Setting Up of Raspberry Pi

SP3175: The Earth

1 Getting to know the Raspberry Pi

The Raspberry Pi is a credit-sized device that plugs into a computer monitor, together with a standard keyboard and mouse. This device is capable of doing almost everything a normal computer can do such as browsing the Internet and programming using Python.

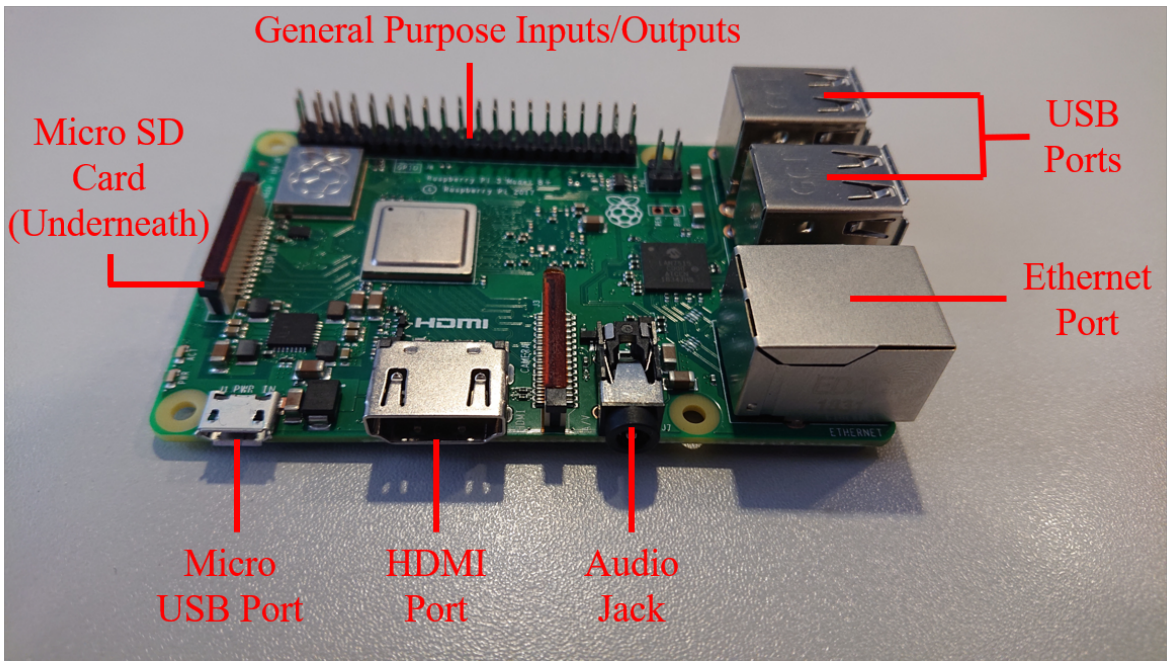

Figure 1: Components on a Raspberry Pi 3 Model B+.

The Raspberry Pi has the following components (Figure 1):

1. **Micro SD card slot:** The micro SD card slot is used to insert the SD card, which stores the Raspbian operating system and additional files.
2. **General purpose inputs/outputs (GPIO):** Allows for connections to be made to other electrical components. This will be used to connect the carbon dioxide sensor.
3. **USB ports:** These ports are used for connection to the mouse and keyboard. It can also be used for connections to a USB drive.
4. **Ethernet port:** The Ethernet port can be used to connect the Raspberry Pi to a network.
5. **Audio jack:** Used for connecting headphones/speakers. (Would not be used here)
6. **HDMI port:** This will be used to connect to the monitor you are currently using to display the output from the Raspberry Pi.
7. **Micro USB port:** This port is used to connect to the power supply.

2 Setting up the Raspberry Pi

2.1 Connecting the Raspberry Pi

The following steps must be followed when connecting the Raspberry Pi:

1. Insert the micro-SD card into the micro-SD slot underneath the Raspberry Pi.
2. Connect the mouse using one of the USB ports of the Raspberry Pi.
3. Then, connect the keyboard in a similar way.
4. Ensure the monitor is plugged into a power socket and turned on. With the help of the HDMI cable, connect the monitor to the Raspberry Pi.
5. Plug the power supply cable into a power socket and connect it to the Raspberry Pi using the micro-USB port.
6. One should see a blinking red light on the Raspberry Pi and raspberry icons on the monitor screen.

