**V**

RASPBERRY 4 PI HAT

MODEL B

MANUAL

Contents

[Description 3](#_Toc74149078)

[Optocoupled inputs circuit 3](#_Toc74149079)

[Relay circuit 4](#_Toc74149080)

[Power Circuit 6](#_Toc74149081)

[Raspberry Hat inputs and outputs pinout 7](#_Toc74149082)

[PCB Layout connection 8](#_Toc74149083)

[3D Layout 9](#_Toc74149084)

[Python test program for the Raspberry Hat 11](#_Toc74149085)

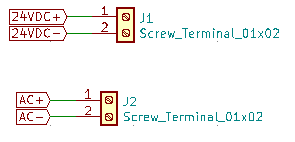
# Description

This hat is a device that allows us to make connections of optocoupled inputs and relay outputs with our Raspberry 4 Pi model B.

Next, the circuit necessary to carry out the elaboration of this board will be described, in addition to the PCB and 3D designs.

The elaboration of this plate was carried out in the free software Kicad, so the file of the plate will be attachedso that it can be downloaded by any user.

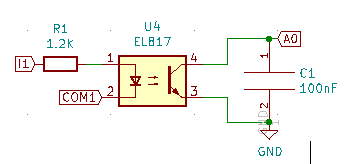
The Raspberry 4 Pi Hat model B card has a 24 Vdc output and a 24 Vac input.



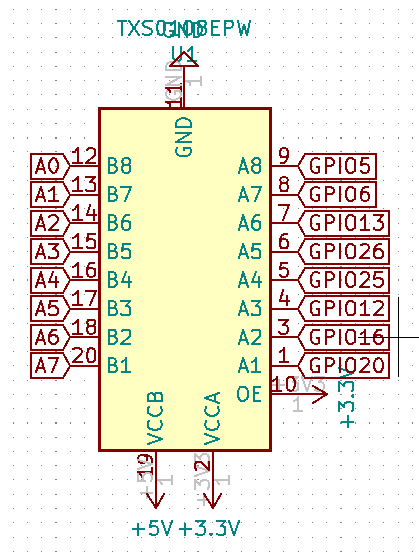
Picture 1. 24 VDC output and a 24 VAC input

# Optocoupled inputs circuit

It includes eight opto-coupled inputs connected to the TXS0108EPW module and eight outputs connected to the pins of the Raspberry card (GPIO5, GPIO6, GPIO13, GPIO26, GPIO25, GPIO12, GPIO16, GPIO20).



Picture 2. optocoupled inputs

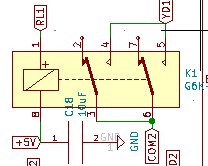
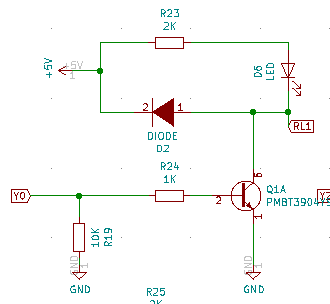


Picture 3. Bidirectional module TXS0108EPW

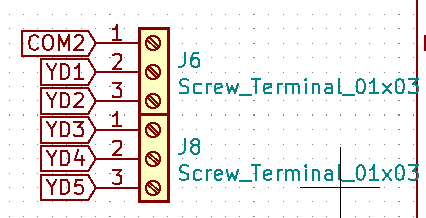
It has resistors connected to the inputs of the TXS0108EPW with a 5 v supply, this device helps us to communicate our inputs and outputs that work with 5V, with the Raspberry GPIO pins that only accept a voltage of 3.3V.

# Relay circuit

The relay circuit works due to the signals that come out of the Raspberry and are passed from 3.3V to 5V by the second module TXS0108EPW, which then goes through a transistor until it reaches the relay, it also has a relay power indicator led.



