

# clickyBoard Documentation

*For board version 1.1*

Download this document at [github.com/SecretImbecile/clickyBoard](https://github.com/SecretImbecile/clickyBoard)

## Contents

<b>1</b>	<b>The clickyBoard</b>	<b>1</b>
<b>2</b>	<b>Parts Required</b>	<b>2</b>
2.1	Mechanical Keyboard Switches . . . . .	2
2.2	LED and Resistor Choice . . . . .	3
<b>3</b>	<b>Assembly Guide</b>	<b>4</b>
<b>4</b>	<b>Example Code</b>	<b>6</b>
<b>5</b>	<b>Technical Documentation</b>	<b>7</b>
5.1	Schematic design for the clickyBoard . . . . .	7
5.2	PCB layout for the clickyBoard . . . . .	8
<b>6</b>	<b>License</b>	<b>10</b>

## 1 The clickyBoard

The *clickyBoard* is an add-on board for the Raspberry Pi computer (model B+ or later), which adds a simple switch and LED circuit for use in GPIO programming.

The clickyBoard has been designed to use Cherry MX series switches or compatible switches from other manufacturers, designed for use in keyboards that are widely available and cheaply purchased.

The board, through the Raspberry PI GPIO, provides four push-buttons inputs, as well as a single output through which you can control the four backlight LEDs under each key.

In addition to technical documentation, this document also includes a parts list and assembly guide, for users wishing to assemble their own clickyBoard from scratch, or kit form.

## 2 Parts Required

The following components are required to fully populate a clickyBoard:

- two  $1k\Omega$  resistors
- four T1 (3mm) LEDs
- four resistors, of a value determined in subsection 2.2
- one TO-220 package logic level power MOSFET
- four Cherry MX compatible switches
- a female 40-pin connector for the Raspberry Pi GPIO header<sup>1</sup>

### 2.1 Mechanical Keyboard Switches

The clickyBoard has mounting holes for MX series or compatible switches. These are switches designed for use in premium computer keyboards, which are widely available, along with compatible keycaps.



Figure 1: A Cherry MX Blue switch

If you require quick delivery, the german manufactured **Cherry MX** series of switches are more readily available from several on-line retailers. In the United Kingdom, one possibility would be a computer hardware retailer like [kustompcs.co.uk](http://kustompcs.co.uk). In the US and internationally, [wasdkeyboards.com](http://wasdkeyboards.com) specialises in mechanical keyboard parts. Cherry switches are likely to cost around £1 per switch.

If you're willing to wait for shipping from China, there are several manufacturers that make compatible switches, notably **Gateron** and *Kailh* switches. Gateron switches can be purchased on eBay for as little as 10 for £1.

There are several types of MX switch available, denoted by the colour of their centre plunger, as shown in Figure 1. We recommend **blue** coloured switches, for full authentic clickiness, but others are available.

If possible, opt for '**PCB mount**' switches (sometimes referred to as '5-pin'), as these have additional plastic pegs which will ensure the switches are correctly oriented. Using 'plate mount' switches will usually result in slightly misaligned switches.

---

<sup>1</sup>a  $2 \times 20$  header with  $2.54mm$  (0.1") pitch

Switch Colour	Click	Tactile Bump	Heavy Spring
Blue	✓	✓	
Green	✓	✓	✓
Brown		✓	
Clear		✓	✓
Red			
Black			✓

## 2.2 LED and Resistor Choice

To fit within the housing of an MX compatible switch, a 3mm (T1) LED must be chosen.

To provide sufficient current, the LEDs on the clickyBoard are powered through the Raspberry Pi's 5 volt supply, rather than the current-limited 3V3 GPIO pins, and are instead controlled using a power MOSFET and a single GPIO output.

