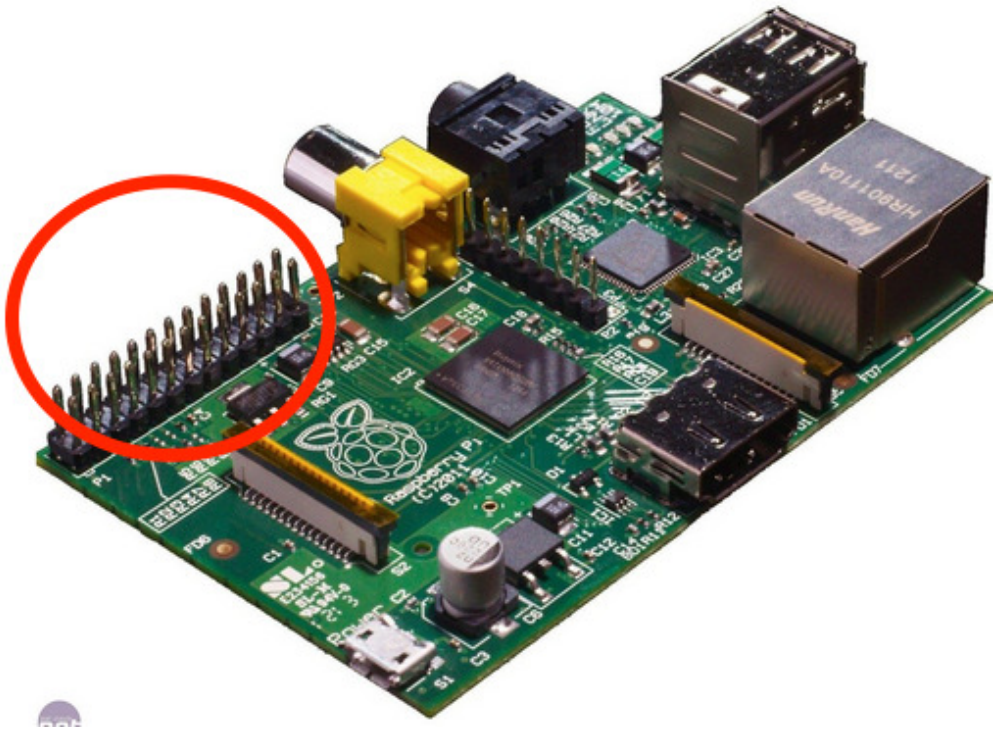
Introduction to RaspberryPi GPIO

Notebook:	iraspberrypi		
Created:	9/4/2014 8:01 PM	**Updated:**	9/5/2014 4:36 PM
Author:	Nayan Mujadiya	**Location:**	Bangalore Urban, Karnataka, I...

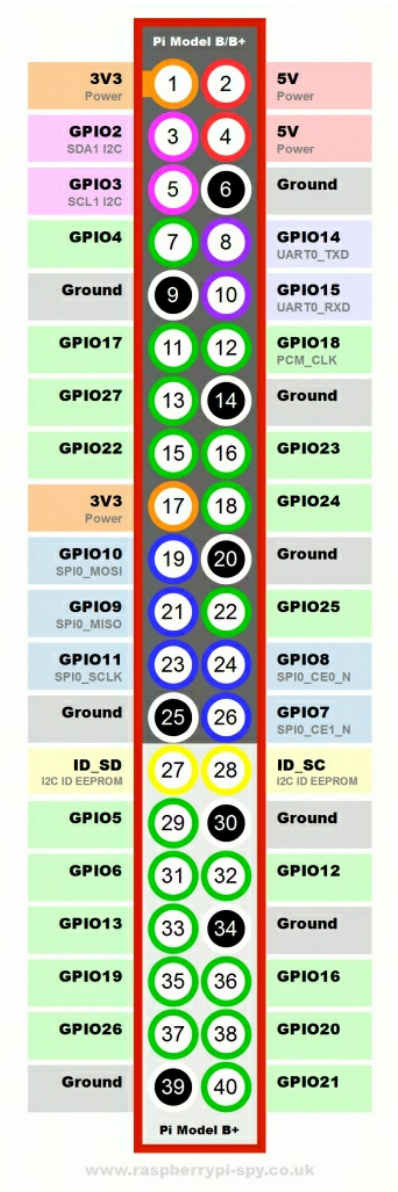
GPIO = General-purpose input/output (GPIO)