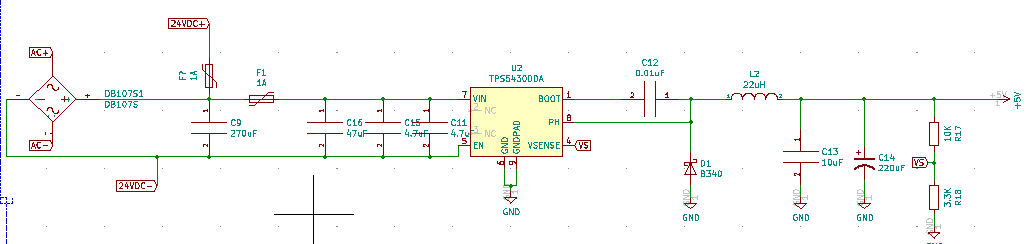
Picture 4. Relay circuit.



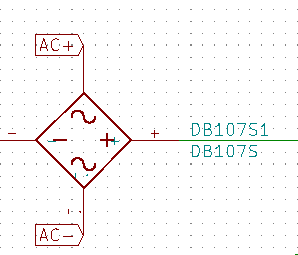
Picture 5. Output relay

# Power Circuit

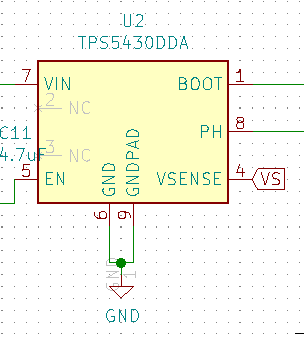
Circuit in charge of converting the voltage from 24 Vac, to 24Vac AND 5Vac. Composed of a DB107S1 diode bridge, fuses to protect the circuit and the TPS5430DDA module.



