

### Step 1: Download Operation System to SD card

Use SD association's formatting tool

[https://www.sdcard.org/downloads/formatter\\_4/eula\\_windows/](https://www.sdcard.org/downloads/formatter_4/eula_windows/)

Install and run the formatting tool

Set "FORMAT SIZE ADJUSTMENT" option to "NO" in the "Options" menu

Download operation system NOOBS and unzip it on to the SD card for the Raspberry Pi.

<http://www.raspberrypi.org/downloads>

### Step 2: Install Operation System

Start up the Raspberry Pi and install the **Raspbian** system.

### Step 3: Login / Change Password (Optional)

Login the Raspberry Pi [user ID: pi] and [password: raspberry]

The default password can be change type:

```
sudo passwd [enter] [type your new password]
```

### Step 4: Enable SSH for remote access

```
sudo apt-get update
```

 update and upgrade the system.

```
sudo apt-get upgrade
```

```
sudo raspi-config
```

 open the system configuration.

Enable SSH for PUTTY to remote control the Raspberry Pi

OR

```
sudo mv /boot/boot_enable_ssh.rc /boot/boot.rc
```

 second way to enable SSH

```
sudo reboot
```

 reboot system to apply changes.

```
sudo shutdown -h now
```

 or safely power down the system.

```
ifconfig
```

 check the Raspberry Pi network IP address under **eth0**, start with **inet addr**:

Now we should be able to use PUTTY to login the Pi with the pervious IP address.

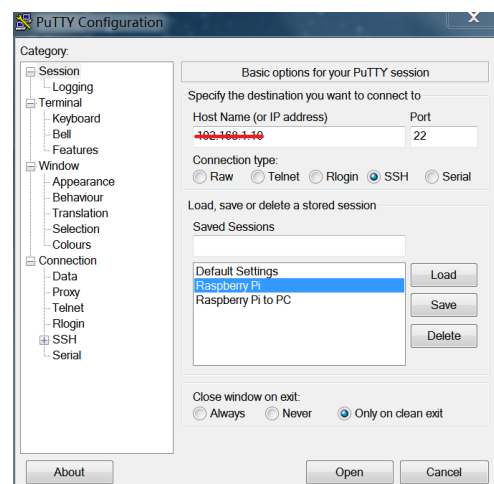


Figure 2 PUTTY Login interface

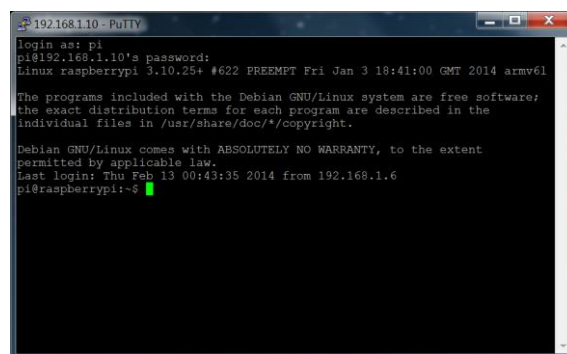


Figure 1 PUTTY's Raspberry Pi Common Line interface

### Step 5: WIFI configuration (optional)

```
startx
```

 turn the Raspberry Pi into graph user interface

### Step 6: Remapping the keyboard (optional)

If different letters appear on-screen from that which you typed, then remap is needed.

```
sudo dpkg-reconfigure keyboard-configuration
```

 follow the prompts.

OR

```
sudo nano /etc/default/keyboard
```

 find **XKBLAYOUT="gb"** change gb to us

```
sudo reboot
```

 reboot system to apply changes.

### Step 7: Installing the web server

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

 install apache and PHP

If error occurs, do the following steps:

```
sudo groupadd www-data
```

```
sudo usermod -g www-data www-data
```

DON'T give RPi's a fix IP address!

```
sudo service apache2 restart
```

 re-start the apache2 service

We could test it by entering the IP address of the Pi to a browser. It will shows "It Works!"

```
sudo nano /etc/apache2/sites-enabled/000-default
```

 change 2<sup>nd</sup> AllowOverride **None** to **All**

Press CTRL + X then Y then Enter to save the changes.

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

 install MySQL

While installing MySQL, it will requests user to enter a new password for root user. Recommend to use the same password as the PRi login password.