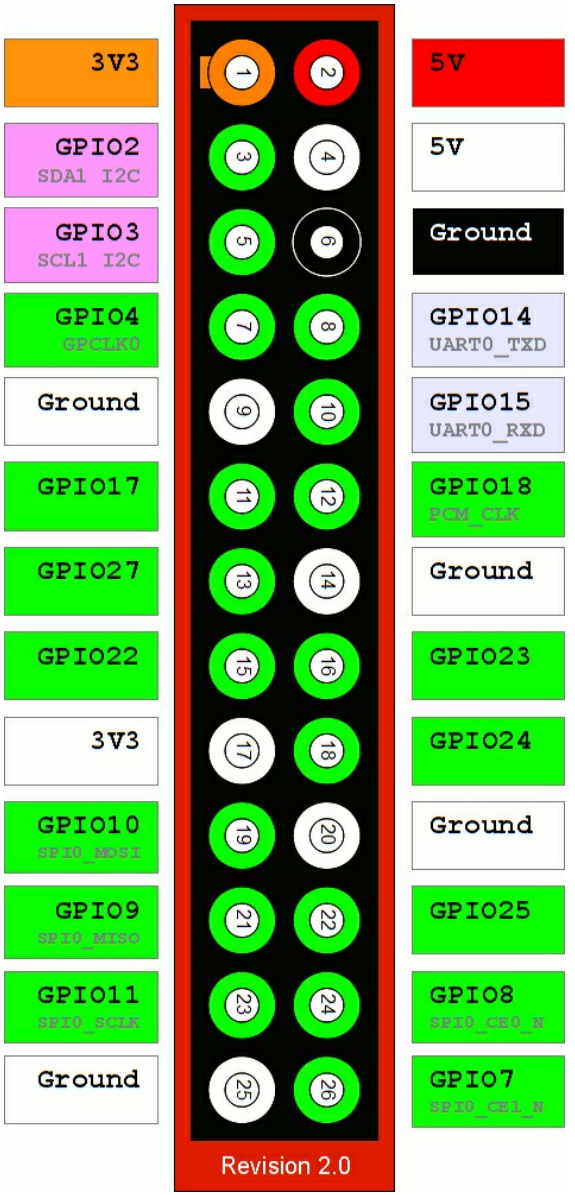


Two different Mode for GPIO.

The **GPIO.BOARD** option specifies that you are referring to the pins by the number of the pin the the plug - i.e the numbers printed on the board (e.g. P1) and in the middle of the diagrams below.

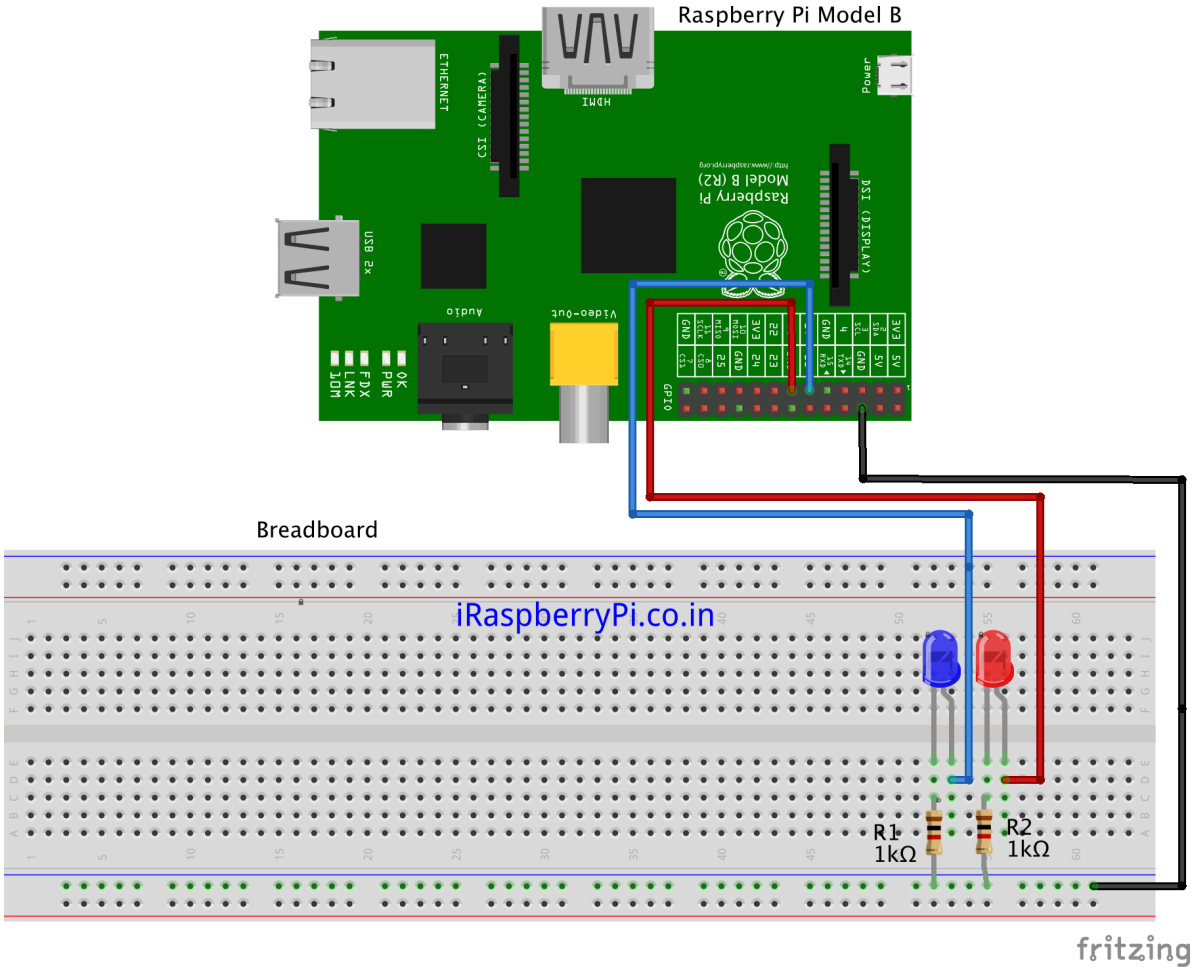
The **GPIO.BCM** option means that you are referring to the pins by the "Broadcom SOC channel" number, these are the numbers after "GPIO" in the green rectangles around the outside of the below diagrams:


