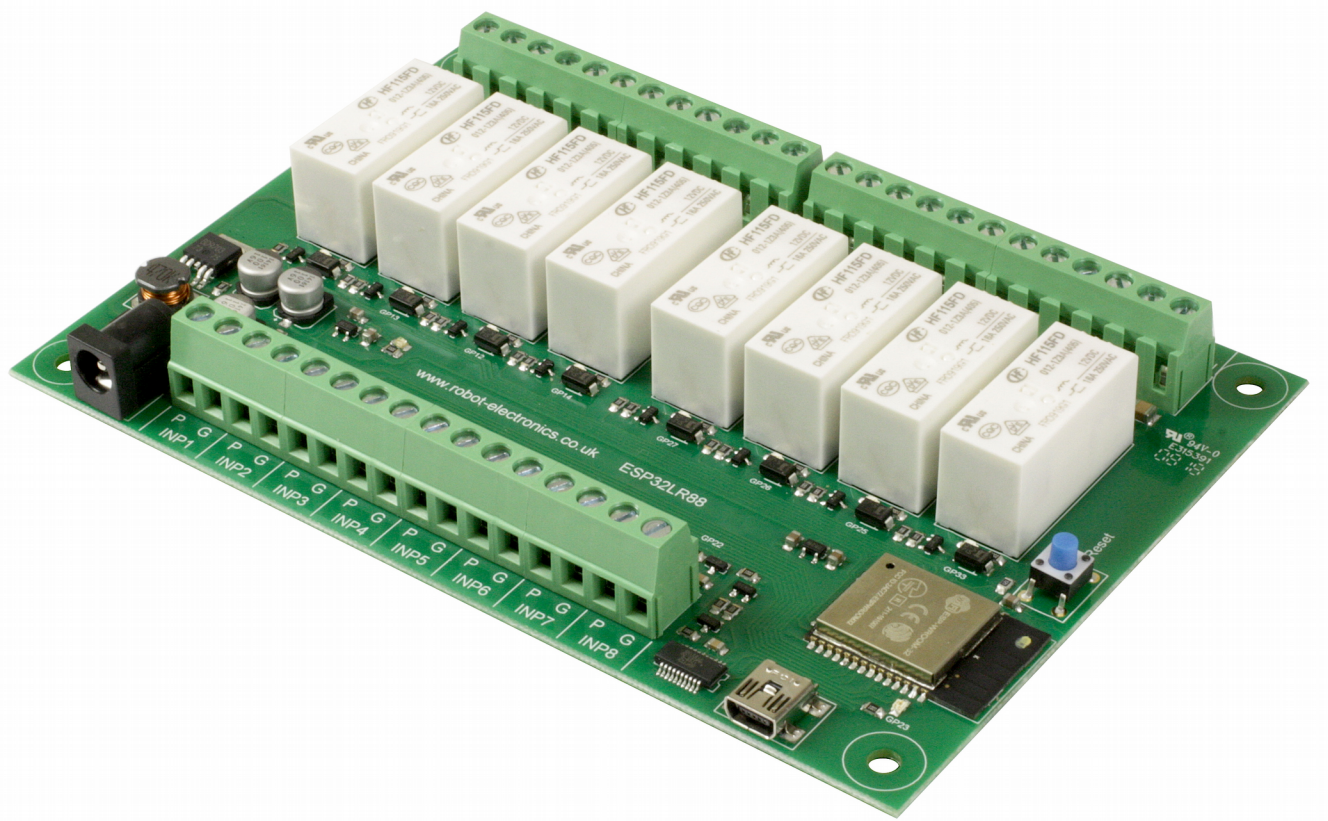


# ESP32LR88

User Manual Version 1.02



## Table of Contents

Overview.....	3
Control Protocols.....	3
WiFi.....	3
Configuration.....	3
USB Configuration Commands.....	4
ST Status. Return the system status.....	4
RB ReBoot.....	5
IP Sets the modules IP address.....	5
SB Sets the SuBnet mask.....	5
GW Sets the GateWay address.....	5
PD Sets the Primary DNS.....	6
SD Sets the Secondary DNS.....	6
SS This sets the SSID.....	6
PW Sets your networks WIFI password.....	6
PA Sets the TCP/IP port number for the ASCII commands.....	6
MS Sets the MQTT broker address.....	7
MD Sets the MQTT ID for this module.....	7
MP Sets the MQTT broker’s port.....	7
R1-R8 Sets the MQTT topic this relay is subscribed to.....	7
N1-N8 Sets the MQTT topic this Input will publish to.....	8
TCP/IP Commands.....	9
SR Set Relay.....	9
GR Get Relay.....	10
GI Get Input.....	10
AL Get All 8 inputs.....	10
HTML Commands.....	11
Webpage.....	12
Schematics.....	13
CPU.....	13
Power Supply.....	14
Relay Outputs.....	15
Digital Input.....	16
PCB dimensions.....	17
Notes.....	18

## Overview

The ESP32LR88 is a WIFI connected relay module using the popular ESP32.

It provides 8 relays capable switching up to 16Amps and 8 digital inputs with pull-ups than can interface directly with volt free contacts. The normally open relay contacts have snubbers fitted across them, and can drive inductive loads such as contactors and solenoids.

Power for the board is 12v dc, which can be provided by a standard universal wall power supply. A 1A or greater supply should be selected.

## Control Protocols

1. Simple plain text commands sent to the module.
2. HTML commands
3. MQTT
4. A built in webpage

## WiFi

The ESP32LR88 connects via WiFi to your network. Therefore it must be located in a position where it gets a good WiFi signal. The module should not be enclosed in a metal box/cabinet as this will shield the WiFi signal.

You can check the signal level by looking at the RSSI figure which is reported by the ST (STatus) command.