Each LED has a resistor in series. The required resistance for the circuit can be calculated using this variant of the standard resistance formula, where  $V_s$  is the supply voltage, and  $V_f$  is the forward voltage of the chosen LED.

$$R = \frac{V_s - V_f}{I}$$

For the red 2V 20mA LEDs used in the schematic, the resistance for the LED resistors was calculated as follows.

$$R = \frac{5V - 2V}{20 \times 10^{-3}A} = 150\Omega$$

## 3 Assembly Guide

### Soldering Basics

This guide will assume that the reader is familiar with basic soldering techniques. The components on the clickyBoard are all through-hole components, so it may be a suitable board for a novice solderer. Be aware that the 40-pin GPIO header can be tricky, especially with an old soldering iron, or thick gauge solder wire.

With that said, here are some resources for learning to solder. Video tutorials are well suited for soldering, as you'll need to be able to identify a good solder joint from a bad one.

- *Soldering basics and choosing a cheap soldering iron* (00:00–12:00), bigclive-dotcom — <https://www.youtube.com/watch?v=aIab66EgfHM>
- *EEVblog #183 - Soldering Tutorial Part 2*, EEVblog — <https://www.youtube.com/watch?v=fYz5nIHH0iY>

### Resistors

For convenience, we always solder components from lowest height to highest, to maintain a steady PCB on which to work. Begin by soldering the two  $1k\Omega$  resistors in slots r1 and r2.

Next, solder the resistors for the LEDs that you chose in subsection 2.2 into the sockets r3–r6.

### MOSFET

Bend the legs of your TO-220 package MOSFET to a 90 degree angle and place it in the socket so that the large metal tab is flush with the PCB. You can optionally solder the tab to the large pad on the board, then solder the three main pins into their holes.

If you are **not** using a MOSFET to control the LEDs, you should jumper the two topmost pins, either by bridging the holes with solder, or with a piece of wire.

### Switches & LEDs

#### Switch assembly

In MX style switches, the 3mm LED sits in a slot at the bottom of the switch, and the leads are fed through holes in the switch housing. It is recommended to place the LED in the switch, then inserting that complete assembly into the board to solder.

Ensure that the polarity of the LED is correct, and the switch is sitting flush with the board with the plastic peg(s) in their respective holes, then solder the connections of both the switch and LED.

## **40-pin GPIO header**

To complete the *clickyBoard*, solder the 40-pin female header to the board, with the female socket on the reverse side of the board.

If you choose not to solder all of the pins, the following pins are required (GPIO board pinout numbering): 1 4 9 11 12 13 15 16, omitting 11 if a MOSFET was not used.

## 4 Example Code

The clickyBoard provides the following GPIO devices (all pins are GPIO board numbers):

- LED brightness control through pin **11**
- switch inputs through pins **12, 13, 15, and 16**

