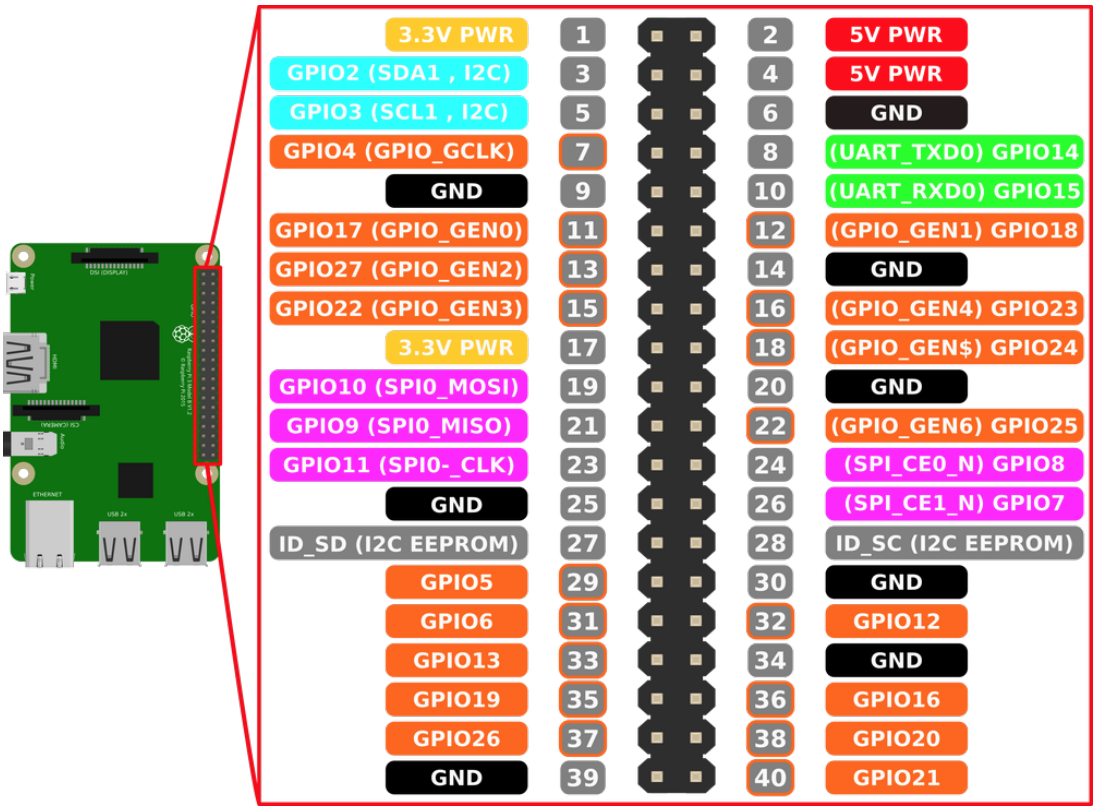
# Raspberry Pi GPIO




## Bootup GPIO States

| Pin | @startup | | Pin | @startup |
|---|---|---|---|---|
| 3v3 | - | | 5v | - |
| GPIO 2 | HIGH | | 5v | - |
| GPIO 3 | HIGH | | GND | - |
| GPIO 4 | HIGH | | GPIO 14 (TxD) | - |
| GND | - | | GPIO 15 (RxD) | - |
| GPIO 17 | LOW | | GPIO 18 | LOW |
| GPIO 27 | LOW | | GND | - |
| GPIO 22 | LOW | | GPIO 23 | LOW |
| 3v3 | - | | GPIO 24 | LOW |
| GPIO 10(MOSI) | LOW | | GND | - |
| GPIO 9(MISO) | LOW | | GPIO 25 | LOW |
| GPIO 11(SCLK) | LOW | | GPIO 8 | ~0.5v |
| GND | - | | GPIO 7 | ~0.5v |
| ID_SD | - | | ID_SC | - |
| GPIO 5 | HIGH | | GND | - |

| | | | | |
|---|---|---|---|---|
| GPIO 6 | HIGH | | GPIO 12 | LOW |
| GPIO 13 | LOW | | GND | - |
| GPIO 19 | LOW | | GPIO 16 | LOW |
| GPIO 26 | LOW | | GPIO 20 | LOW |
| GND | - | | GPIO 21 | LOW |