## Configuration

The ESP32LR88 is configured by connecting a USB cable to your PC and running a terminal program. PuTTY is a good option if you don't have any other preferences.

The serial port should be set to 115200 baud, 8 bit, 1 stop, no parity, no flow control.

## USB Configuration Commands

### ST Status. Return the system status

```
Status:
Firmware Version: 1.2
IP: 0.0.0.0 (192.168.0.48)
Subnet: 255.255.255.0
Gateway: 192.168.0.1
Primary DNS: 192.168.0.1
Secondary DNS: 8.8.4.4
SSID: *****
Password: *****
ASCII TCP Port: 17126
RSSI: -62
MQTT Server: 192.168.0.121
MQTT Port: 1883
MQTT ID: ModuleESP1
Relay1 Topic: R1Topic
Relay2 Topic: R2Topic
Relay3 Topic: R3Topic
Relay4 Topic: R4Topic
Relay5 Topic: R5Topic
Relay6 Topic: R6Topic
Relay7 Topic: R7Topic
Relay8 Topic: R8Topic
Input1 Topic: myNew/TopicA
Input2 Topic: myNew/TopicB
Input3 Topic: N3Topic
Input4 Topic: N4Topic
Input5 Topic: N5Topic
Input6 Topic: N6Topic
Input7 Topic: N7Topic
Input8 Topic: N8Topic
```

When the IP address is set to 0.0.0.0 this means that the IP address is being provided by your networks DHCP server. In that case the assigned IP address is also provided, as above.

When the SSID and Password are setup, they will be displayed until the next reset, After that they will only show as \*\*\*\*\*.

## RB ReBoot.

This will restart the module. It may produce a lot of random characters as the ESP32's boot logging runs at a different baud rate. If it succeeds in connecting to your network it will report the IP address.

Re-Booting.. .

嶮???????#XL# ##C????5)5)????ia?b???????命#??\*UY?o????? #i#U?5 ?Q?????

WiFi connected.

IP address:

192.168.0.6

## IP Sets the modules IP address.

Enter IP followed by the required IP address. Entering address 0.0.0.0 means the IP will be obtained from your networks DHCP server. The new IP address will take effect after the next re-boot.

IP "192.168.0.123"

OK. Saved IP Address: 192.168.0.123

## SB Sets the SuBnet mask.

SB "255.255.255.0"

OK. Saved Subnet Mask: 255.255.255.0

## GW Sets the GateWay address.

This is normally the IP address of your router.

