

Raspberry Pi Lightsaber

Tutorial by Andy Baker, instructions by Hal Motley

This tutorial is available in the public domain.

In this tutorial you will connect a multi-colour LED to the Raspberry Pi to create a lightsaber like in *Star Wars*.

The kit

- 4 male-female GPIO jumper cables (also called DuPont cables, preferably of different colours)
- 1 breadboard (a small 5×5 breadboard works well, but any size is fine)
- 1 68 Ohm (blue grey black) resistor
- 1 4-pin colour-changing LED

Connecting the components

To setup the lightsaber, do the following (refer to the Fritzing diagram above or in the PDF):

Step 1: Connect the 4 jumper cables from the GPIO pins 13, 19, 26 and GND (last four bottom right pins, assuming the USB and Ethernet ports are to the right) to the bottom left four holes of the breadboard. I recommend trying to use the colour cables that match the diagram exactly, though any colour combination can be used.

Step 2: Connect the 68 Ohm resistor into the breadboard by placing one connector into the first hole and another into the last hole of the row above the jumper cables.



Step 3: Connect the colour-changing LED to the 4 holes on the right of the last row, make sure that the longest pin (the ground) is lined up with the black cable so the LED is properly grounded.

Step 4: Run each of the Python scripts in order, which are `lightsaber.py`, `ls1.py`, `ls2.py` and `ls3.py`. Try altering the code and see if you can make your own colour combinations.

The program

```
#!/usr/bin/env python

# Pull in the code libraries that the code will need to use.
from __future__ import division
import RPi.GPIO as GPIO
import time

# Set up the GPIO library to use the numbering of the pin on the board
# i.e. 1 - 40 of the main GPIO connector.
GPIO.setmode(GPIO.BOARD)

# Tell the code the red LED is plugged into pin 35, and that pin 35
# is an output, and set the output to low (i.e. red LED is off)
RED_LED = 33
GPIO.setup(RED_LED, GPIO.OUT)
GPIO.output(RED_LED, GPIO.LOW)

# A new pulse starts every second, and half of that time, the LED is on.
pulse_period = 1
on_fraction = 1/2

# This try and the except below allow the code to stop cleanly by
# capturing the exception from a keyboard ctrl-C.

try:
    # Store off when the while loop starts.
    start_time = time.time()
```