Picture 6. Power circuit

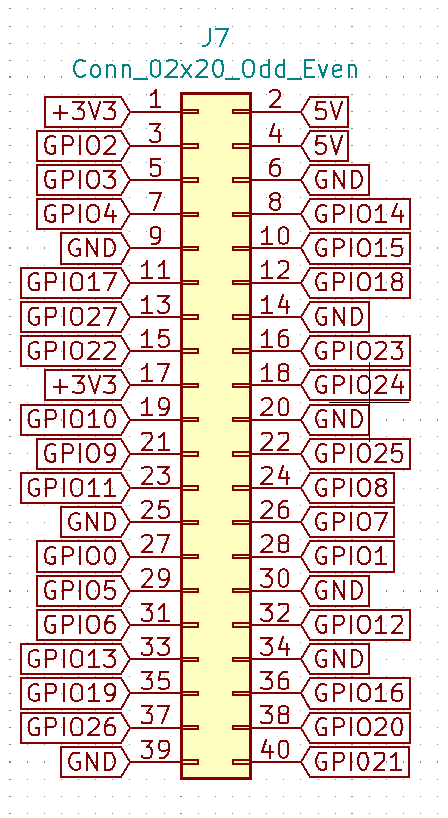


Picture 7. DB107S



Picture 8. TPS5430DDA

# Raspberry Hat inputs and outputs pinout

**Optocoupled inputs**

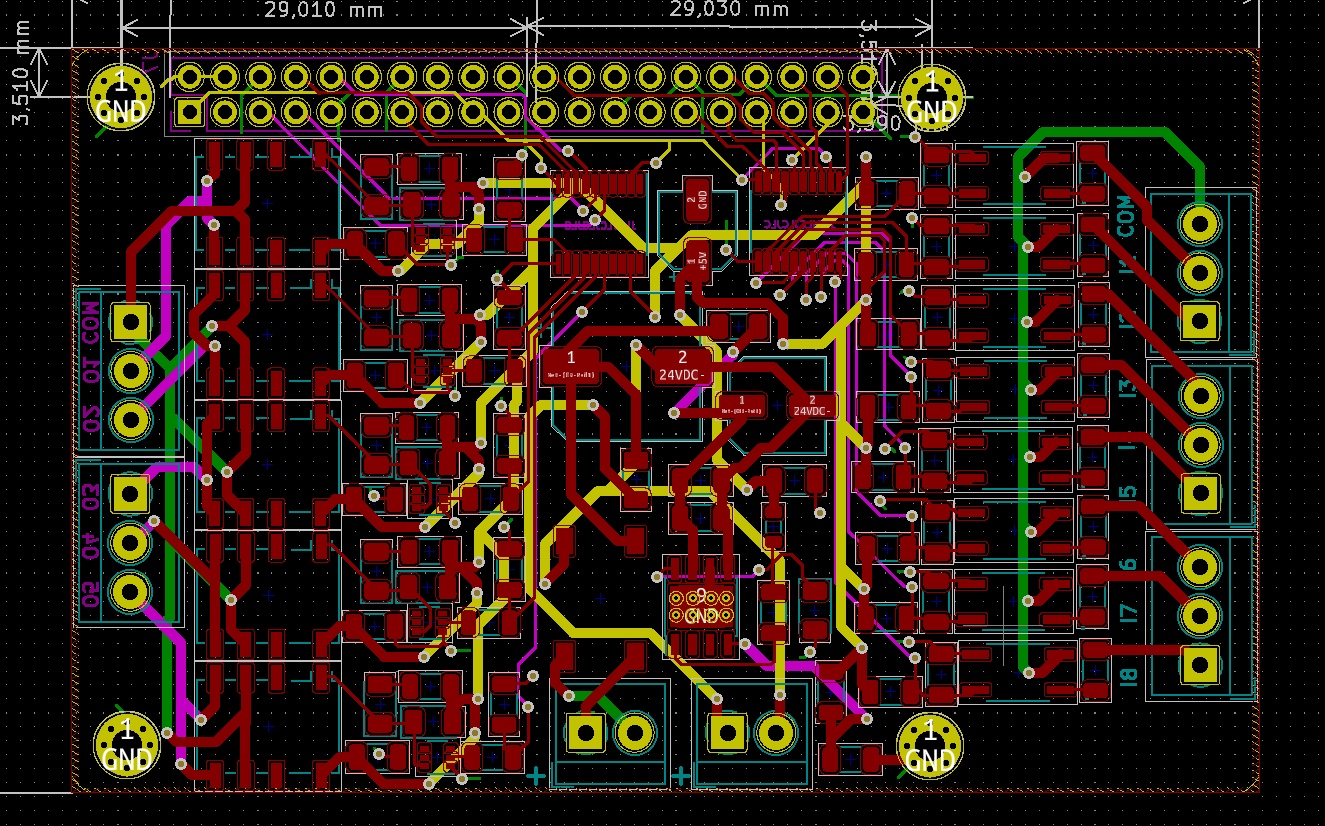
Picture 9. Pinout RaspBerry 4 Pi Hat B

1. PIN 22 / GPIO25
2. PIN 29 / GPIO5
3. PIN 31 / GPIO36
4. PIN 32 / GPIO12
5. PIN 33 / GPIO13
6. PIN 36 / GPIO16
7. PIN 37 / GPIO26
8. PIN 38/ GPIO20