GW "192.168.0.1"

OK. Saved Gateway Address: 192.168.0.1

## PD Sets the Primary DNS.

The can be the IP address of your router which will then use your ISP provided DNS. You can also specify the DNS such as 8.8.8.8 for Googles DNS server.

PD "192.168.0.1"

OK. Saved Primary DNS: 192.168.0.1

## SD Sets the Secondary DNS.

The can be the IP address of your router which will then use your ISP provided DNS. You can also specify the DNS such as 8.8.4.4 for Googles DNS server.

SD "8.8.4.4"

OK. Saved Secondary DNS: 8.8.4.4

## SS This sets the SSID.

The SSID is the public name of your WIFI network Enter your WIFI's SSID here.

SS "Devantech"

OK. Saved SSID: Devantech

## PW Sets your networks WIFI password.

PW "K]~kCZUV\*UGA6SG~"

OK. Saved Password: K]~kCZUV\*UGA6SG~

## PA Sets the TCP/IP port number for the ASCII commands.

PA 17126

OK. Saved ASCII port number: 17126

## **MS Sets the MQTT broker address**

MS "192.168.0.121"  
OK. Saved MQTT Server: 192.168.0.121

## **MD Sets the MQTT ID for this module**

MS "UniqueModuleName"  
OK. Saved MQTT ID: UniqueModuleName

## **MP Sets the MQTT broker's port.**

Normally, this should be set to 1883.

mp 1883  
OK. Saved MQTT port number: 1883

If you are not using MQTT, set the port to 0. This will turn off MQTT, otherwise it will continuously try to connect if there is no MQTT broker.

## **R1-R8 Sets the MQTT topic this relay is subscribed to.**

R3 "Workshop/Heater"  
OK. Saved Relay 3 Topic: Workshop/Heater

In use, the payload for relay topics should be a string with the first character set to '0' or '1' (ASCII characters 0x31/0x30).

## **N1-N8 Sets the MQTT topic this Input will publish to.**

N2 "Workshop/Heater"

OK. Saved Input 2 Topic: Workshop/Heater

The payload generated for input topics is a string with the first character set to '1' if the input is open or unconnected, and '0' if the input pins are shorted. (ASCII characters 0x31/0x30).



## TCP/IP Commands.

The ESP32LR88 has a built in TCP/IP command set which allows you to control the module remotely.

All commands are sent using plain ASCII text. PuTTY is a good cross platform terminal program to use for testing. The TCP/IP port is the one you set-up with the PA command during USB configuration. Do not use port 80 as that is reserved for the HTML commands and Webpage.

### SR Set Relay.

This is used to turn a relay on or off

To turn Relay 1 on:

```
SR 1 1
```

The first number is the relay number from 1 to 8.

The second number is 1 or 0, on or off.

So turn turn relay 1 off again:

```
SR 1 0
```

The command will respond with ok or fail.

```
SR 1 1
```

```
ok
```

```
SR 1 6
```

```
fail
```

< 6 is not valid, only 1 or 0 for on/off

```
SR 9 1
```

```
fail
```

< relay 9 does not exist.

## GR Get Relay.

Will return the state of the relay.

To get the status of relay 6:

GR 6

1

GR 6

0

GR 9

fail

< relay 9 does not exist.

## GI Get Input.

Will return the status of an input.

GI 2

0

Input 2 is low (Green Led is on)

GI 2

1

Input 2 is high (Green Led is off)

GI 9

fail

Only 8 inputs available

## AL Get All 8 inputs.

AL

11111011

Here, Input 6 is low, all others are high.

Inputs are numbered from left to right, 1 to 8.

## HTML Commands.

There are a set of HTML commands that can be used to control the module.

?Rly3=1      This will turn on relay 3  
?Rly3=0      This will turn off relay 3  
?Rly3=2      This will toggle relay 3 to the opposite state.

You can enter the commands into a browser immediately after the IP address.

<http://192.168.0.3/?Rly3=1>

This will turn on relay 3.