### Step 8: Install Python-Serial / GPIO connection

To use the Raspberry Pi's serial port, first disable getty (the program that displays login screen)

```
sudo nano /etc/inittab
```

 comment # T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

To prevents the Raspberry Pi from sending out data to the serial ports when it boots

```
sudo nano /boot/cmdline.txt
```

 remove "console=ttyAMA0, 115200 kgdboc=ttyAMA0, 115200"

```
sudo apt-get install minicom
```

 install minicom

Load the program code into Moteino MCU and connect it to Pi's GPIO pins

\*Baud rate: ensure the baud you set inside the Arduino code is matched to Raspberry Pi's code.

\*Arduino code: whatever Serial.print() or Serial.println() data will send to serial communication.

\*Unlike the Moteino, Arduino UNO's serial pin is held at 5 volt and the Raspberry Pi's at 3.3 volts.

Therefore a Logic Level Converter or voltage divider would be required.



Figure 1: Pi's GPIO

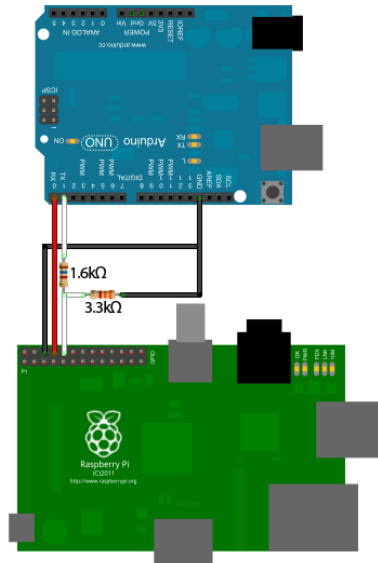


Figure 2: Voltage Divider

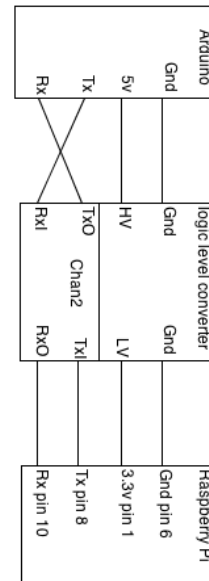


Figure 3: Logic Level Converter

```
sudo apt-get install python-serial
```

`nano xxxx.py` where the xxxx is the name of the Python file you are creating

`python xxxx.py` run the python script (have to run this every time the Raspberry Pi reboot)

\*The python script will stop running after you exit PUTTY. If you want to keep the python script runs continuously, please see the instruction located at the end of this document.

Press CTRL + C to exit/stop a running python script

`sudo nano xxxx.py` edit a saved python script

### Step 9: Setup FTP server

`sudo chown -R pi /var/www` take the ownership of this server directory

`sudo apt-get install vsftpd`

`sudo nano /etc/vsftpd.conf` change the following statements:

anonymous enable=YES to NO

Uncomment **local enable=YES** (uncomment by deleting the #)

Uncomment **write enable=YES**

force\_dot\_files=YES (add this line to the end of the file)

Press CTRL + X then Y then Enter to save the changes.

`sudo service vsftpd restart` re-starting the service

`sudo passwd root` change the root user password (Recommend to set it as same as normal Pi's)

`exit` exit the PUTTY

Reopen the PUTTY again and login as **ROOT** user:

Login as: root

Password: [whatever you just set]

`nano /etc/passwd` find the line begin with "pi:x:1000:1000..." comment it with #

Press CTRL + X then Y then Enter to save the changes.

`usermod -d /var/www pi`

If error occurred: 'pi' does not exist, do the following steps:

`nano /etc/passwd` uncomment the "pi" line we just did

`usermod -d /var/www pi` should shows an error process code

`kill process xxxx` where the xxxx is the process code

`exit` exit the PUTTY

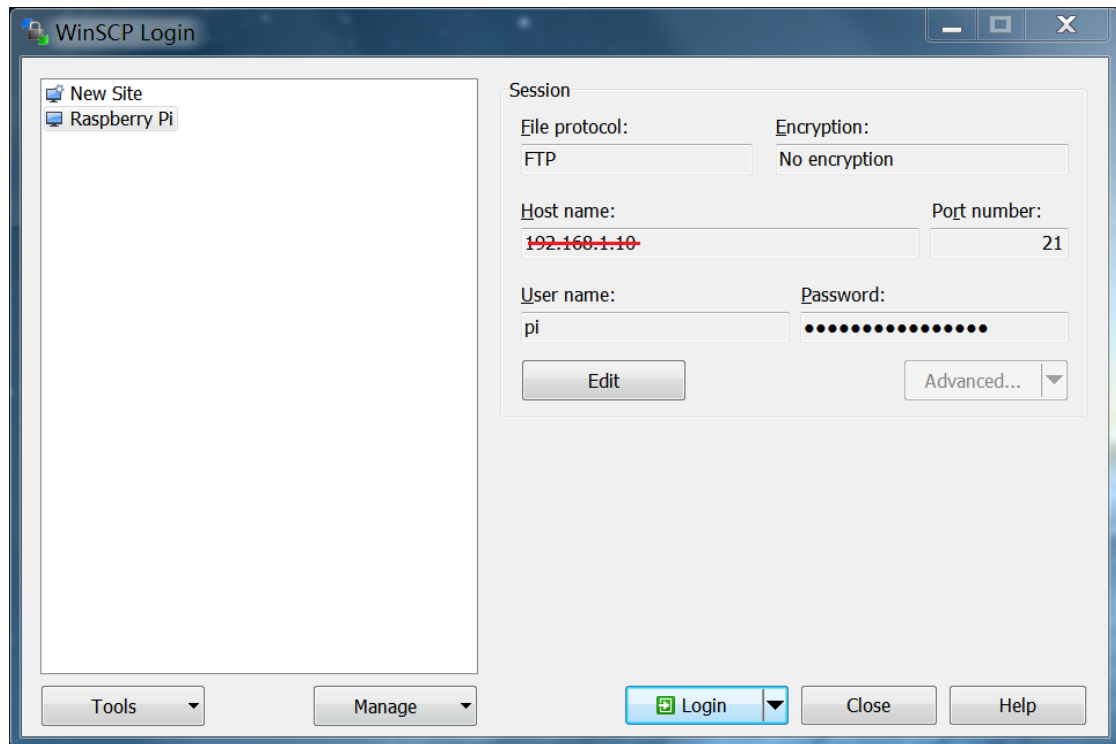
Reopen the PUTTY again and login as **NORMAL** pi user.