2.2 Booting up for the first time

When the Raspberry Pi is turned on for the first time, it automatically loads the application. To begin the setup, click **Next**.

The application will prompt you for your location. Set the country as **Singapore**, language as **English** and the timezone at local Singapore time (Figure 2). Then click **Next**.

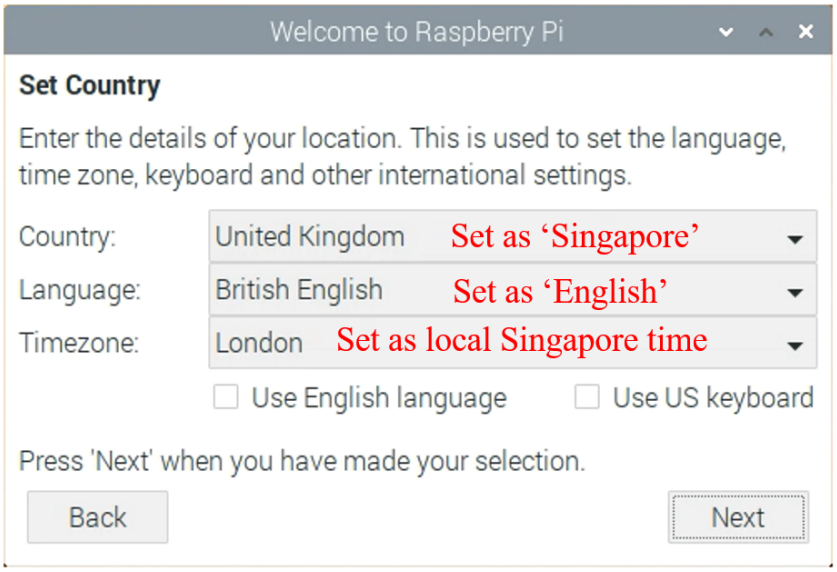

Figure 2: Set-up of location.

The next window will prompt you to set-up a new password for the Raspberry Pi (Figure 3). Click **Next** after performing this step.

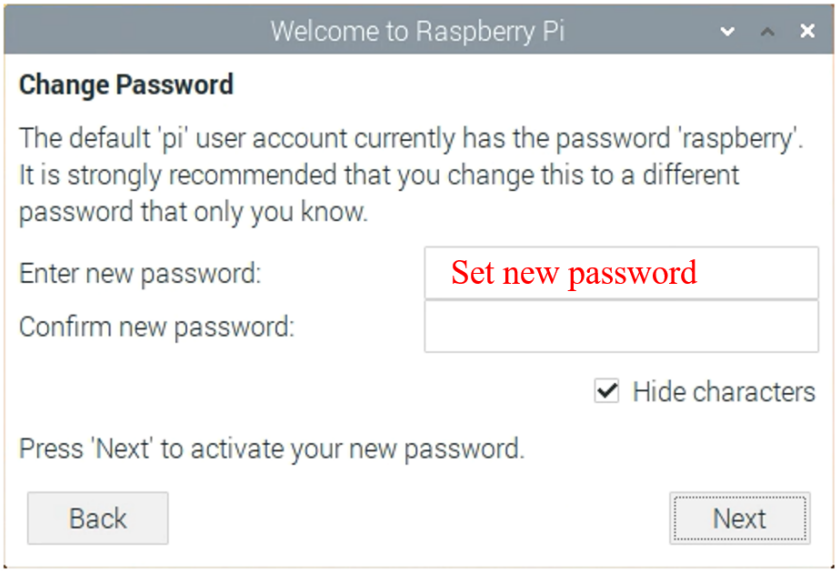

Figure 3: Set-up of new user password.

The Raspberry Pi operating system unfortunately does not allow you to connect to NUS_STU Wi-Fi by conventional means. Please refer to the next section for establishing Wi-Fi connection. For now, click on **Skip**. Allow Installation Wizard to check for any updates and install them. (Note that this may take longer than expected.) After the installation of updates (if any), click on **Restart** to reboot the system for the new settings to take effect (Figure 4).