# Python Sample Code

> vi sample.py

**sample.py**

```
# External module imports
import RPi.GPIO as GPIO
import time

# Pin Definitons:
PIN_TRUNK_SENSOR = 27
PIN_DOOR_SENSOR  = 6
PIN_DOOR_1            = 12
PIN_DOOR_2            = 25
PIN_TRUNK_1            = 22
PIN_TRUNK_2      = 13

# Pin Setup:
GPIO.setmode(GPIO.BCM)
GPIO.setup(PIN_TRUNK_SENSOR, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(PIN_DOOR_SENSOR, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

GPIO.setup(PIN_DOOR_1, GPIO.OUT)
GPIO.setup(PIN_DOOR_2, GPIO.OUT)
GPIO.setup(PIN_TRUNK_1, GPIO.OUT)
GPIO.setup(PIN_TRUNK_2, GPIO.OUT)

# Initial state
GPIO.output(PIN_DOOR_1, GPIO.LOW)
GPIO.output(PIN_DOOR_2, GPIO.LOW)
GPIO.output(PIN_TRUNK_1, GPIO.LOW)
GPIO.output(PIN_TRUNK_2, GPIO.LOW)

doorSensorVal=False
trunkSensorVal=False

newDoorSensorVal=False
newTrunkSensorVal=False

print("Here we go! Press CTRL+C to exit")
try:
    while 1:

                #read DOOR sensors
                newDoorSensorVal=GPIO.input(PIN_DOOR_SENSOR)
                if (newDoorSensorVal != doorSensorVal):
                        doorSensorVal=newDoorSensorVal
                        if(doorSensorVal==True):
                                #lock
                                print(">DOOR SENSOR = LOW (LOCK)")
                                GPIO.output(PIN_DOOR_1, GPIO.HIGH)
                                GPIO.output(PIN_DOOR_2, GPIO.LOW)
                                time.sleep(0.075)
                                GPIO.output(PIN_DOOR_1, GPIO.LOW)
                                GPIO.output(PIN_DOOR_2, GPIO.LOW)
                        else:
                                print(">DOOR SENSOR = HIGH (UNLOCK)")
```

```
                                GPIO.output(PIN_DOOR_1, GPIO.LOW)
                                GPIO.output(PIN_DOOR_2, GPIO.HIGH)
                                time.sleep(0.075)
                                GPIO.output(PIN_DOOR_1, GPIO.LOW)
                                GPIO.output(PIN_DOOR_2, GPIO.LOW)

                #read TRUNK sensors
                newTrunkSensorVal=GPIO.input(PIN_TRUNK_SENSOR)
                if (newTrunkSensorVal != trunkSensorVal):
                        trunkSensorVal=newTrunkSensorVal
                        if(trunkSensorVal==True):
                                #lock
                                print(">TRUNK SENSOR = LOW (LOCK)")
                                GPIO.output(PIN_TRUNK_1, GPIO.HIGH)
                                GPIO.output(PIN_TRUNK_2, GPIO.LOW)
                                time.sleep(0.075)

                                GPIO.output(PIN_TRUNK_1, GPIO.LOW)
                                GPIO.output(PIN_TRUNK_2, GPIO.LOW)
                        else:
                                print(">DOOR SENSOR = HIGH (UNLOCK)")
                                GPIO.output(PIN_TRUNK_1, GPIO.LOW)
                                GPIO.output(PIN_TRUNK_2, GPIO.HIGH)
                                time.sleep(0.075)
                                GPIO.output(PIN_TRUNK_1, GPIO.LOW)
                                GPIO.output(PIN_TRUNK_2, GPIO.LOW)

except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
    GPIO.cleanup() # cleanup all GPIO
```

**Run the program**

> sudo python sample.py

# NodeJS Sample Code

The onoff library has show to be buggy! Use gpio library instead.

**Sample using "onoff" node js library**

```
var Gpio = require('onoff').Gpio;
var sleep = require('sleep');

var TRUNK_SENSOR = new Gpio(27, 'in','both');
var DOOR_SENSOR =  new Gpio(6, 'in','both');

var DOOR1 = new Gpio(8,'out');
var DOOR2 = new Gpio(7,'out');
var TRUNK1 = new Gpio(9,'out');
var TRUNK2 = new Gpio(11,'out');

DOOR_SENSOR.watch(function(err,value){
  if(err){
    console.error('There was an error', err); //output error message to console
    return;
  }

  if(value==0){
    DOOR1.writeSync(1);
    DOOR2.writeSync(0);
    sleep.sleep(10); // sleep for ten seconds
    DOOR1.writeSync(0);
    DOOR2.writeSync(0);
  }else{
    DOOR1.writeSync(0);
    DOOR2.writeSync(1);
    sleep.sleep(10); // sleep for ten seconds
    DOOR1.writeSync(0);
    DOOR2.writeSync(0);
  }
});

TRUNK_SENSOR.watch(function(err,value){
  if(err){
    console.error('There was an error', err); //output error message to console
    return;
  }
  if(value==0){
    TRUNK1.writeSync(1);
    TRUNK2.writeSync(0);
    sleep.sleep(10); // sleep for ten seconds
    TRUNK1.writeSync(0);
    TRUNK2.writeSync(0);
  }else{
    TRUNK1.writeSync(0);
    TRUNK2.writeSync(1);
    sleep.sleep(10); // sleep for ten seconds
    TRUNK1.writeSync(0);
    TRUNK2.writeSync(0);
  }
});

function unexportOnClose() { //function to run when exiting program
  TRUNK1.writeSync(0); // Turn off
  TRUNK2.writeSync(0); // Turn off
  DOOR1.writeSync(0); // Turn off
  DOOR2.writeSync(0); // Turn off
  TRUNK1.unexport(); // Unexport GPIO to free resources
  TRUNK2.unexport(); // Unexport GPIO to free resources
  DOOR1.unexport(); // Unexport GPIO to free resources
  DOOR2.unexport(); // Unexport GPIO to free resources
  TRUNK_SENSOR.unexport();
  DOOR_SENSOR.unexport();
};

process.on('SIGINT', unexportOnClose); //function to run when user closes using ctrl+c
```

**Run the Code**

> npm sample.js

# References

| Reference | URL |
| --- | --- |
| raspberrypi.org | https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/README.md |
| Sparkfun Tutorial | https://learn.sparkfun.com/tutorials/raspberry-gpio |
| Interactive Pinout Diagram | https://pinout.xyz/ |
| NodeJS - wiring-pi library | https://www.npmjs.com/package/wiring-pi |
| NodeJS - wiring-pi Documentation | https://github.com/WiringPi/WiringPi-Node/blob/master/DOCUMENTATION.md |
| NodeJS - onoff library (buggy) | https://www.npmjs.com/package/onoff<br>https://www.w3schools.com/nodejs/nodejs_raspberrypi_gpio_intro.asp |
| NodeJS - gpio library (Deprecated) | https://www.npmjs.com/package/gpio |