ISSN 2319–7757 EISSN 2319–7765

Indian Journal of Engineering

Design of a Low Cost Weather Monitoring Station using Raspberry Pi

Shyamala C[1], Divya PA[2], Ifrah Naaz[3]

1. Assistant Professor, Department of Electronics and Communication Engineering, GSSSIETW, Mysuru-17, Email: shyamalac@gsss.edu.in
2. Department of Electronics and Communication Engineering, GSSSIETW, Mysuru-17, Email: divya.arkanath@gmail.com
3. Department of Electronics and Communication Engineering, GSSSIETW, Mysuru-17, Email: ifrahnaaz16@gmail.com

Publication History

Received: 07 November 2015

Accepted: 15 December 2015

Published: January-March 2016

Citation

Shyamala C, Divya PA, Ifrah Naaz. Design of a Low Cost Weather Monitoring Station using Raspberry Pi. *Indian Journal of Engineering*, 2016, 13(31), 51-58

ABSTRACT

Weather affects a wide range of activities. So monitoring weather and environment is very crucial. This project is based on the concept of using the Raspberry Pi as an interfacing device as well as a web server. The aim of this project is to design a weather-monitoring station that can take measurements of the temperature, rain, wind velocity and direction, humidity and save them on a server so that this data can be accessed from anywhere via the Internet. This system is designed using a Raspberry Pi. Every home, school or workplace has its own microclimate, so by taking measurements, unique local data is generated.

Keywords: Raspberry Pi, Raspbian, micro-weather, web server, anemometer.

1. INTRODUCTION

Sensing the winds and weather has been important to man over the centuries. Today, the winds and other weather variables are of equal concern and can have an even greater impact on our modern, high-tech life style. Weather affects a wide range of man's activities, including agriculture, transportation and leisure time. A weather station is that facility on land or sea, which has instruments and devices for observing and measuring atmospheric parameters to provide the information for weather forecasts[1.]. Modern weather monitoring systems and networks are designed to make the measurements necessary to track these movements in a cost effective manner[2.][3.][4.].

The aim of this project is to design a weather-monitoring station that can take measurements of the temperature, air pressure, wind velocity and direction, humidity and save them on a server so that this data may be accessed from anywhere via the Internet. The Raspberry Pi is a low cost, credit-card sized single board computer that has the ability to interact with the outside world by interfacing with various types of sensors. The Raspberry Pi has a number of features such as an ARMv7 processor, GPU, RAM, SD card slot, USB port etc. It's cheap, small and rugged, and it needs a small power supply.

The data obtained from this system can then be used for various purposes such as automated irrigation systems, automated temperature control for homes, offices, warehouses and factories, green house climate control, for tracking hazardous materials released into the air is the Biological Identification and Detection System (BIDS), pollution monitoring and many more such applications.