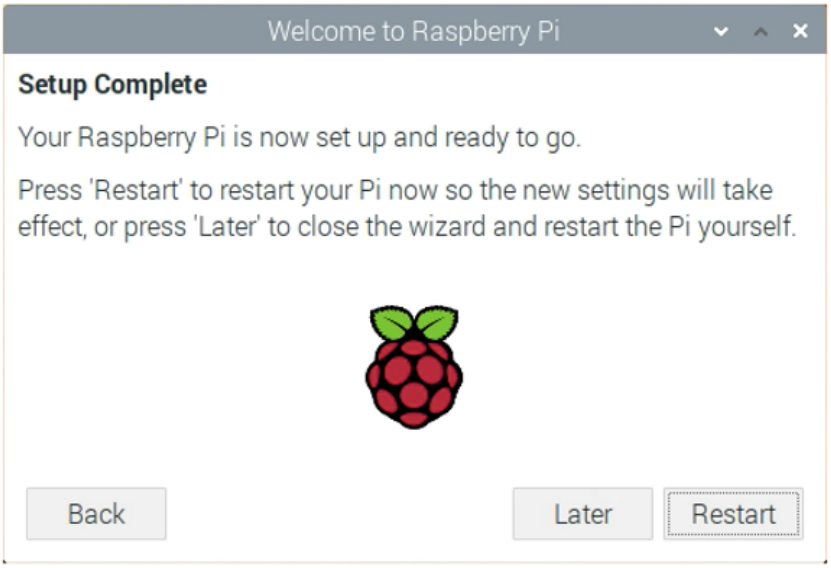

Figure 4: Completion of set-up.

2.3 Connecting to NUS_STU WiFi Network

To connect the Raspberry Pi to the NUS_STU Wi-Fi network, run the `wifi.sh` script (which will be given to you) on the command prompt using the following command:

```
bash wifi.sh
```

When running the `wifi.sh` file, it will prompt you for your NUS email address and password. Enter those into the command prompt. After which, the system will reboot again.

3 CO_2 Sensor

The CO_2 sensor used for this experiment is the K30 10,000ppm CO_2 Sensor manufactured by the company CO_2 meter. You may find specifications and related documentation on their website.

3.1 Connecting the CO_2 Sensor

The components needed to connect the Raspberry Pi to the CO_2 sensor are as follows:

1. Raspberry Pi
2. CO_2 sensor
3. One 4-way, 2.54mm pitch spacing terminal block
4. Four male to female jumper wires (Connecting Wires)
5. Screw driver

Use a terminal block (Figure 5 Left) to connect the connecting wires to the CO_2 sensor. Each of the connecting wires used for set-up has a male and female lead (Figure 5 Right).

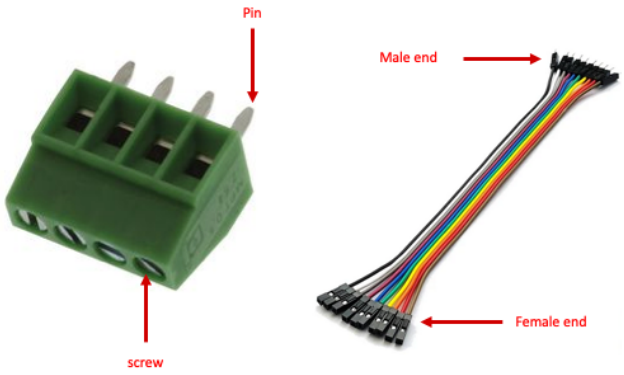

Figure 5: (Left) Terminal block displayed with its screws and pins. (Right) Male to female jumper connecting wires.

The screws on the terminal block can be loosened with the help of the screwdriver. Open the screw of each port on the terminal block by screwing in the anti-clockwise direction. Insert the male end of a connecting wire into the port and close the screw by screwing in the clockwise direction. This is shown in Figure 6. Ensure the wire does not move when it is secured to the terminal block.