Pi Model B/B+
3V3 Power
5V Power
GPIO2 SDA1 I2C
5V Power
GPIO3 SCL1 I2C
Ground
GPIO4
GPIO14 UART0_TXD
Ground
GPIO15 UART0_RXD
GPIO17
GPIO18 PCM_CLK
GPIO27
Ground
GPIO22
GPIO23
3V3 Power
GPIO24
GPIO10 SPI0_MOSI
Ground
GPIO9 SPI0_MISO
GPIO25
GPIO11 SPI0_SCLK
GPIO8 SPI0_CE0_N
Ground
GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM
ID_SC I2C ID EEPROM
GPIO5
Ground
GPIO6
GPIO12
GPIO13
Ground
GPIO19
GPIO16
GPIO26
GPIO20
Ground
GPIO21
Pi Model B+
www.raspberrypi-spy.co.uk


3V3
GPIO2 SDA1 I2C
GPIO3 SCL1 I2C
GPIO4 GPCLK0
Ground
GPIO17
GPIO27
GPIO22
3V3
GPIO10 SPI0_MOSI
GPIO9 SPI0_MISO
GPIO11 SPI0_SCLK
Ground
5V
5V
Ground
GPIO14 UART0_TXD
GPIO15 UART0_RXD
GPIO18 PCM_CLK
Ground
GPIO23
GPIO24
Ground
GPIO25
GPIO8 SPI0_CE0_N
GPIO7 SPI0_CE1_N
Revision 2.0

GPIO & LEDs

circuit diagram…



we can use GPIO 3 ways..

1) GPIO control from shell

Export the required GPIO

echo “17” > /sys/class/gpio/export

Set the direction

echo “out” > /sys/class/gpio/gpio17/direction

Set the value

echo “1” > /sys/class/gpio/gpio17/value

Configure GPIO27 as input and read its value

1. Export the required GPIO

echo “27” > /sys/class/gpio/export

2. Set the direction

echo “in” > /sys/class/gpio/gpio27/direction

3. Set the value

cat /sys/class/gpio/gpio27/value

2) through python api

```
#!/usr/bin/python
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()
GPIO.setwarnings(False)
GPIO.setup(17,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)
print "Lights on"
GPIO.output(17,GPIO.HIGH)
GPIO.output(27,GPIO.HIGH)
```

```
#!/usr/bin/python
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()
GPIO.setwarnings(False)
GPIO.setup(17,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)
print "Lights off"
GPIO.output(17,GPIO.LOW)
GPIO.output(27,GPIO.LOW)
```