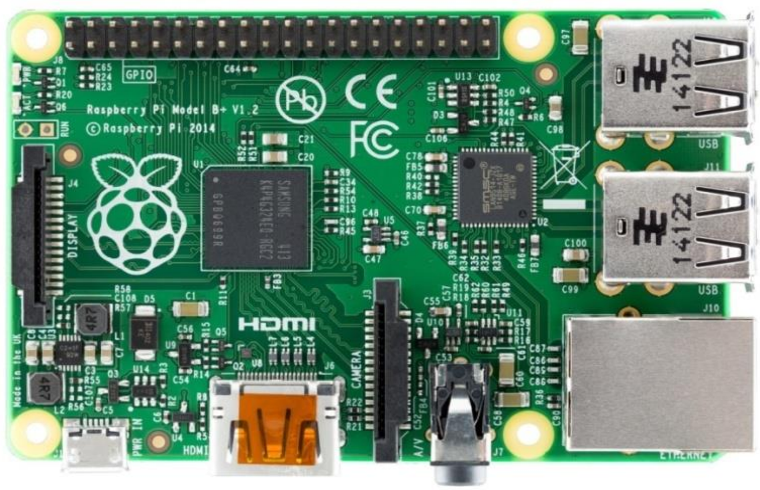


Figure 1.1 Raspberry Pi model B+

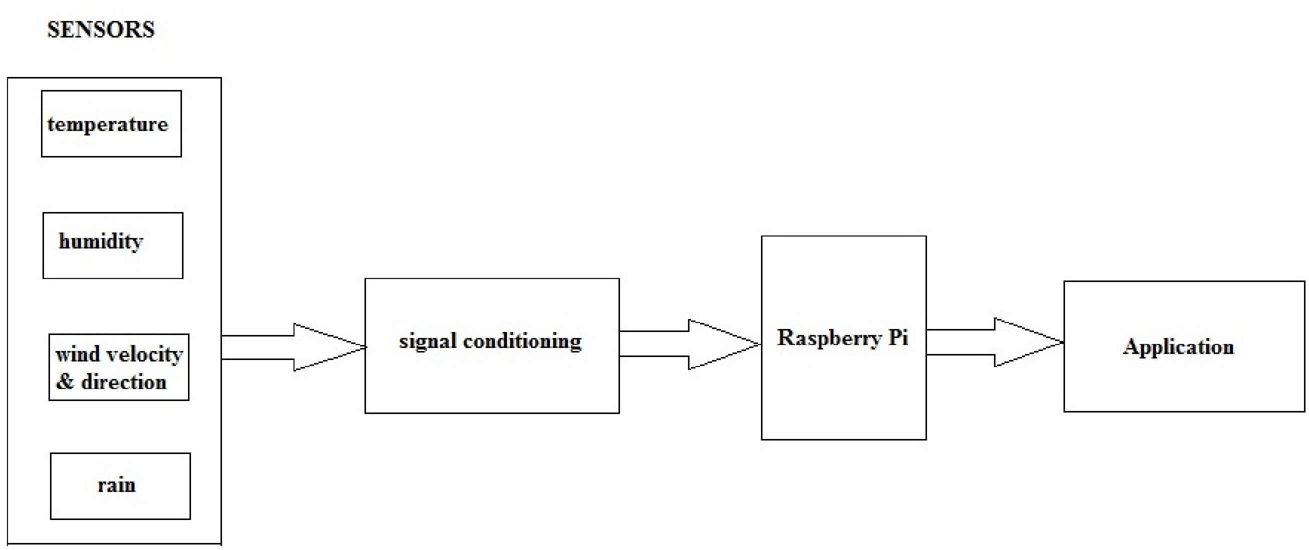


Figure 2.1 Block Diagram

2. DESIGN METHODOLOGY

The Raspberry Pi the heart of the weather monitoring system. It is used as a controlling device as well as a database to store the weather information and used as a server. Different sensors are interfaced with the Pi. The micro-weather of a local area is obtained directly with the help of these sensors or with measuring tools built using the sensors. The block diagram is shown in figure 2.1. The five different entities measured using RPi weather monitoring station are:

- Temperature
- Humidity
- Wind velocity
- Wind direction
- Rainfall

The readings of various sensors and measuring tools are given to the RPi through signal conditioners. These may be ADC, drivers, switching transistors etc. The signal conditioners provide the required 0-3.3V digital values to RPi.

3. SOFTWARE & HARDWARE

3.1 Software

A. Loading the OS on Pi :

The OS most suitable for this project is the Raspbian Wheezy. Raspbian is the most popular RPi operating system because it is easy to use for a Linux beginner. It gives the user a functional desktop. Using Xfce as the desktop environment means the RPI's resources are kept well in hand, and not wasted on inefficient eye-candy. It is an easy-to-use distro that can be set up relatively quickly.