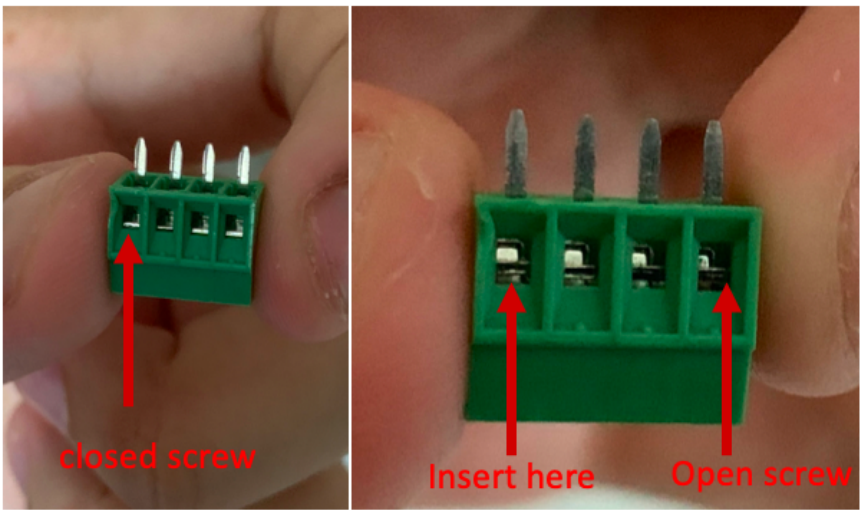

Figure 6: Image displaying the open and close screw positions of a terminal block.

The female ends of the connecting wires are to be connected to their respective pins on the Raspberry Pi as shown in Figure 7 below.

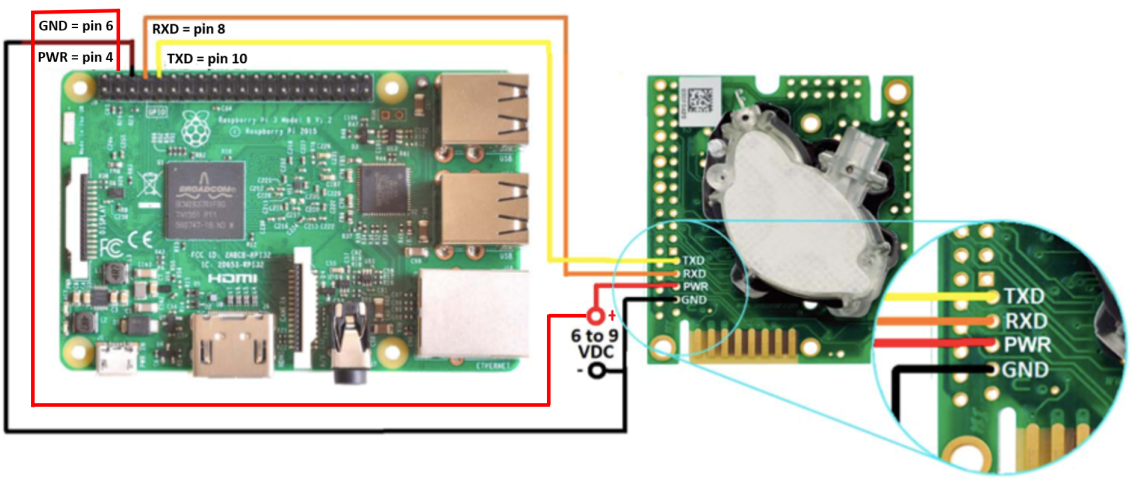

Figure 7: Circuit connection between CO_2 sensor and the Raspberry Pi.

A blinking light should be seen on the sensor if the connection is successful when the power supply is switched on. The final setup is shown in Figure 8.