**Relay outputs**

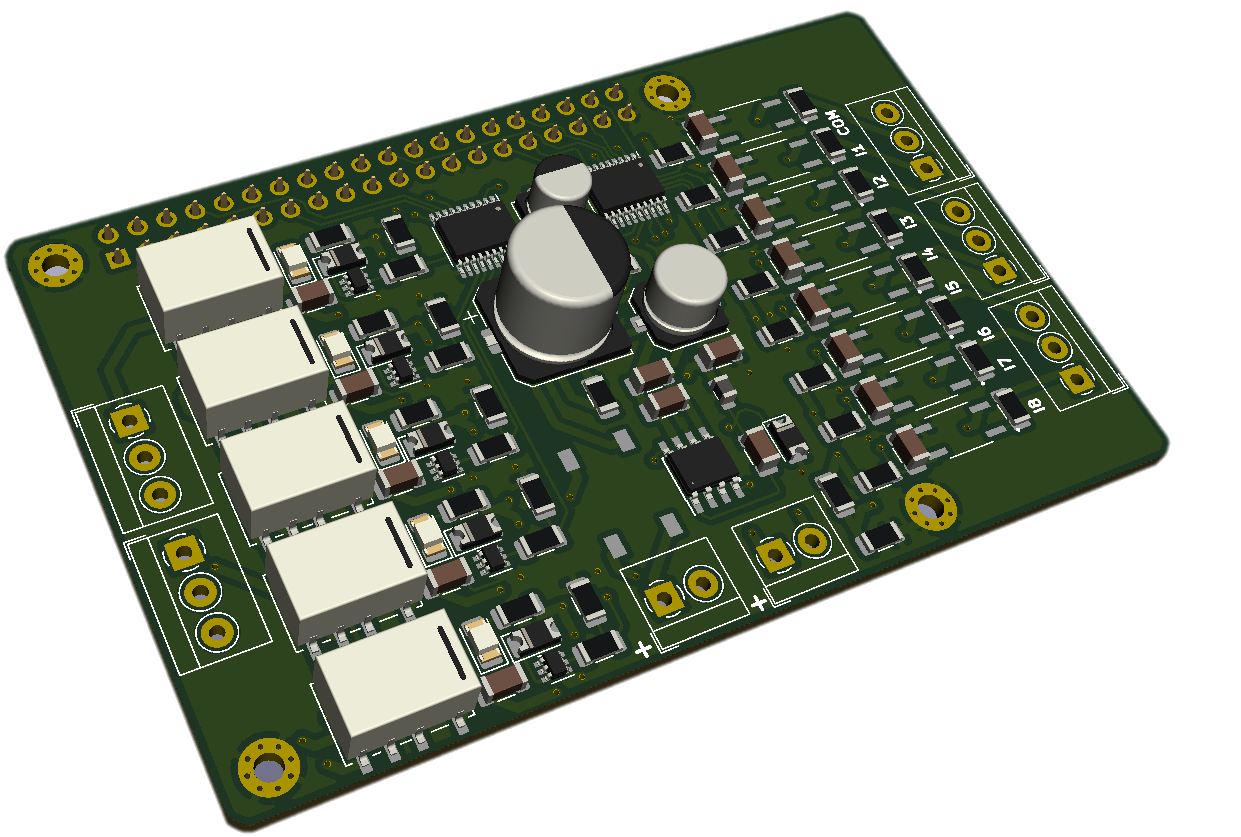
1. PIN 5 / GPIO3
2. PIN 7 / GPIO4
3. PIN 8/ GPIO14
4. PIN 13 / GPIO27
5. PIN 10 / GPIO15