`sudo usermod -L root`

`exit` exit the PUTTY

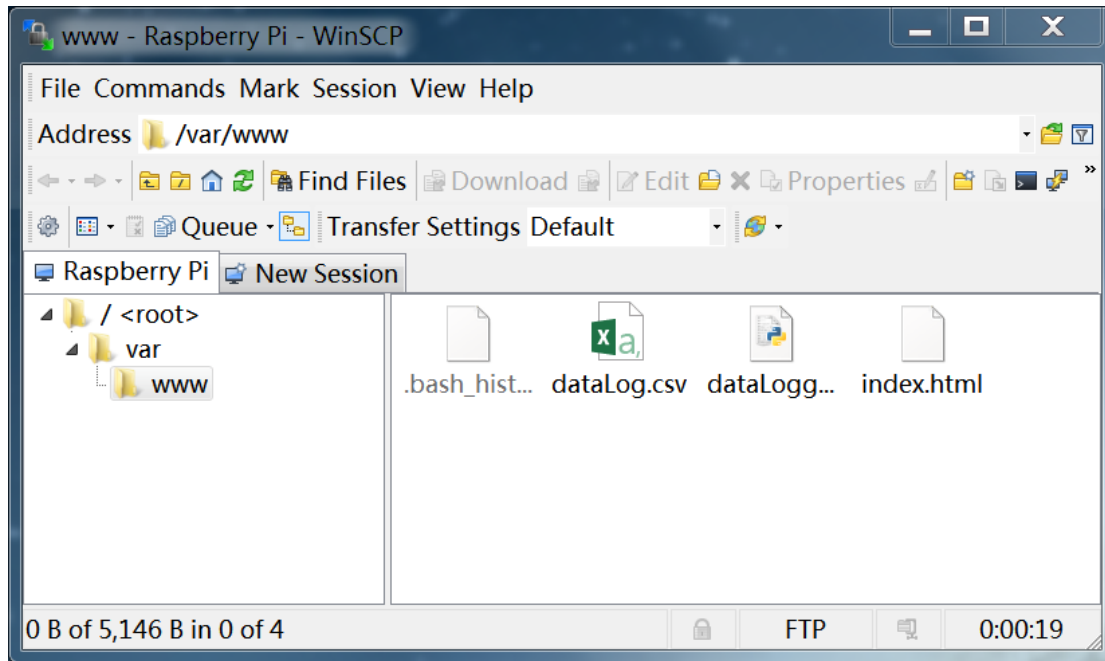
### Step 10: WinSCP for FTP login

Download a FTP software named WinSCP to login into the Raspberry Pi



Host name is the IP address of the Raspberry Pi we found from [Step 4].

Not sure it is using the ROOT user or NORMAL user password, give a try!



The python code will save all the data from serial communication into a CSV (or other format) file which located inside the var/www/ directory, as well as the python code itself. We could download or delete the files from the right click menu.

### Connect Raspberry Pi to PC without Router (ETHERNET ONLY)

```
sudo nano /boot/cmdline.txt
```

Add the **ip=x.x.x.x** to the end of the line (ensure you do not add any extra lines)

1) RPi to PC only, without changing anything on PC:

In this case **ip=169.254.0.2** (or any between 169.254.0.0 – 169.254.255.254)

Press CTRL + X then Y then Enter to save the changes.

Plug in the Ethernet cable between the Pi and PC

```
sudo reboot
```

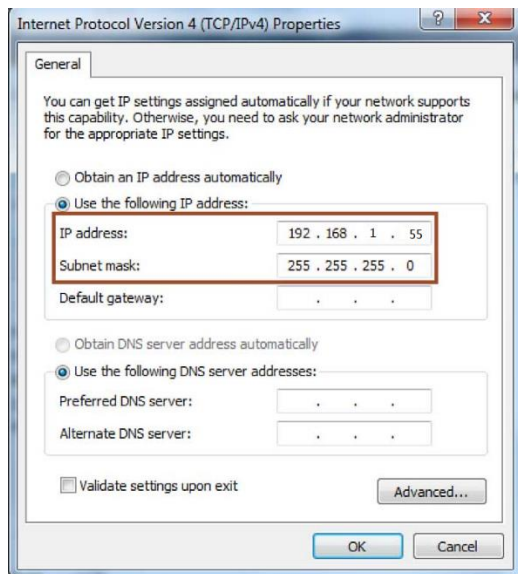
 reboot system to apply changes.

\*Be sure to comment out the ip=x.x.x.x before attach the Pi back to router.

2) RPi to PC or Router, needs to change PC's network IPv4: (Recommended method)

For network settings where the IP address is fixed. Use an address which matches the PC except the last one. For example PC is 192.168.1.105, then use **ip=192.168.1.155**

Under PC's IPv4, give an IP address to the PC (for example: 192.168.1.55 and subnet 255.255.255.0)



### Check attached USB Device Port

`lsusb` list all attached ports of USB device  
`cd /dev/` go to system's device hierarchy  
`ls -la tty*`  
`sudo fdisk -l` list storage devices  
`tail -f /var/log/messages` plug and unplug to see USB device's info  
`cd` exit the /dev/ directory

### Mount/Unmounts Storage device

`cd /dev/`  
`sudo fdisk -l` look for the device's sda# info (sda1, sda2...sdb1, sdb2...etc)  
`cd /mnt`  
`sudo mkdir 4GB_USB` make a directory for the device, can be any name (no space)  
`sudo mount -t vfat -o uid=pi,gid=pi /dev/sda# /mnt/4GB_USB` mount the device (sda# of yours)  
Now we can access the storage device  
`cd /mnt/4GB_USB` go the device's directory we just created  
`ls -lap` list all files inside the storage device  
\*Unmounts the device before unplug from RPi  
`cd /mnt` go to mount directory  
`sudo umount /mnt/4GB_USB` ensure it is **unmount** NOT unmount.  
`sudo umount /dev/sda#` Or this method to un-mount (sda# of yours)  
\*umount error: device is busy.

Might be there is client (WinSCP) connected to the RPi and viewing the device's directory.  
Disconnect the client and try to unmount again

Might be a python program is associated with this device's directory. Change the directory of this program to something else and run the program for once. After these, your storage device could be safely remove from the raspberry pi.