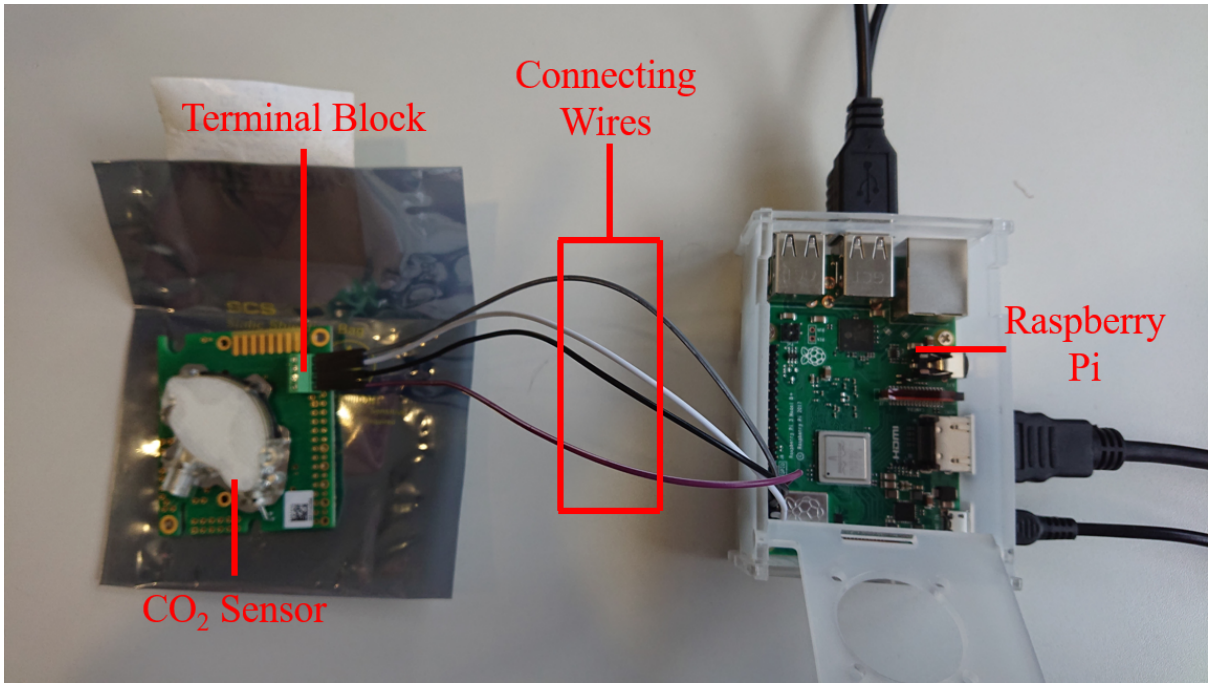

Figure 8: CO_2 Sensor connected to the Raspberry Pi.

3.2 Making Measurements on the CO_2 Sensor

The CO_2 sensor uses the Serial port on the Raspberry Pi to transfer data from the sensor to the Raspberry Pi. In order to use this Serial port, it must be activated. This can be done using following the next steps:

1. Open the terminal window and run `sudo raspi-config`. This opens up the Raspberry Pi configuration settings.
2. Select **Interfacing Options** and then select **Serial**.
3. Specify **Serial Console** as **No** and enable **Serial Port** hardware by selecting **Yes**.
4. Lastly, save the settings by selecting **Finish** and reboot the system.

To make measurements using the CO_2 sensor, make use of the code in `sensor.py` file to write your code in the `measurements.py` file.

3.3 Calibration of CO_2 Sensor

Enclose the sensor inside a sealed chamber filled with pure Nitrogen gas. Wait for 5-10 minutes for the chamber to saturate. Ideally, measurements made using the sensor outdoors should

register readings between 370ppm and 430 ppm.

3.4 Making CO_2 Measurements on Raspberry Pi Boot up

The `crontab` command line tool is used to ensure the python script automatically runs on the Raspberry Pi boot up. The following command can be run on the terminal:

```
crontab -e
```

If this is the first time you are running `crontab`, you can choose `nano` as the text editor. After which the following line of code can be added to this document,

```
@reboot sudo python /path/to/file/measurements.py
```

3.5 Connecting the Real-time Clock to the Raspberry Pi

The Raspberry Pi updates the time from the internet. However, when internet connection is not available, the time can be recorded using a real-time clock. The DS3231 is a real-time clock (RTC) that can be connected to the Raspberry Pi. The DS3231 can be connected to the Raspberry Pi with the help of the female-female connecting wires as shown in figure 9. In addition, the coin-sized battery must be inserted on the bottom of the RTC to ensure it is functional when the Raspberry Pi is turned off.

