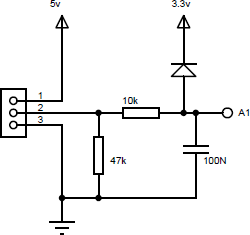
Four independent analogue input channels are provided for sampling voltages up to 3.3V. Each channel is also filtered with a 10k resistor and a 100n capacitor to stabilise high frequency jitter, there is also a pull down resistor so the port will read around 0 when nothing is connected. 5V inputs can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

### Examples

### Example input - connecting a potentiometer



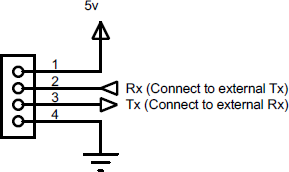
Connecting a potentiometer should be the simplest of tasks, either end of the pot should be wired to the 5v and 0v respectively, the output pin of the pot is then wired into the analogue (A) pin. Please note the reading from the conversion will reach maximum at 3.3V.



### Representative Analogue Schematic

## Serial port connections

TTL serial port.

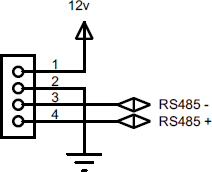


The modules have one serial port header which is serial port 1 in dScript.

Pin 1 is marked on the PCB.

Tx and Rx operates at 3.3v levels, however the Rx input is 5v tolerant. Most 5v TTL level serial ports will accept a full 3.3v input ensuring the port is compatible with most 5v devices such as the LCD05 display modules. That is the reason 5v (rather than 3.3v) is available on pin1 to power external modules.

RS485 serial port.



Referred to as serial port 3 in dScript.

The 12v power is not part of the RS485 specification, it is there to provide convenient power for any 12v sensors you may wish to connect such as our SRF485 ultrasonic rangers.

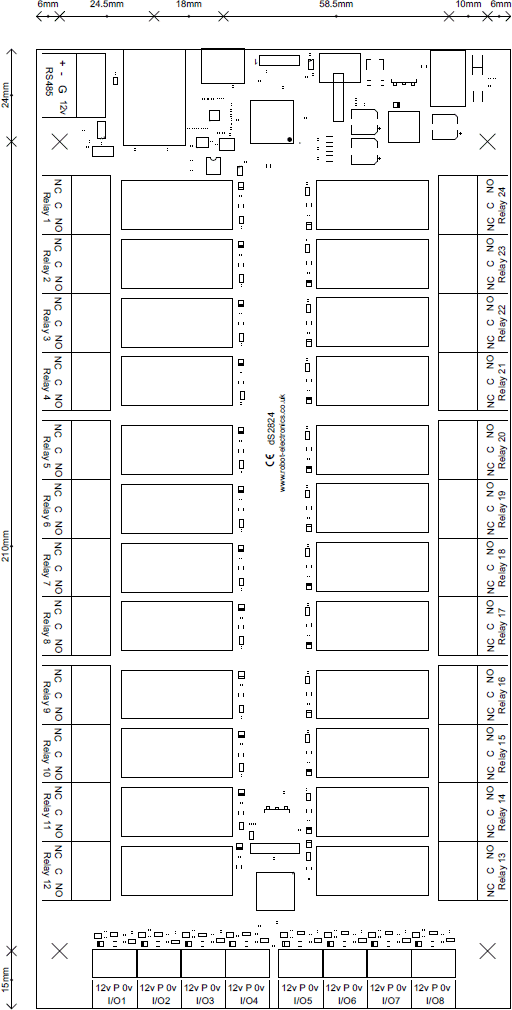
Connections for the RS485 port are also clearly labeled on the PCB.

The two pin link near to the RS485 terminal block should be shorted to use the on-board 120 ohm terminating resistor.

Notes.

1. Serial port 2 is not available on the dS2824.
2. Serial ports are available when programming in dScript

# dS2824 dimensions



# dS3484 dimensions