The major advantage of Raspbian over the other distros is the selection of educational and teaching material included on the distro. It also supports all the recommended over clocking limits, the Raspberry Pi camera and any future official hardware add-ons as well. A benefit of using Raspbian is that a lot of tutorials, projects and third-party hardware run off are based on it as a standard. It makes it easier to learn coding the distro is quite fast and light. It is essentially the default Linux distribution and this status has given Raspbian a lot of advantages over everything else while still being as flexible as Linux can be[5.].

Win32dsk:

For loading the Raspbian on the Pi Win32dsk is used. This is an image burner that installs the OS on the SD card directly from windows. Even if there is a ready-prepared SD card with the Raspberry Pi, sooner or later a newer, better version of the operating system (OS) will be released. Unless we know how to make a bootable SD card from the downloaded OS image file, we won't be able to take advantage of the improvements.

Steps involved:

1. Insert the SD card
2. (re)Start Win32Diskimager.
3. Choose the drive you want to copy the image to (in our case E:).
4. Then click on the folder icon and choose the unzipped .img file from earlier that you want to put on the SD card.
5. Then click Write, to write the Operating system on the card from the .img file.

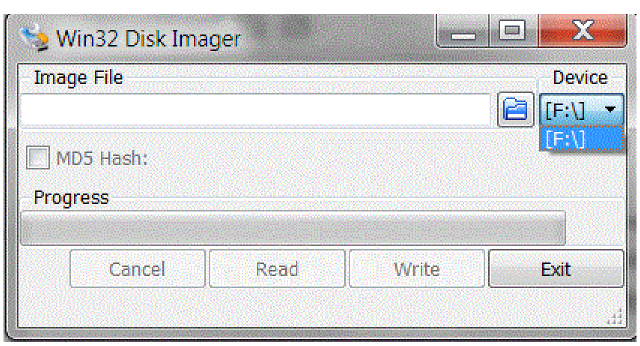


Figure 3.1.1 Win32dsk Confirming

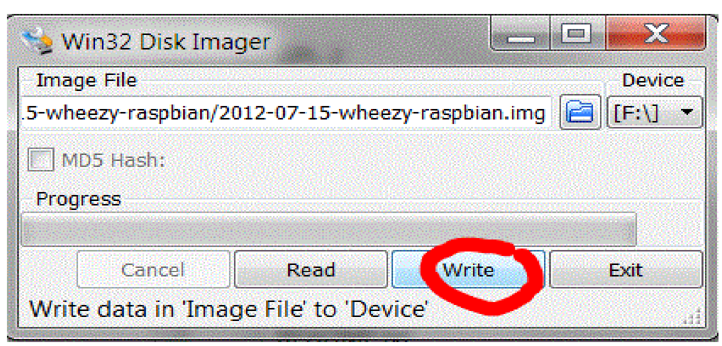


Figure 3.1.2 Win32dsk write page

You will then be asked to confirm. Check carefully that you are writing to the correct device and if so, click yes. When it's finished, eject the card reader and remove the SD card. Then insert in the Raspberry Pi. The Pi will boot up (start) into the new operating system. It takes a couple of minutes[10.].

B. RPi initial configurations:

- Update: Initial booting process requires the updation of softwares in Raspberry Pi using the internet. The steps involved are as follows.
- Type the following command in LX Terminal and press Enter: **sudo raspi-config**
- Select Advanced Options and select Update.
- Select Finish to get back to the Terminal screen.
- Install software packages: The command used to install a software package on the Pi is **sudo apt-get install –name of the software;** Here, **Sudo** allows a permitted user to execute command as the super user or another user, as specified in the sudo user's file. Advanced Packaging Tool (APT) command is used for the installation of new software packages, upgrade of existing software packages etc.

The various software packages installed are

- Apache web server : This is used to configure the RPi as a web server. It is used to host a number of websites
- Python : To interface the various sensors with the Raspberry Pi, Python is used.
- MySQL and PHP : To create a database for the Raspberry Pi and upload the values on a web page.