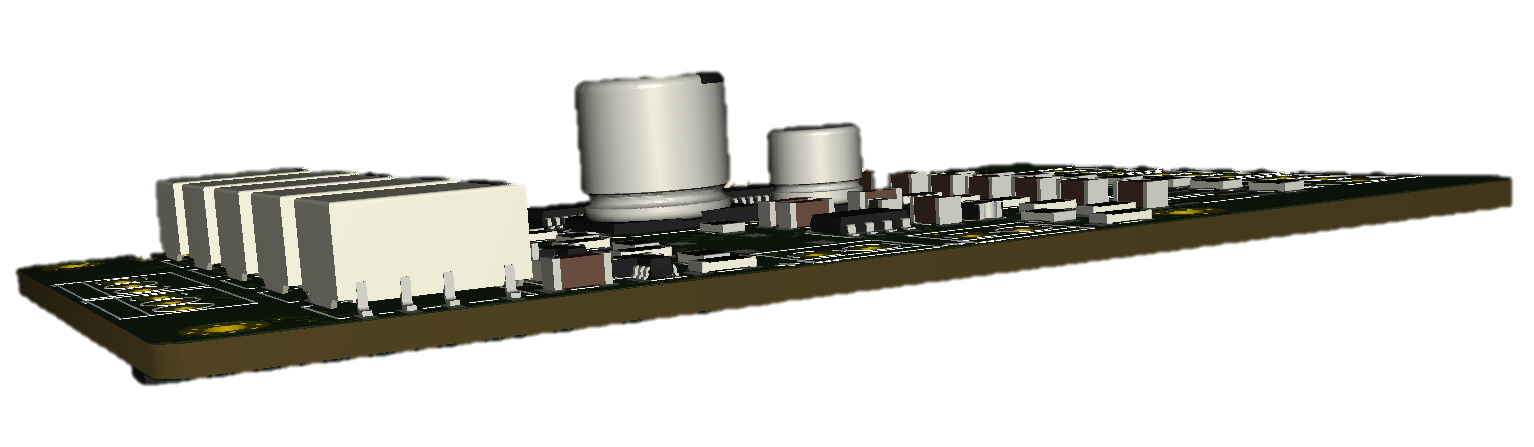
# PCB Layout connection

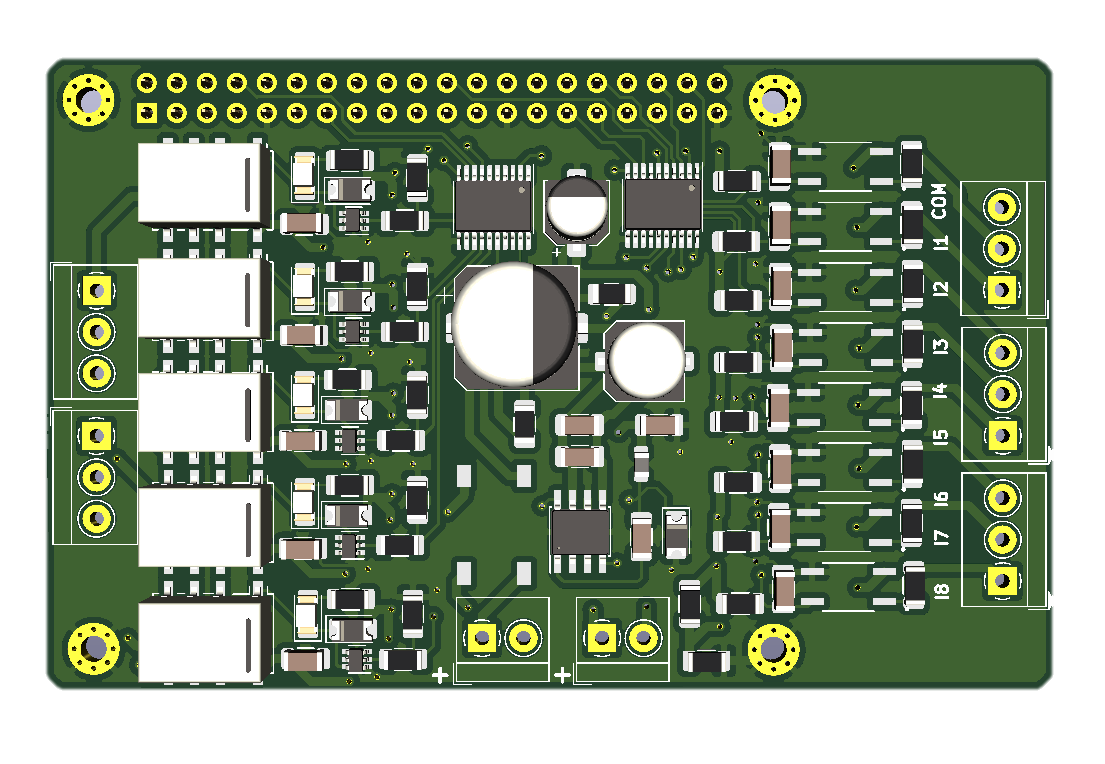
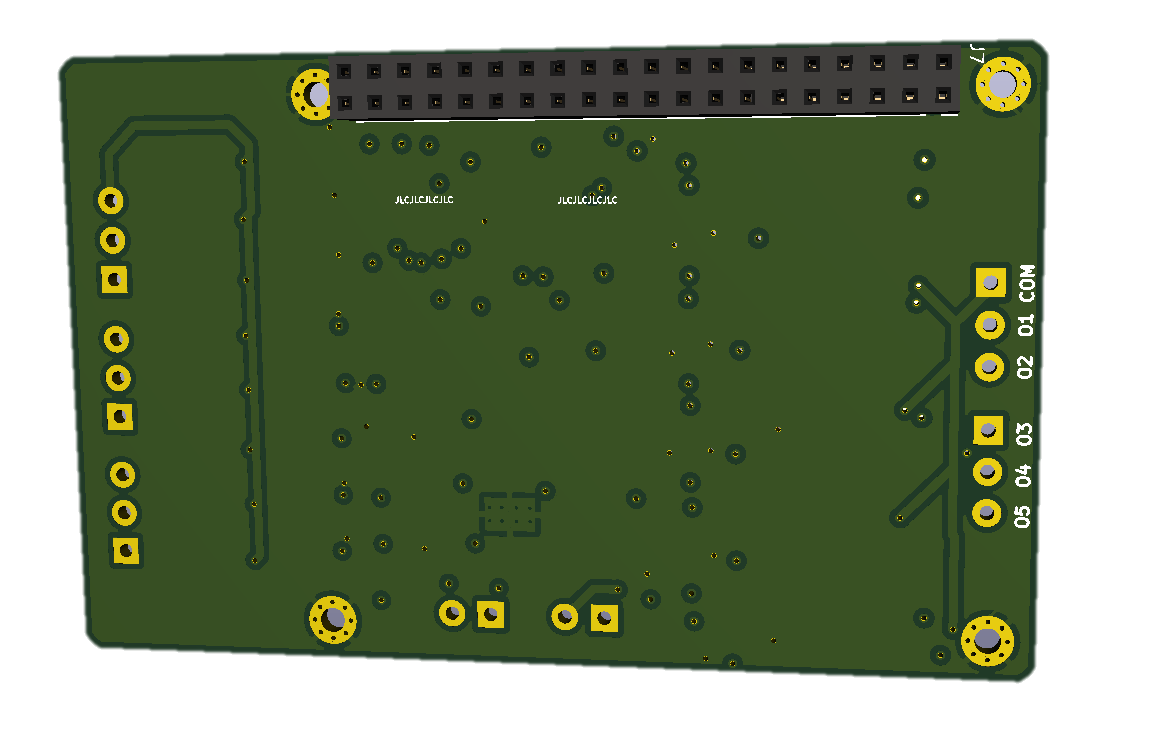
**

