1. red
2. ground
3. blue
4. green

Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015

GPIO

DSI (DISPLAY)

HDMI

CSI (CAMERA)

Audio

Power

ETHERNET

USB 2x

USB 2x

fritzing

```python
#!/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()

    # Start looping around the code forever.
    while True:

        # Work out what time it is since we started, and from that, the
        # fraction of the pulse period.
        time.sleep(1/1000)
        clock_time = (time.time() - start_time) % pulse_period

        # For the LEDs, check how far through its fraction of the pulse it
        # is and turn the LED on if it's less or off if it's more.

        #============================ RED ============================#

        if clock_time < on_fraction * pulse_period:
            GPIO.output(RED_LED, GPIO.HIGH)
        else:
            GPIO.output(RED_LED, GPIO.LOW)

# The except here needs to do nothing but has to do something; the pass
# statement means do nothing.
except KeyboardInterrupt as e:
    pass

# Finally turn off the LEDs and cleanup the GPIO.
GPIO.output(RED_LED, GPIO.LOW)
GPIO.cleanup()
```