C. Enabling the SSH:

Enabling the Secure shell(SSH) makes it possible to operate the RPi through another laptop connected to the same network as the Pi. This avoids the use of a monitor and keyboard thus making it more convenient. For this to work, the Putty software package needs to be available on the laptop[11.]. The SSH enable option is available under the raspberry Pi configuration advance settings.

3.2 Hardware

A. Temperature and Humidity sensing

The sensor used to measure the temperature and humidity is the DHT11 sensor. This sensor features a temperature & humidity sensor complex with a calibrated digital signal output. It makes use of the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology and ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component and provides a calibrated digital signal output[9.][12.][13.].

The connections of the sensor with the Pi are as shown in the figure 3.2.1. VCC of DHT11 Module is connected to Raspberry Pi pin 1 (3.3V), GND to pin 6 (gnd) and DATA(s) to pin 7 (GPIO4).

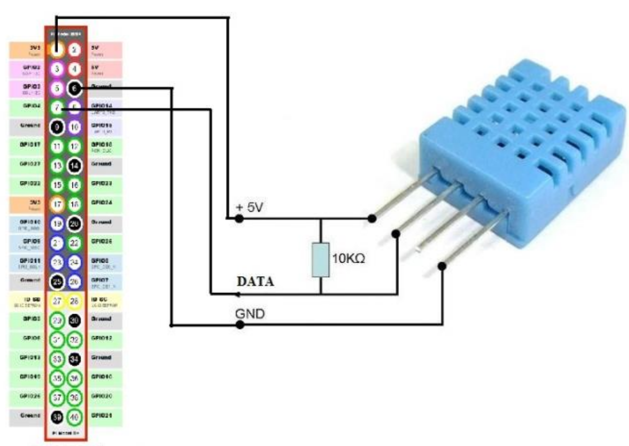


Figure 3.2.1 Circuit diagram of DHT11

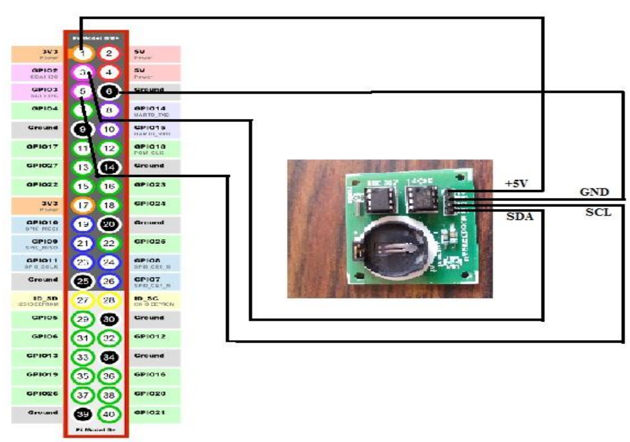


Figure 3.2.2 circuit diagram of RTC

The interfacing of DHT11 with Raspberry Pi is done using a python code[6.]. The temperature and humidity is measured every 2s. The readings are displayed for temperature in both ^{0}C and ^{0}F, humidity in percentage.

B. Real Time Clock

The Raspberry Pi does not have a on board real time clock. It reads the time from the Internet. But in case of a failure in the internet connection, the Pi is unable to access the time. Use of a hardware real time clock (RTC) allows the Raspberry Pi to be used without internet access and still provide services that use time stamps. DS1307 is a commonly available I2C based RTC IC. Using the I2C protocol supported by the Raspberry Pi, real time clock support can easily be added. The connections of the RTC with the RPi is shown in figure 3.2.2.