In response the module will return an XML file, which your browser will display.

```
<?xml version="1.0" encoding="UTF-8"?>
<ESP32LR88DATA>
  <RELAYS>
    <RLY1>off</RLY1>
    <RLY2>off</RLY2>
    <RLY3>on</RLY3>
    <RLY4>off</RLY4>
    <RLY5>off</RLY5>
    <RLY6>off</RLY6>
    <RLY7>off</RLY7>
    <RLY8>off</RLY8>
  </RELAYS>
  <INPUTS>
    <INP1>1</INP1>
    <INP2>1</INP2>
    <INP3>1</INP3>
    <INP4>1</INP4>
    <INP5>1</INP5>
    <INP6>0</INP6>
    <INP7>1</INP7>
    <INP8>1</INP8>
  </INPUTS>
</ESP32LR88DATA>
```

The XML file is generated after the command has executed, therefore will reflect the new status of the relays.

## Webpage

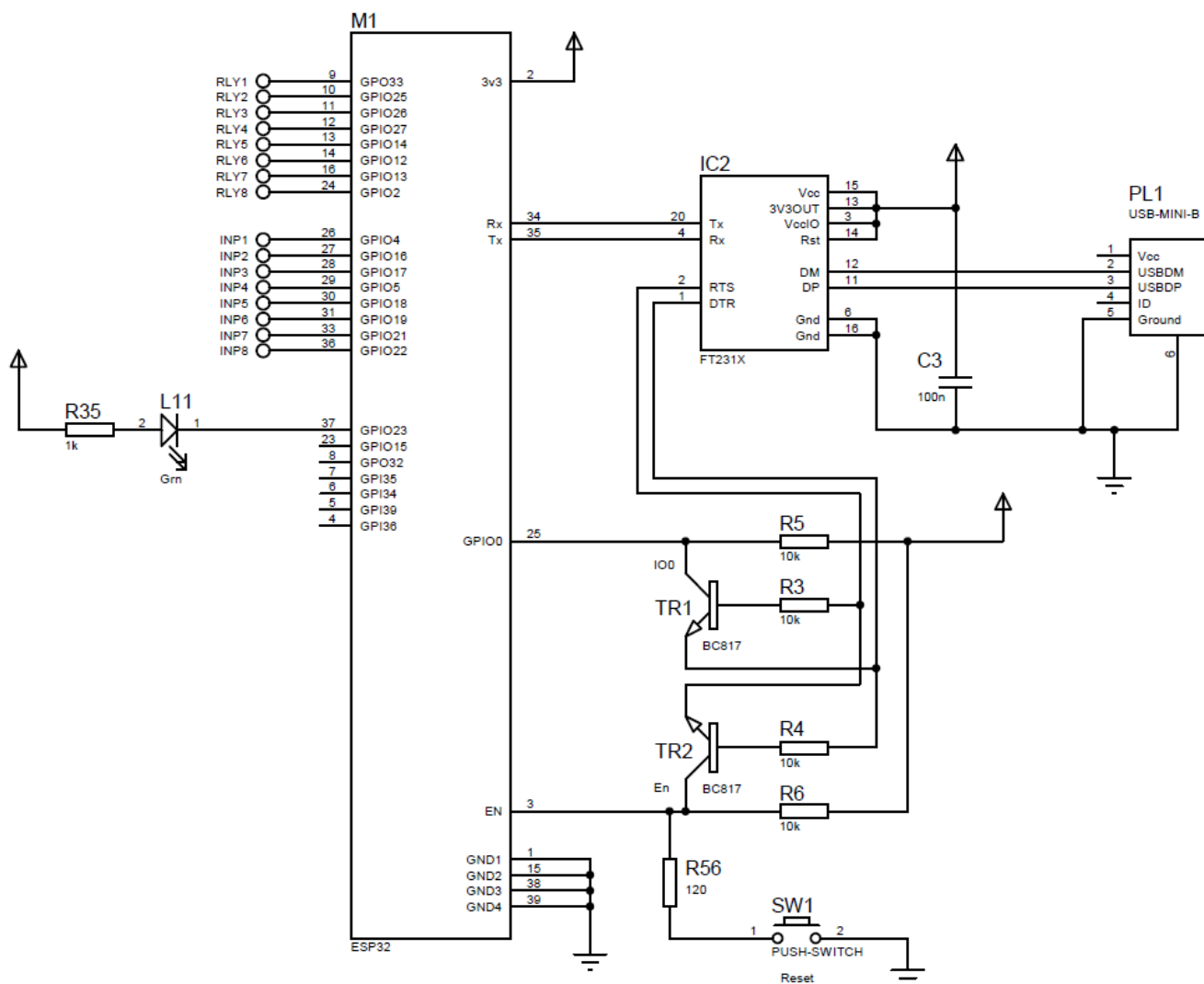
The built in webpage can be used as a remote app to monitor and control the relays. You can access the page as a default with just the IP address or by specifying index.htm.



