

# SETTING UP RASPBERRY PI

---

## Contents

Installing OS (wheezy raspbian) .....	1
Add New User to RPi .....	2
USE of LIGHT SENSOR - interfacei2clightsensor.py .....	3
USE of TEMP SENSOR - ds1820_final_1.py .....	6
USE of PIR SENSOR - PIR.py .....	6
SETUP OF WEBSERVER .....	7
SETUP OF INIT SCRIPT .....	8
SETUP OF RPI_SENSORS PROJECT .....	9
References .....	9

---

## Installing OS (wheezy raspbian)

1. Download the Wheezy Raspbian distribution from the RPi site –  
<http://www.raspberrypi.org/downloads>
2. Install the image in SD card using the "Win32 Disk Imager" tool (for Windows only).  
<http://sourceforge.net/projects/win32diskimager/>  
Or, use UNIX 'dd' tool.
3. Insert SD card into RPi
4. Attach RPi to a router, Power ON the RPi, and get its IP Address from the router's configuration page.
5. SSH to RPi (Enter Password: raspberry)

```
sudo ssh pi@<RPi-IP-ADDR>
```

### TERMINAL DUMP

```
rishav@ubuntu:~$ sudo ssh-keygen -R 192.168.1.50
/root/.ssh/known_hosts updated.
Original contents retained as /root/.ssh/known_hosts.old

rishav@ubuntu:~$ sudo ssh pi@192.168.1.50
The authenticity of host '192.168.1.50 (192.168.1.50)' can't be established.
ECDSA key fingerprint is 09:e8:d7:00:5a:f1:5c:a6:ed:7d:f3:e3:a0:f9:12:5f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.50' (ECDSA) to the list of known hosts.
pi@192.168.1.50's password:
Linux raspberrypi 3.6.11+ #474 PREEMPT Thu Jun 13 17:14:42 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

NOTICE: the software on this Raspberry Pi has not been fully configured. Please run
'sudo raspi-config'

pi@raspberrypi ~ $
```

6. Run command – raspi-config

```
sudo ssh pi@<RPi-IP-ADDR>
```

Set the following options –

- 1) Expand Filesystem  
- to enable RPi to use full amount of memory space in SD card
- 2) Advanced Options -> Hostname  
- to change the hostname of RPi

## TERMINAL DUMP

Raspberry Pi Software Configuration Tool (raspi-config)		
Setup Options		
1	Expand Filesystem	Ensures that all of the SD card s
2	Change User Password	Change password for the default u
3	Enable Boot to Desktop	Choose whether to boot into a des
4	Internationalisation Options	Set up language and regional sett
5	Enable Camera	Enable this Pi to work with the R
6	Add to Rastrack	Add this Pi to the online Raspber
7	Overclock	Configure overclocking for your P
8	Advanced Options	Configure advanced settings
9	About raspi-config	Information about this configurat
<Select>		<Finish>

## Add New User to RPi

1. Run command –

```
sudo useradd <user-name>
```

## TERMINAL DUMP

```
pi@raspberrypi ~ $ sudo adduser rishav
Adding user `rishav' ...
Adding new group `rishav' (1002) ...
Adding new user `rishav' (1001) with group `rishav' ...
Creating home directory `/home/rishav' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for rishav
Enter the new value, or press ENTER for the default

Is the information correct? [Y/n] Y
pi@raspberrypi ~ $ sudo adduser rishav
```

2. Give SUDO permissions to the new user

- edit the sudoers file

```
sudo visudo
```

- add the following line at the end

```
<user-name> ALL=(ALL) ALL
```

OR

```
<user-name> ALL=(ALL) NOPASSWD: ALL
```

## USE of LIGHT SENSOR - interfacei2clightsensor.py

1. Install required dependency – smbus module

```
sudo apt-get install python-smbus
```

### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ sudo apt-get install python-smbus
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  i2c-tools
Suggested packages:
  libi2c-dev
The following NEW packages will be installed:
  i2c-tools python-smbus
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 71.0 kB of archives.
After this operation, 317 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
```

Reference for further steps - I2C Installation for Raspberry Pi – Step by Step Guide  
<http://www.skpang.co.uk/blog/archives/575>

2. Open the following file (as ROOT)

```
/etc/modprobe.d/raspi-blacklist.conf
```

And comment both the lines

## TERMINAL DUMP

FROM

```
-----
GNU nano 2.2.6      File: /etc/modprobe.d/raspi-blacklist.conf
-----
# blacklist spi and i2c by default (many users don't need them)

blacklist spi-bcm2708
blacklist i2c-bcm2708

[ Read 4 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
-----
```

TO

```
-----
GNU nano 2.2.6      File: /etc/modprobe.d/raspi-blacklist.conf
-----
# blacklist spi and i2c by default (many users don't need them)

#blacklist spi-bcm2708
#blacklist i2c-bcm2708

[ Read 4 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
-----
```

3. Add 'i2c-dev' to the file '/etc/modules'

## TERMINAL DUMP

FROM

```
-----
GNU nano 2.2.6      File: /etc/modules
-----
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835

[ Read 9 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
-----
```

TO

```
-----
GNU nano 2.2.6      File: /etc/modules
-----
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev

[ Read 9 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
-----
```

4. Check if 'i2c-tools' is installed, if not install them

```
sudo apt-get install i2c-tools
```

### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ sudo apt-get install i2c-tools
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
i2c-tools set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
rishav@rishav-rpi ~ $ sudo apt-get install i2c-tools
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
i2c-tools set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
rishav@rishav-rpi ~ $ sudo apt-get install i2c-tools
```

5. add current user to i2c group, then, reboot the RPi

```
sudo adduser rishav i2c
```

### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ sudo adduser rishav i2c
```

```
Adding user `rishav' to group `i2c' ...
Adding user rishav to group i2c
Done.
```

```
rishav@rishav-rpi ~ $ sudo reboot
```

```
Broadcast message from root@rishav-rpi (pts/0) (Wed Nov 20 04:22:17 2013):
The system is going down for reboot NOW!
```

6. After reboot, check if i2c works or not, using the following command (1 for Rev 2 Board)

```
i2cdetect -y 1
```

### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ i2cdetect -y 1
```

```
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- 29 -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

7. Run the script - **interfacei2clightsensor.py**  
(present in the project folder)

#### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ python interfacei2clightsensor.py
reading sensors
Lux Val is:
0
reading sensors
Lux Val is:
0
```

## USE of TEMP SENSOR - ds1820\_final\_1.py

1. Just run the script '**ds1820\_final\_1.py**' as root  
(present in the project folder)

#### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ sudo python ds1820_final_1.py
17.125
17.187
17.187
```

## USE of PIR SENSOR - PIR.py

1. Just run the script '**PIR.py**' as root  
(present in the project folder)

#### TERMINAL DUMP

```
rishav@rishav-rpi ~ $ sudo python PIR.py
('PIR Value : ', 0)
('PIR Value : ', 0)
('PIR Value : ', 0)
```

## SETUP OF WEBSERVER

1. For installing Apache Server,

```
sudo apt-get install apache2
```

2. For Starting Apache Server,

```
sudo service apache2 [start|stop|restart]
```

3. Installing PHP

```
sudo apt-get install php5 libapache2-mod-php5
```

4. Enable use of root permissions by Webserver for,

- a. RESTART command
- b. run PYTHON script as root

Add the following lines to the '/etc/sudoers' file

```
www-data ALL=NOPASSWD: /usr/bin/python, /sbin/reboot
```

Other commands can also be added (separated by comma)

**NOTE:** Use full path for the command  
To get path of the command: **which <command-name>**

5. To add password to the server files –

- a. Create a passwords file –

### TERMINAL DUMP

```
rishav@ubuntu:~$ htpasswd -c /home/rishav/rpi-sensors/passwords admin
New password:
Re-type new password:
Adding password for user admin
```

- b. Add the following lines to the file **apache.conf** or **httpd.conf** (/etc/apache2/httpd.conf) –

```
<Directory /var/www/>
    AuthType Basic
    AuthName "RPi Configuration Restricted Access"
    AuthUserFile <password-file-from-step-1>
    Require user admin
</Directory>
```



## 6. Restart Apache server

```
sudo apache2ctl restart
```

## SETUP OF INIT SCRIPT

To start the program at startup

1. Create a script in '/etc/init.d/'

### TERMINAL DUMP

```
rishav@rishav-rpi /etc/init.d $  
rishav@rishav-rpi /etc/init.d $ sudo nano sensors  
rishav@rishav-rpi /etc/init.d $  
  
-----  
GNU nano 2.2.6 File: sensors  
  
#!/bin/sh  
# /etc/init.d/sensors  
  
case "$1" in  
  start)  
    echo "Starting Sensors Monitoring"  
    # run application you want to start  
    python /home/rishav/rpi-sensors/main.py &  
    ;;  
  stop)  
    echo "You will have to stop it manually"  
    # kill application you want to stop  
    ;;  
  *)  
    echo "Usage: /etc/init.d/sensors {start|stop}"  
    exit 1  
    ;;  
esac  
  
[ Read 20 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell  
-----
```

2. Give execute rights to the file

```
sudo chmod 755 sensors
```

3. Set the script to run at boot

```
sudo update-rc.d sensors defaults
```

## TERMINAL DUMP

```
rishav@rishav-rpi /etc/init.d $  
rishav@rishav-rpi /etc/init.d $ sudo update-rc.d sensors defaults  
update-rc.d: using dependency based boot sequencing  
insserv: warning: script 'sensors' missing LSB tags and overrides  
rishav@rishav-rpi /etc/init.d $
```

## SETUP OF RPI\_SENSORS PROJECT

1. Get the project on the RPi by cloning the following repository –  
<https://github.com/amarjeet-iitd/IITD-Pervasive>
2. Set the path to project folder (rpi\_sensors) which contains the main.py file in the following files –
  - config.php – in ServerPages
  - config.py – in rpi\_sensors/profile\_io
  - /etc/init.d/sensors – the init script created in above steps
3. Copy the files in ServerPages to folder /var/www
4. Update the init script using the above steps.
5. Go to the RPi's IP Address via Internet Browser, to open the configuration page.
6. Update the configuration and then reboot the RPi.
7. The program should load at startup and running  
(can check using 'ps aux', if the script is running or not)

## References

For further help –

1. So you got a Raspberry Pi: now what?  
- <http://www.engadget.com/2012/09/04/raspberry-pi-getting-started-guide-how-to/>
2. I2C Installation for Raspberry Pi – Step by Step Guide  
- <http://www.skpang.co.uk/blog/archives/575>
3. How to use GPIOs on raspberry pi (Simple I/O, PWM and UART)  
- <https://sites.google.com/site/semilleroadt/raspberry-pi-tutorials/gpio>
4. Start python script from init.d  
- <http://www.pietervanos.net/knowledge/start-python-script-from-init-d/>
5. Tutorial – Install Apache, PHP and MySQL on Raspberry Pi  
- <http://www.dingleberrypi.com/2012/09/tutorial-install-apache-php-and-mysql-on-raspberry-pi/>
6. Authentication and Authorization (Apache Server)  
- <http://httpd.apache.org/docs/2.2/howto/auth.html>