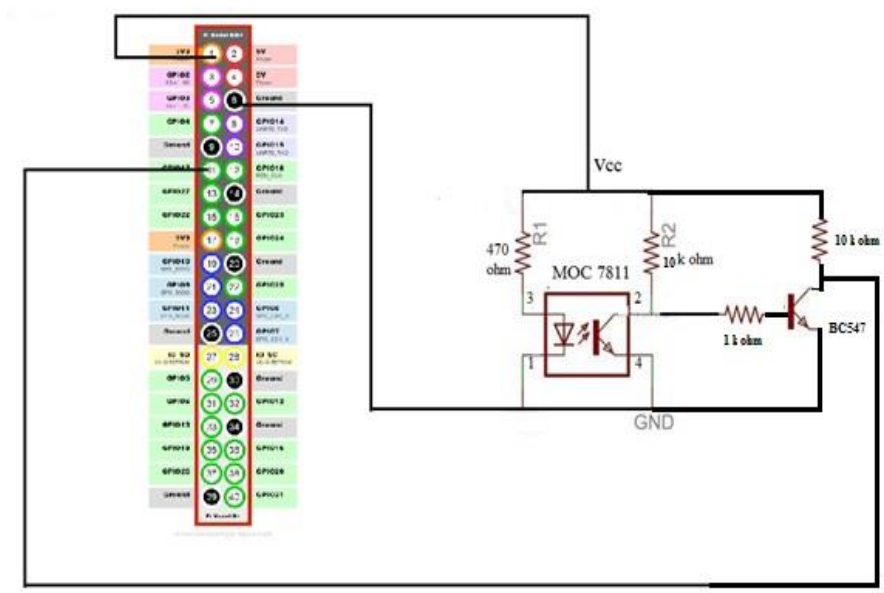
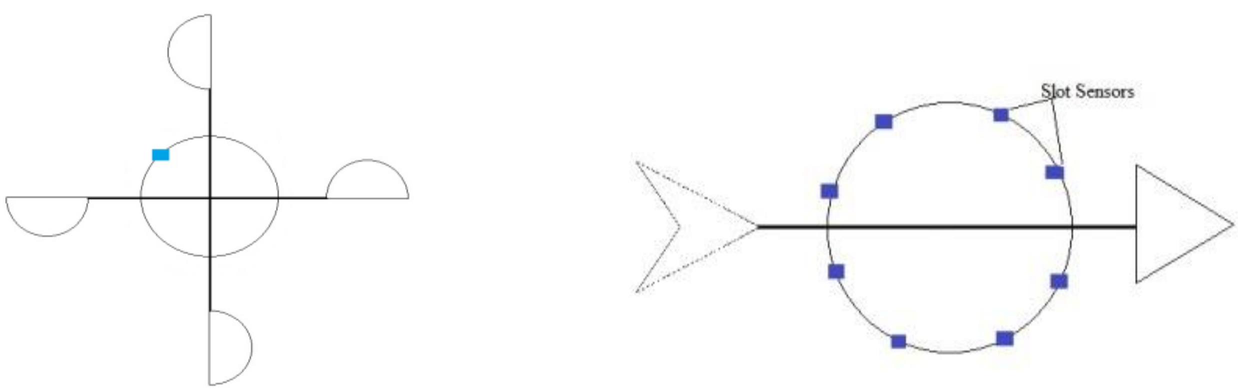


Figure 3.2.3 circuit diagram for slot sensor



(a) (b)

Figure 3.2.4 (a) measurement of wind speed. (b) measurement of wind direction.

C. Wind speed and direction measurement

The wind speed and direction are measured using a wind anemometer. It is a cup type anemometer with an 8 switch wind vane. Opto-interrupters, also called as slot sensors, are used to measure the rpm and to find the direction of the wind. It is a slotted opto-isolator module, with an IR transmitter & a photodiode mounted on it. It detects an interruption between the diode and transmitter.

The sensor used here is MOC7811. 8 sensors are used to detect the 8 different directions for wind and a single sensor is used to measure the rpm. This design is shown in figure 3.2.4. The circuit for opto-interrupters is shown in figure 3.2.3. Each of the 8 sensor data pins are connected to eight GPIO pins of the Pi. The interfacing code is written using Python.