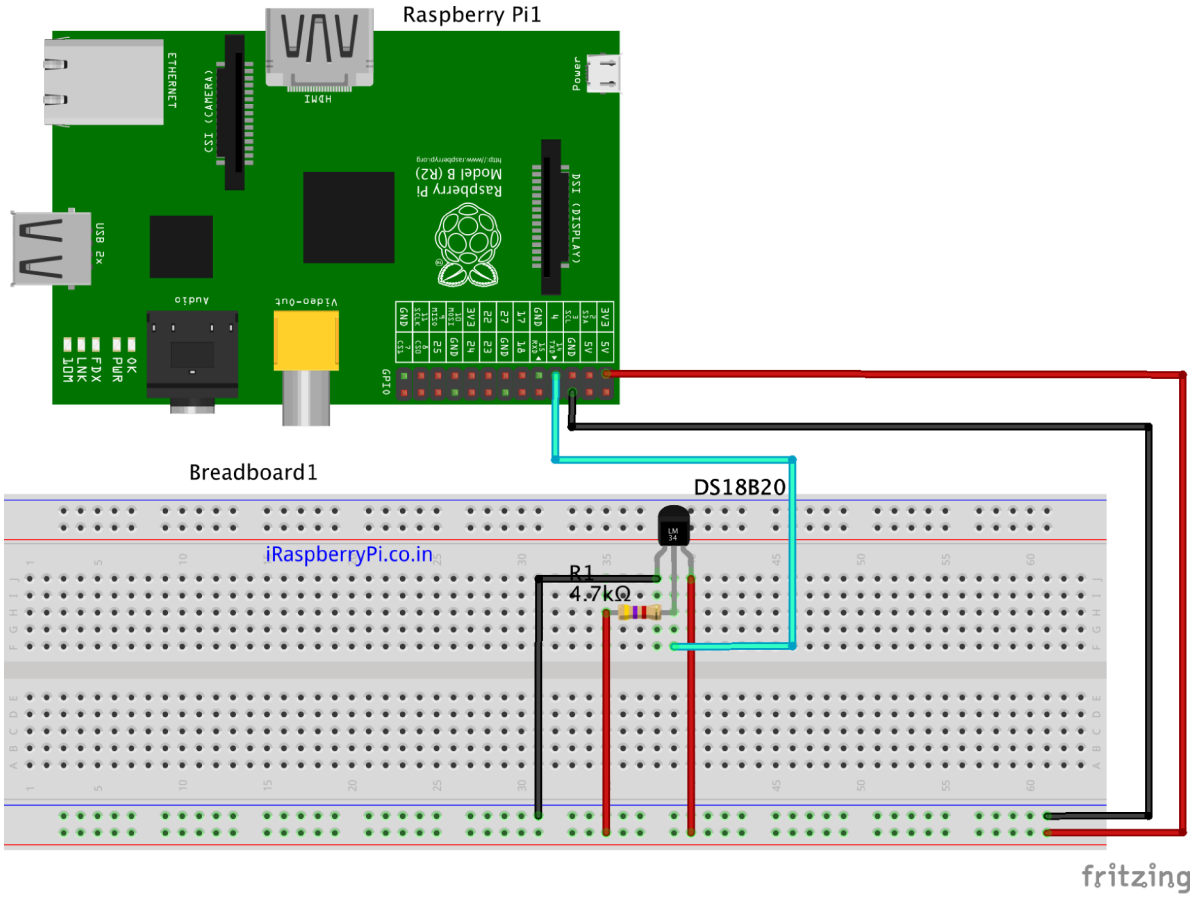
3) through web

access python script through web OR WEBIOPI

/var/www
visudo /etc/sudoers

GPIO & temperature sensor

circuit diagram...



enable 1-wire protocol

sudo modprobe w1-gpio
sudo modprobe w1-therm

go to /sys/bus/w1/devices

Python Code

```
import os
import glob
import time
#initialize the device
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'
```