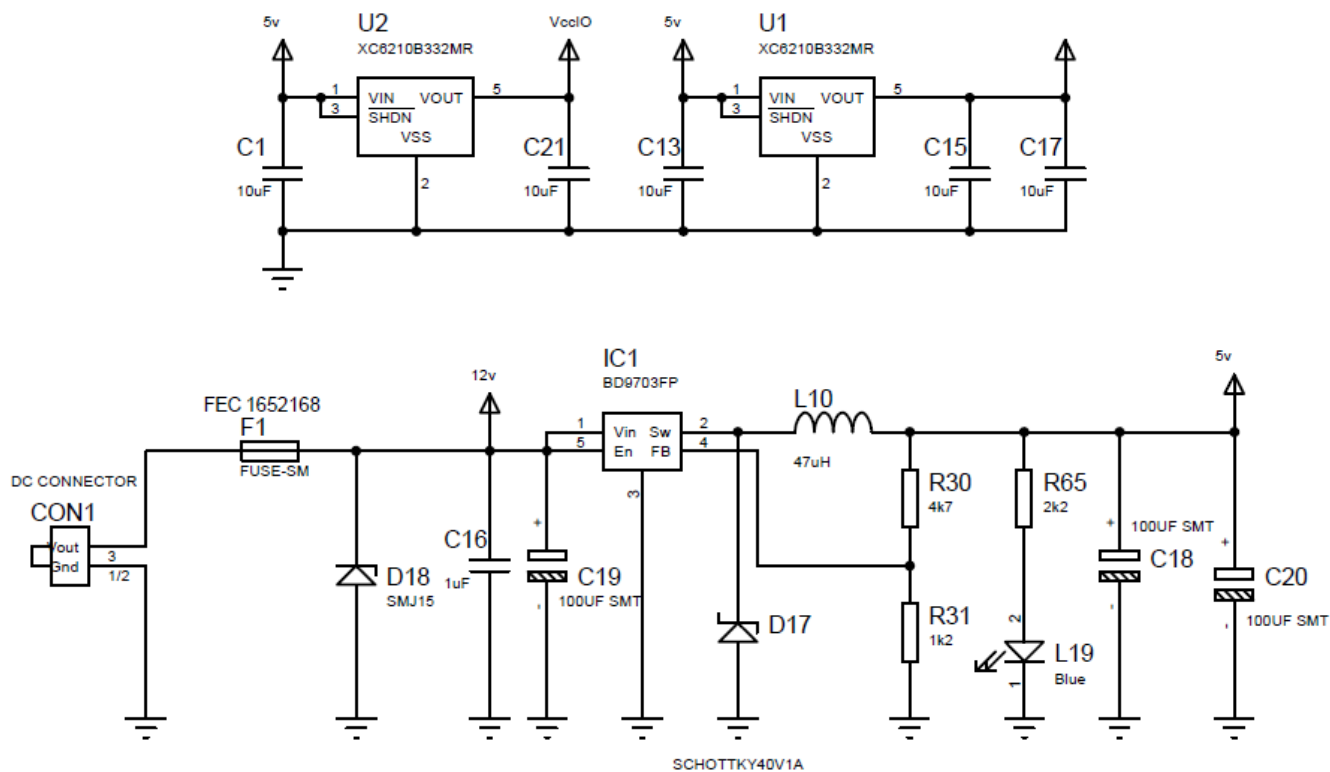
The webpage contains the Javascript to send an HTML toggle command, as described in the previous section. It will send the toggle command each time a button is clicked. It then uses the responding XML file to colour the buttons and set the Input buttons to indicate input status.