Wind speed is calculated using formula:

$$Speed = RPM \times d \times \pi \div 60\ kmph$$

Where *d* is the diameter of the anemometer and RPM is the count of interrupts per minute.

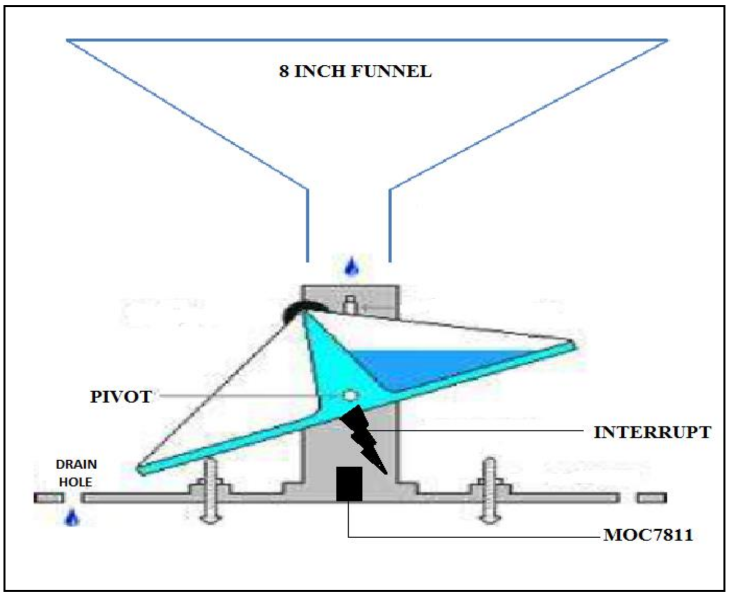


Figure 3.2.5 Tipping bucket rain gauge

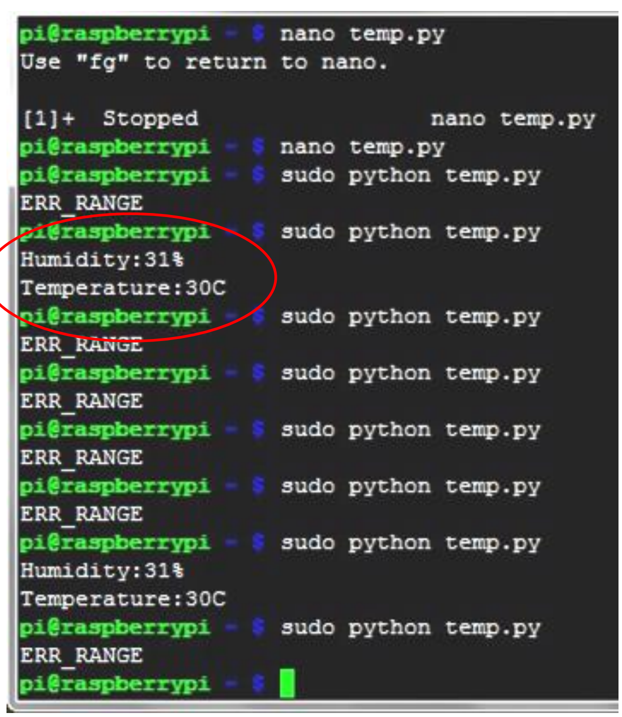


Figure 4.1. Output of temperature and humidity sensor

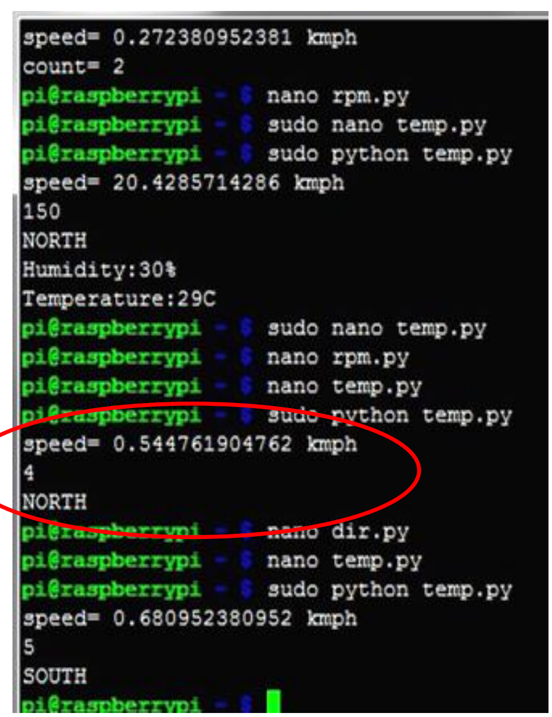


Figure 4.2. Output of wind anemometer

D. Rainfall measurement

A tipping bucket rain gauge is used to measure rainfall. Slot sensors are used here to detect the number of times the bucket tips. Calibration is done based on the amount of water required for the bucket to tip. Figure 3.2.5 shows the setup of the tipping bucket rain gauge.

4. IMPLEMENTATION AND RESULTS

A screenshot of the outputs of the various modules of the weather monitoring station is shown in the figures 4. After the execution of the program, the temperature is shown in ºC. Humidity is shown in relative percentage (%). The output is shown every 2sec. This is shown in figure 4.1. The number of times the slot sensor is interrupted is counted and displayed. This is done every 60 sec, thus giving the RPM of the wind anemometer. 8 slot sensors are used to find the wind direction. The RPM is then converted to kmph using the formula. The speed in kmph, RPM count and the direction is shown in figure 4.2.