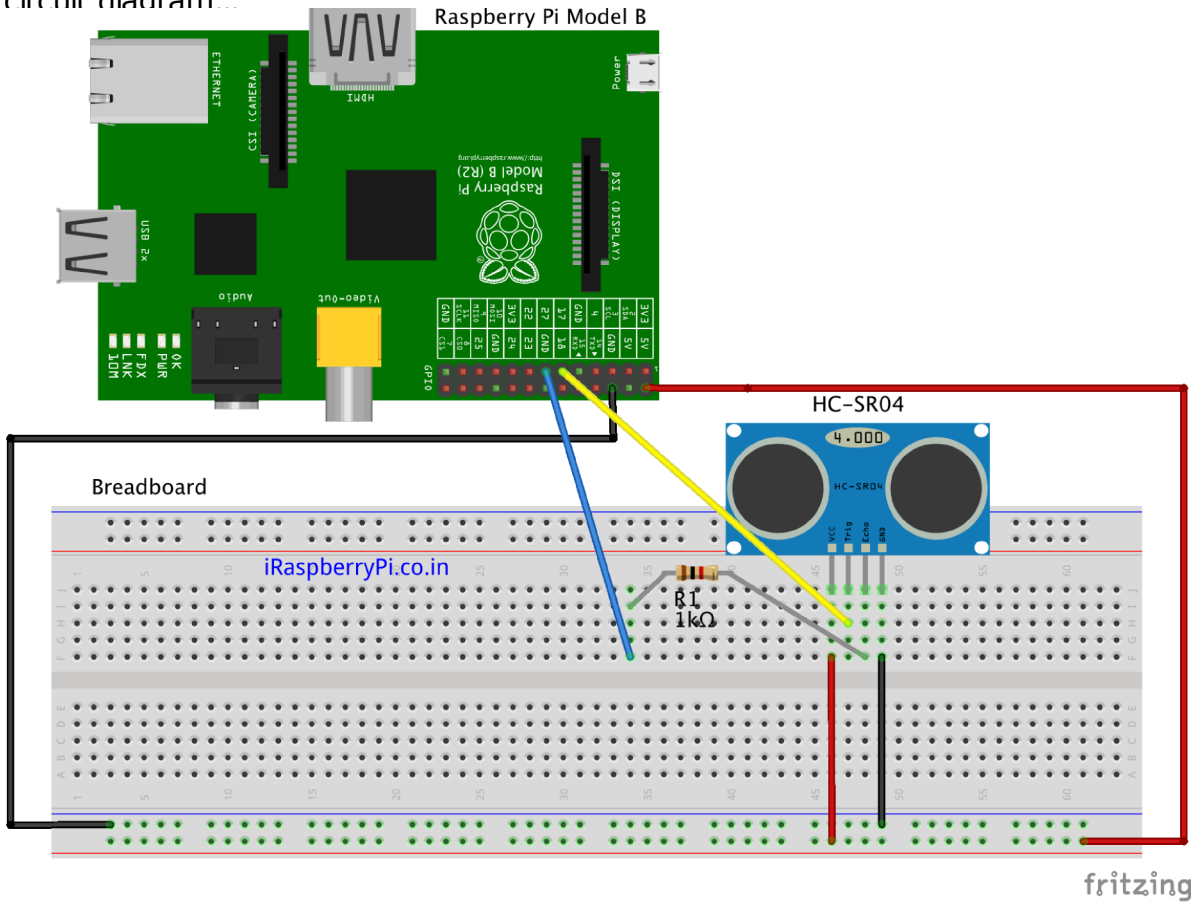
```
def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f

while True:
     print(read_temp())
     time.sleep(1)
```

GPIO & HC-SR04 Ultrasonic Distance Measuring Sensor

circuit diagram...



Now, we will use GPIO to measure distance with HC-SR04 sensor...

What is an Ultrasonic Module HC-SR04 Distance Sensor?

It's a cheap sensor that can be used to measure the distance between itself and an object in front of it by sending an ultrasonic pulse and listening for its echo. The HC-SR04 can be connected to many things including the Raspberry Pi.

Why do I need a resistor and where can I get it?

You need a resistor because the HC-SR04 sensor sends a 5v signal to a GPIO pin on the Raspberry Pi that is only rated for 3.3v. The GPIO pins on the Raspberry Pi are unprotected, which means if you were to connect the sensor without the resistor you are likely to permanently damage the GPIO pins.

How do I connect everything?

There are four pins on the HC-SR04 sensor. The pin labelled VCC requires connecting to a 5V pin, the pin labelled "Gnd" requires connecting to a ground pin, and the pins "Trig" and "Echo" need to be each wired to a unique GPIO pin on the Raspberry Pi.

You need to connect your 1K resistor between the echo pin and the GPIO pin (to protect the Raspberry Pi from receiving a 5V signal to a 3.3V pin).

source code:- http://nayaneye.co.in/rpi/usonicl.py