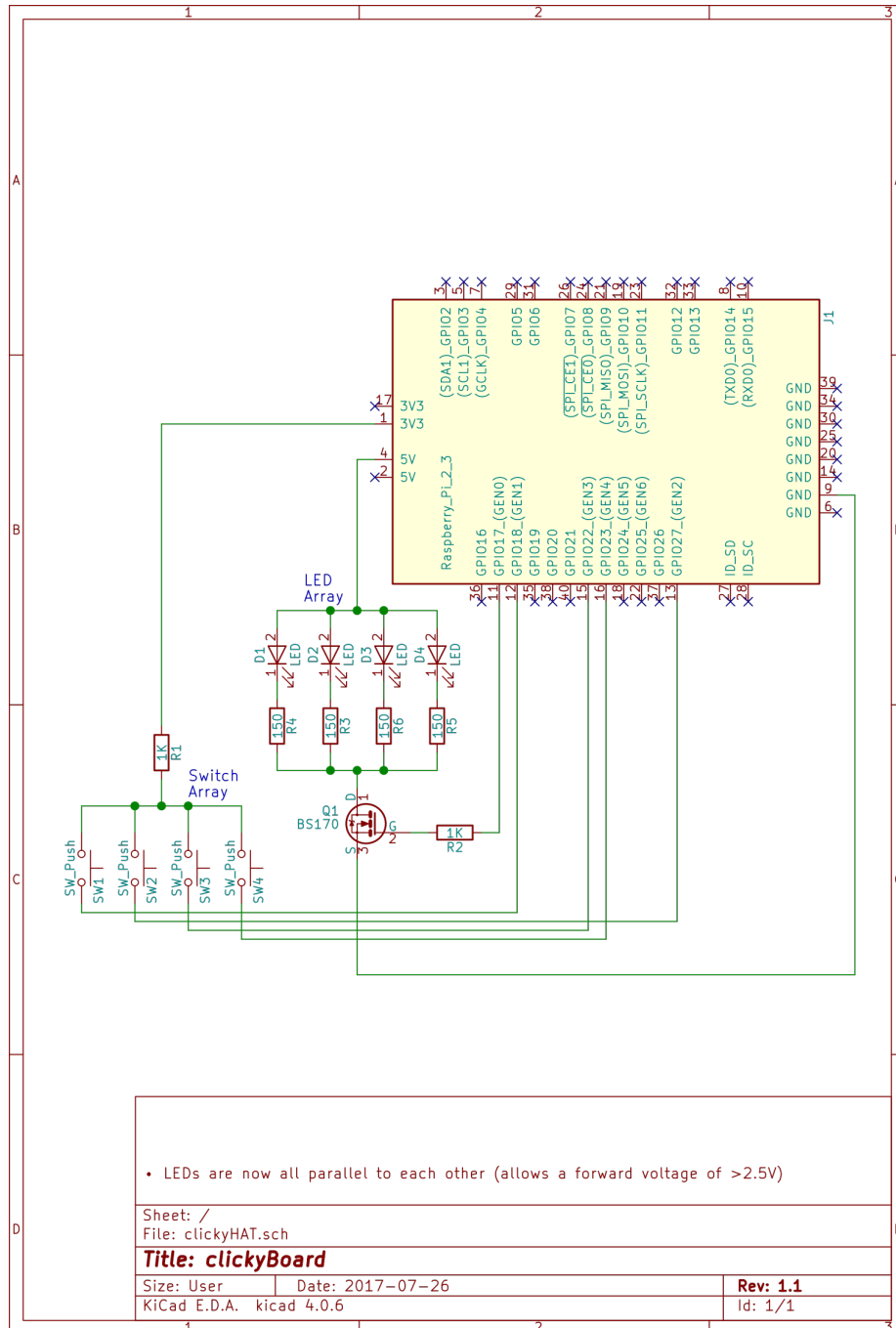
The switch inputs require the internal GPIO pull-down resistors in the Raspberry Pi to be enabled. enable them by setting up the pins with the `pull_up_down` parameter set to `GPIO.PUD_DOWN`.

The following example Python program will demonstrate the functionality of the clickyBoard. (note that `GPIO.cleanup()` is not run following this program)

```
clickyboardtest.py
1 import RPi.GPIO as GPIO
2 import time
3
4 # Setup Pins
5 GPIO.setmode(GPIO.BOARD)
6
7 GPIO.setup(11, GPIO.OUT)
8 GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
9 GPIO.setup(13, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
10 GPIO.setup(15, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
11 GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
12
13 # Setup Backlight
14 pwm = GPIO.PWM(11, 60)
15 pwm.start(100)
16
17 while True:
18     #Create a count of the buttons being pressed
19     buttons_pressed = 0
20
21     if GPIO.input(12) == GPIO.HIGH:
22         buttons_pressed += 1
23     if GPIO.input(13) == GPIO.HIGH:
24         buttons_pressed += 1
25     if GPIO.input(15) == GPIO.HIGH:
26         buttons_pressed += 1
27     if GPIO.input(16) == GPIO.HIGH:
28         buttons_pressed += 1
29
30     #Reduce PWM brightness based on number of buttons
31     pwm.ChangeDutyCycle(100 - (25 * buttons_pressed))
32
33     time.sleep(0.1)
```