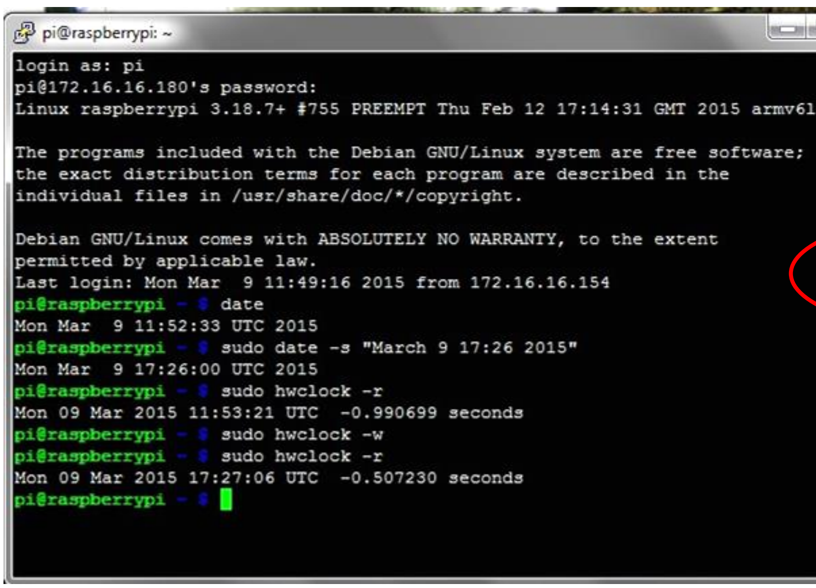


Figure 4.3. Output of RTC

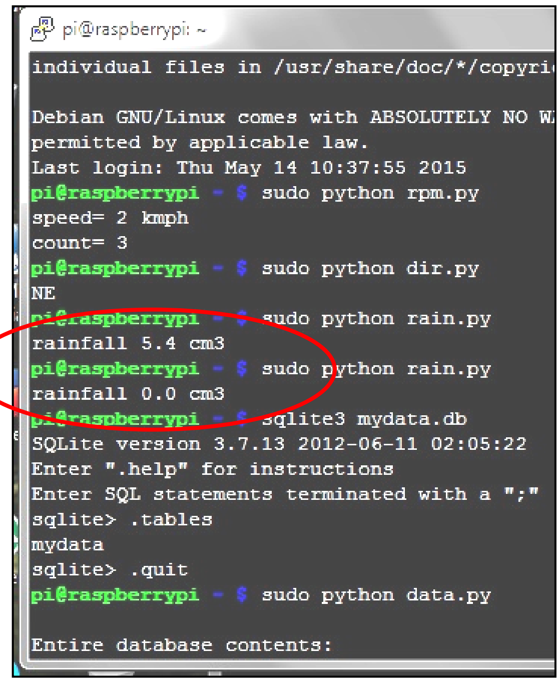


Figure 4.4 Output of tipping bucket rain gauge

Initially the RTC is set to a default time by the manufacturer. When connected to the Pi, this default time and date has to be rewritten to the current time and date. After this is done, the hardware clock will be set to the current date and time and it can be used to provide services that require time stamp. This is shown in figure 4.3. A slot sensor is used to measure the amount of rainfall for 1 min. The number of times the bucket tips is counted and accordingly the rainfall is measured. A python code is written for the interfacing. Fig 4.4 shows the output of the rain gauge. A python script is written to integrate all sensor readings. MySQL database is created with these readings and then they are automaticolly updated to the Internet[8.].

5. CONCLUSION AND APPLICATIONS

Keeping in mind the advantages of the Raspberry Pi, the a low cost weather station is designed using Pi. The temperature , humidity, wind speed, wind direction and the amount of rainfall is measured and monitored using this device. The RPi is configured as a web server so that the weather station serves as