## Schematics

### CPU

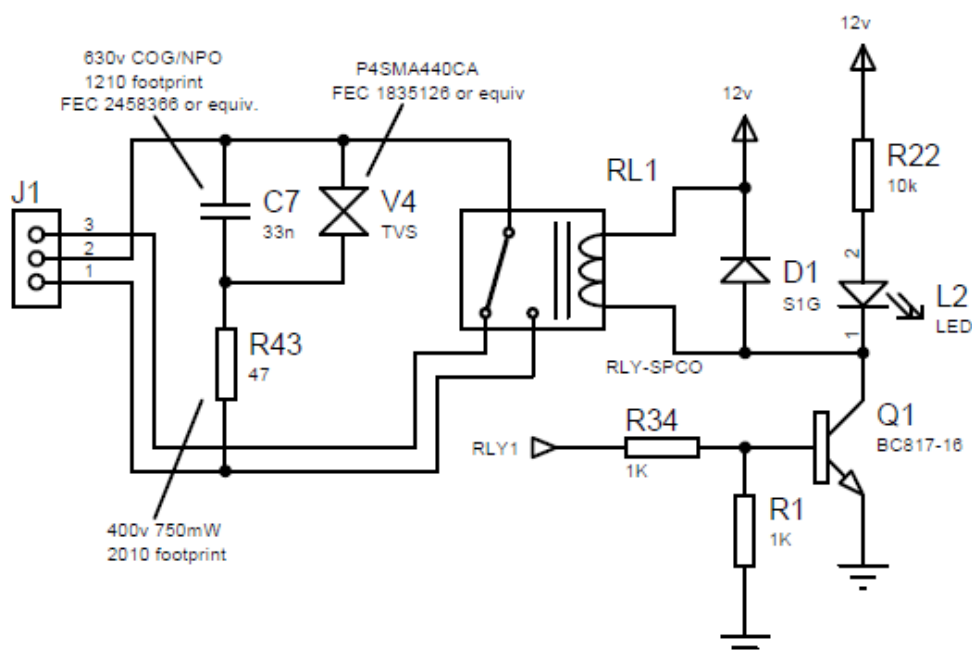


## Power Supply



## Relay Outputs

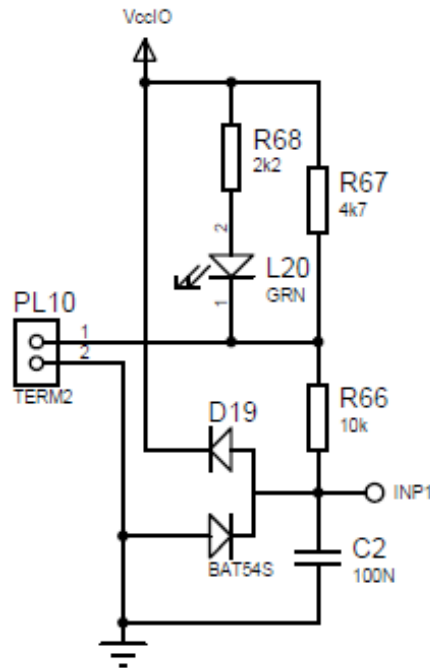
1 of 8 identical circuits shown



The relays are capable of switching up to 16 Amps at 24vdc or 230vac. A data sheet of the relay can be found [here](#). Only the normally open (N/O) contact has the snubber circuitry.