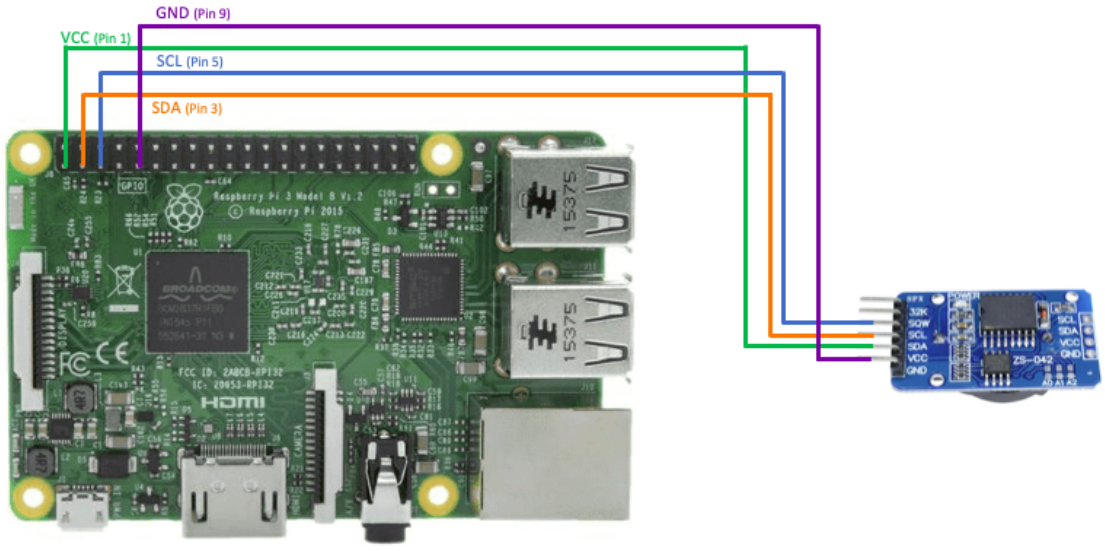

Figure 9: Circuit connection between RTC and Raspberry Pi

The DS3231 uses the I2C interface to communicate between the Raspberry Pi and itself. To enable the I2C interface on the Raspberry Pi, the following steps are to be performed:

1. Open the terminal window and run `sudo raspi-config`. This opens up the Raspberry Pi configuration settings.

2. Select **Interfacing Options** and then select **I2C**.

3. Enable the I2C interface by selecting **Yes**.

4. Lastly, save the settings by selecting **Finish** and reboot the system.

After enabling the I2C interface on the Raspberry Pi, the next step is to program the Raspberry Pi to use the time stored on the RTC. This involves a number of steps:

1. Run the following command on the terminal:
   `sudo nano /boot/config.txt`

2. This should open an editor window, and then add the following line at the end of the file:
   `dtoverlay=i2c-rtc,ds3231`
   Save the file, and reboot the system.

3. Run `sudo nano /lib/udev/hwclock-set` and comment the following lines, with a `#` in front of the line.
   `# if [ -e /run/systemd/system ]; thenexit 0fi`
   `# /sbin/hwclock --rtc-$dev --systz --badyear`
   `# /sbin/hwclock --rtc-$dev --systz` This can be seen in figure 10.

Lastly, the time must be synced from the Raspberry Pi to the RTC. Perform the following steps:

1. Run `sudo hwclock -D -r` to check that the time on the RTC is accurate.

2. Update the system's date by the following command: `sudo date -s "2020-02-04 22:17:30"`

3. Sync the RTC time with the Raspberry Pi time using the following command: `sudo hwclock -w`

4. The RTC time can be checked by running: `sudo hwclock -r`

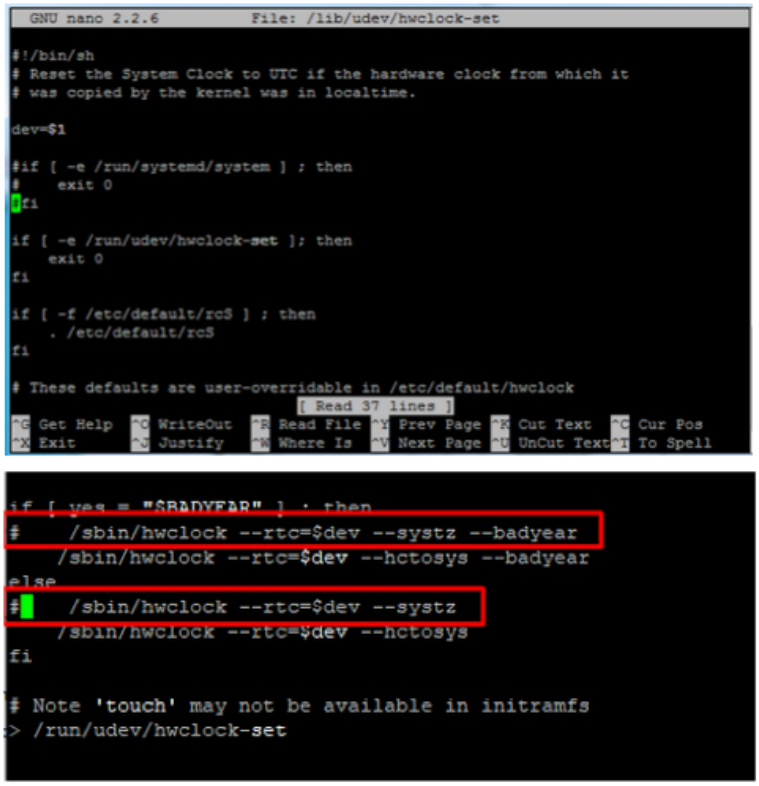

Figure 10: The lines to be commented out in step 4 of setting up the RTC.