\*We have to repeat the mount procedures for every time we plug in the device again.

## Run a python script at boot using Cron (exit PUTTY without stopping the python)

`cat /var/www/dataLogger.py`      make sure the you got the correct path  
`sudo crontab -e`      add a new cron job, add the following line to the bottom  
`@reboot python /var/www/dataLogger.py &`      start your python script at boot

Press CTRL + X then Y then Enter to save the changes.

`sudo reboot`      reboot system to apply changes.

To stop (kill) the auto run python script:

`ps aux | grep /var/www/dataLogger.py`      find out the process number and kill it.

For example you will see, `root 1863 0.0 1.0.....python /var/www/dataLogger.py`

`sudo kill 1863`      1863 is the example process number, it might vary on your device.

## Change file's permissions:

`ls -l /var/www/`      list the permissions of all files in this directory.

```
pi@raspberrypi:~$ ls -l /var/www/
total 72
drwxr-xr-x 2 pi pi 4096 Feb 13 05:23 Desktop
-rw-r--r-- 1 root root 5625 Feb 16 17:42 dataLog.csv
-rw-r--r-- 1 pi pi 559 Feb 15 19:47 dataLogger.py
-rw-r--r-- 1 pi root 858 Feb 17 06:01 index.html
-rw----- 1 pi pi 0 Feb 15 09:50 nohup.out
-rw-r--r-- 1 pi pi 42744 Feb 17 05:03 untpic.jpg
-rw-r--r-- 1 pi pi 956 Feb 17 06:35 weather.html
drwxrwxrwx 2 pi pi 4096 Feb 17 06:31 webserver
```

d rwx rwx rwx  
| | |  
| | | others  
| | group  
| user  
Is a directory

u = owner of the file (user)

g = groups owner (group)

o = anyone else on the system (other)

+ = add permission

- = remove permission

r = read permission

w = write permission

x = execute permission

`chmod g+r /var/www/weather.html`      add read permission to the group

`chmod o-w /var/www/weather.html`      remove write permission from the other

## Set time and date to RPi

Without internet access:

`sudo date 030109152013`      March 01th, 09:15am, 2013

With internet access:

`sudo apt-get install ntpdate`      install ntpdate

`sudo raspi-config`      change to a correct time zone

`sudo ntpdate ntp.pool.org`      get date and time from internet

**References:**

Raspberry Pi (RPi) Beginners:

[http://elinux.org/RPi\\_Beginners](http://elinux.org/RPi_Beginners)

Startup setting:

<http://www.youtube.com/watch?v=H1jSudsIJfA>

Download system:

<http://www.raspberrypi.org/downloads>

GPIO Serial connection:

<http://blog.oscarliang.net/raspberry-pi-and-arduino-connected-serial-gpio/>

<http://www.benk.ca/node/10>

Python:

<http://www.classthink.com/2013/09/15/getting-started-python-raspberry-pi/>

<http://www.python.org/doc/QuickRef.html>

FTP web server:

[http://tinkernut.com/wiki/page/Episode\\_320](http://tinkernut.com/wiki/page/Episode_320)

<http://www.youtube.com/watch?v=WgcNBjIJNYs#t=25>

Connect Pi to PC directly with Ethernet cable:

<http://pihw.wordpress.com/guides/direct-network-connection/in-a-nut-shell-direct-network-connection/>

Adding USB Drives (USB Flash/Hard Drives):

<http://www.youtube.com/watch?v=TfqKCrMtFAs>

[http://elinux.org/RPi\\_Adding\\_USB\\_Drives](http://elinux.org/RPi_Adding_USB_Drives)

RPi basics learning the shell:

[http://linuxcommand.org/learning\\_the\\_shell.php](http://linuxcommand.org/learning_the_shell.php)

[http://elinux.org/CLI\\_Spells](http://elinux.org/CLI_Spells)

Run a python script at boot using Cron:

<http://www.raspberrypi-spy.co.uk/2013/07/running-a-python-script-at-boot-using-cron/>

Raspberry Web Server:

<http://raspberrypiwebserver.com/>

Linux basics (file permissions):

<http://raspberrypiwebserver.com/linux-basics/>

<http://www.penguintutor.com/raspberrypi/file-permissions-reference>