Picture 10. PCB Layout

# 3D Layout







# Python test program for the Raspberry Hat

// Import of the Rpi.GPIO library //

import time

// The way the Raspberry GPIO pins will be recognized in the code is established, for this case it is established that it will be by the pin number (BOARD) and not by the GPIO number (BCM) //

GPIO.setmode(GPIO.BOARD)

// Declaration of input and output pins //

GPIO.setup(5, GPIO.OUT)

GPIO.setup(7, GPIO.OUT)

GPIO.setup(8, GPIO.OUT)

GPIO.setup(13, GPIO.OUT)

GPIO.setup(10, GPIO.OUT)

GPIO.setup(22, GPIO.IN)

GPIO.setup(29, GPIO.IN)

GPIO.setup(31, GPIO.IN)

GPIO.setup(32, GPIO.IN)

GPIO.setup(33, GPIO.IN)

GPIO.setup(36, GPIO.IN)

GPIO.setup(37, GPIO.IN)

GPIO.setup(38, GPIO.IN)

// Cycle where the states of the opto-coupled inputs will be evaluated and depending on whether the value is equal to high or 1, the relay will turn on (For this case, only the use of 4 inverse logic relays was used, so the high state = 0 and low state = 0) //

while True:

if GPIO.input (22) or GPIO.input(33) == 1:

GPIO.output(5, False)

else:

GPIO.output(5, True)

if GPIO.input (29) or GPIO.input(36) == 1:

GPIO.output(7, False)

else:

GPIO.output(7, True)

if GPIO.input (31) or GPIO.input(37) == 1:

GPIO.output(8, False)

else:

GPIO.output(8, True)

if GPIO.input (32) or GPIO.input(38) == 1:

GPIO.output(13, False)

else:

GPIO.output(13, True)