## Digital Input

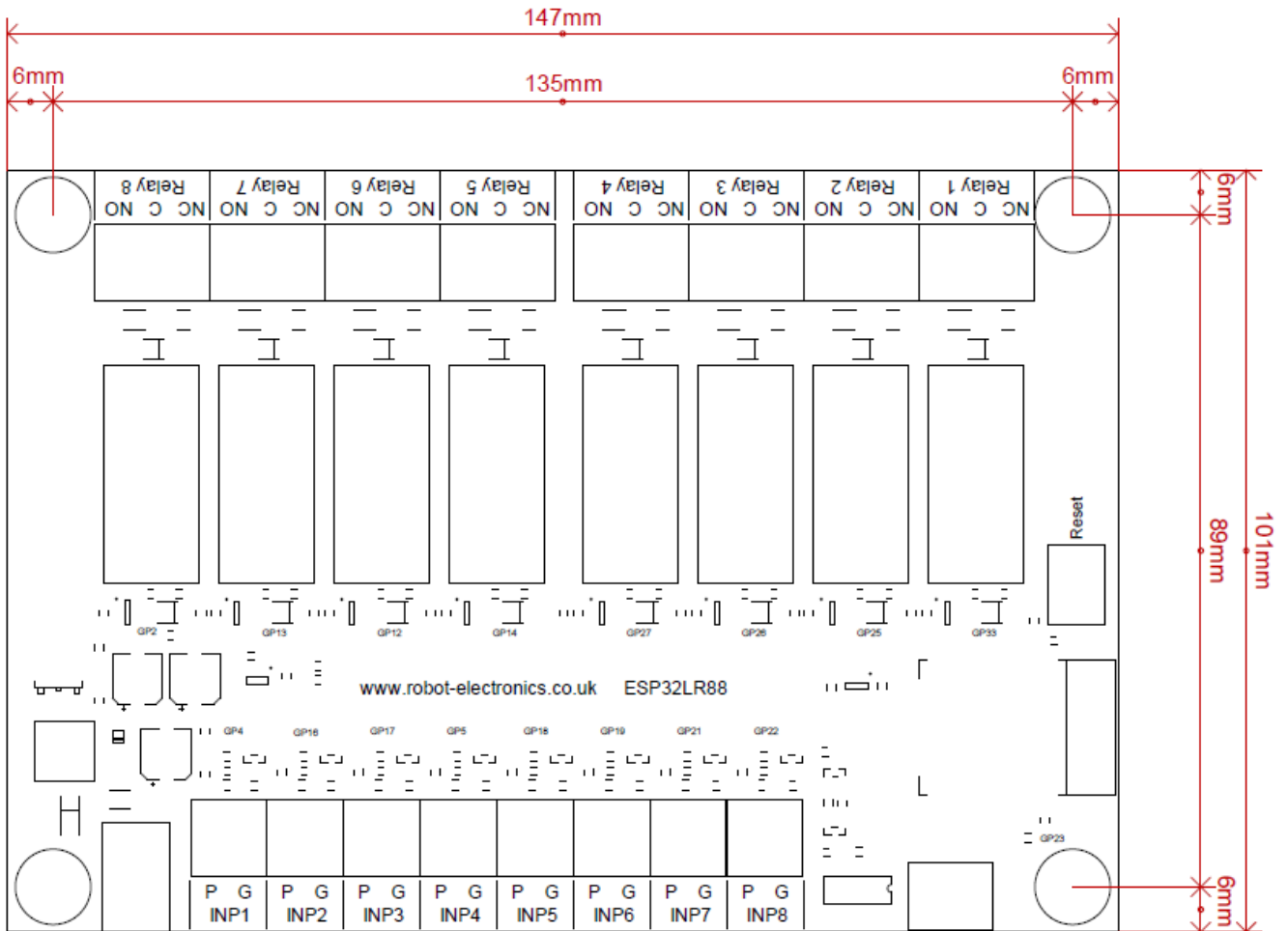
1 of 8 identical circuits shown



Digital Inputs have a pull-up resistor to 3.3v and will operate with a simple contact closure to ground. Alternatively, they may be connected to 3.3v logic. The clamping diodes prevent operation at voltages higher than 3.3v, so don't connect to 5v logic.



## PCB dimensions



## Notes