4 Setting up Raspberry Pi for Remote Access

4.1 Secure Shell Login (SSH)

To enable Secure Shell Login (SSH) service on the Raspberry Pi, run the `ssh.sh` script on the command prompt using the following command:

```
bash ssh.sh
```

The output of this command will be an I.P. address that can be used to establish remote access for the Raspberry Pi.

4.1.1 Windows Users

For Windows users, the SSH service have to be used together with the Putty application. Launch Putty to open the Putty configuration set-up (Figure 11).

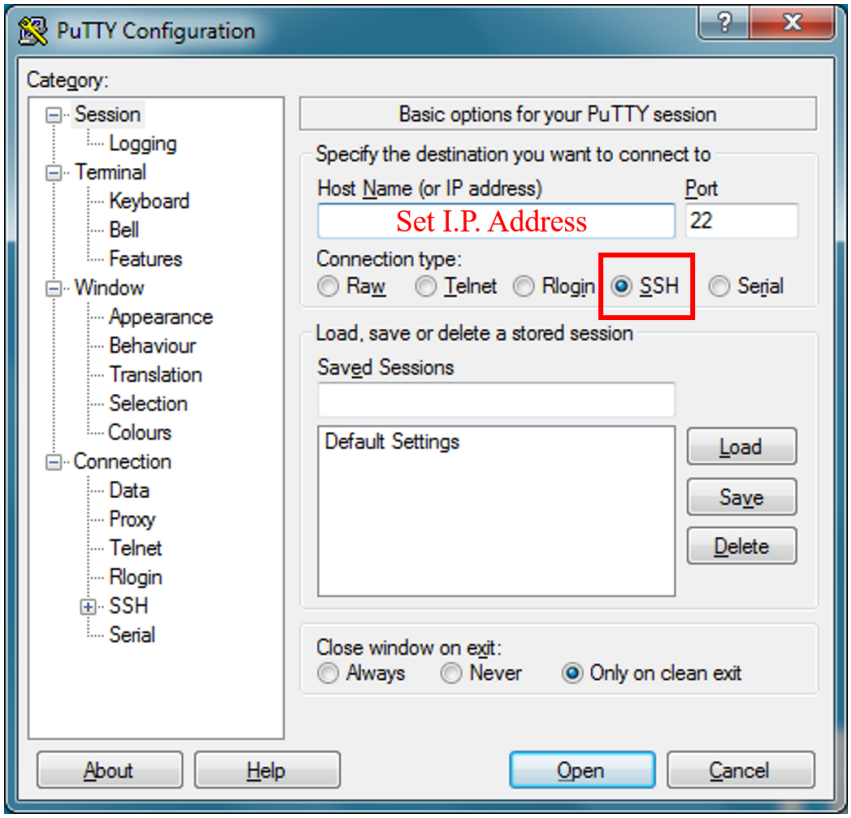

Figure 11: Putty configuration set-up.

After launching Putty application, key the Raspberry Pi I.P. address as the host-name. The port is to be set to 22. Ensure that SSH is selected before clicking **Open**. A security warning may be shown when the connection is secured. Click **Yes** to continue. You will only see this warning when Putty connects to the Raspberry Pi it has not seen before. Follow the on-screen instructions to connect to the Raspberry Pi remotely. If everything is done successfully, you should see the Raspberry Pi prompt on your computer's command prompt, identical to one on Raspberry Pi.

4.1.2 MacOS and Linux Users

Thankfully, MacOS and Linux users do not need any additional installations. One can launch the terminal window, and enter the following command to connect the Raspberry Pi remotely:

```
ssh pi@172.17.176.127
```

* Replace the I.P. address with the I.P. address obtained from your Raspberry Pi. Once again, you should see the Raspberry Pi prompt on your computer's command prompt.