## 5 Technical Documentation

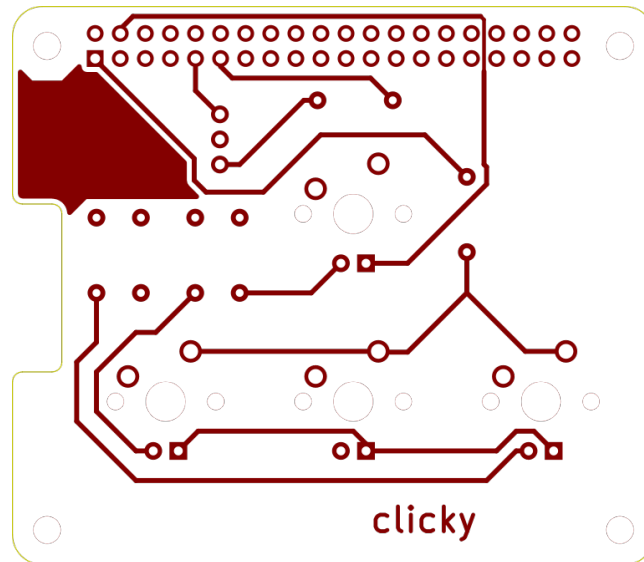
### 5.1 Schematic design for the clickyBoard



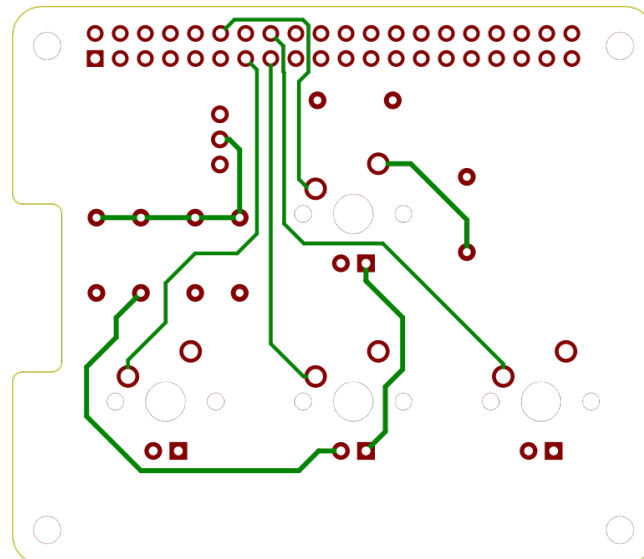
## 5.2 PCB layout for the clickyBoard

Gerber outputs (with additional layers) are available on the GitHub repo.

### Front Copper

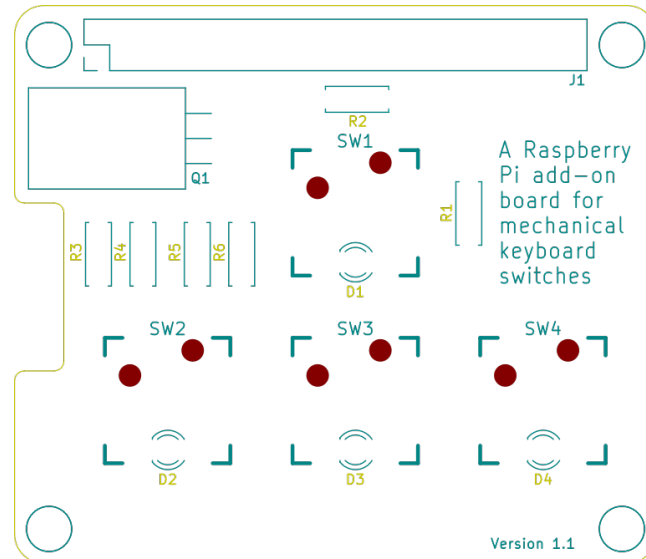


### Back Copper





## Front Silkscreen



## Design rules for PCB

The design rules used for this PCB were as follows.

Clearance	Track Width	Via Size	Silkscreen line thickness
0.2mm	$\geq 0.375mm$	n/a	$\geq 0.15mm$

## 6 License

The complete hardware design for the clickyBoard, including the files made available on the clickyBoard GitHub repo and all documentation, are released in accordance with the Open Source Hardware (OSHW) Definition, under the **Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)** license.

Every effort has been made to comply with the OSHW statement of principles. If you feel any aspect of the OSHW principles have been neglected, please raise the issue with the creator (a GitHub issue report would be suitable).