a remote device whose data can be accessed from any other system connected to the same network as that of the Pi. The data obtained from this system can then be used for various purposes such as automated irrigation systems, automated temperature control for homes, offices, warehouses and factories, green house climate control, for tracking hazardous materials released into the air is the Biological Identification and Detection System (BIDS), pollution monitoring and many more such applications.

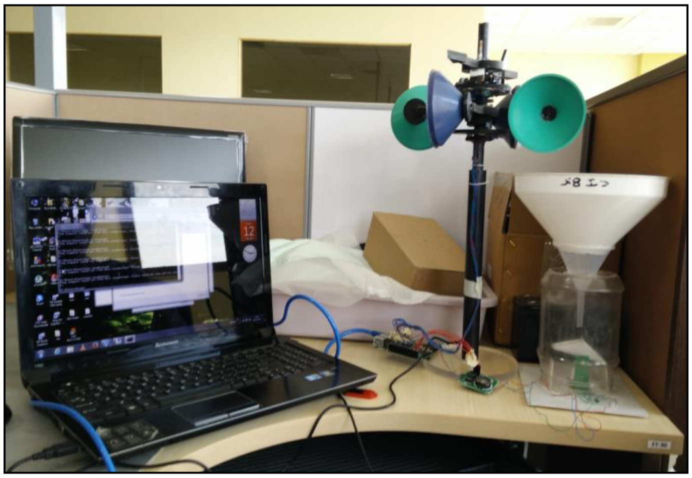


Figure 4.5 Weather Monitoring System Module

ACKNOWLEDGEMENT

We would like to thank Mr. Shankaranand P V, Manager, VinyasIT, Mysuru, for providing constant support and guidance for carrying out this work.

REFERENCES

1. The importance of weather monitoring, Canadian Environmental Protection (May/June 2006) 6-7.
2. A Low-Cost Microcontroller-based Weather Monitoring System, Kamarul Arifin Noordin, Chow Chee Onn and Mohamad Faizal Ismail, CMU. Journal (2006) Vol. 5(1).
3. Microcontroller based Real Time weather monitoring device with GSM, K C Goudal, Preetham V R and M N Shanmukha Swami, IJSETR Vol. 3, Issue 7, July 2014.
4. Design and Implementation of Weather Monitoring and Controlling system, P Susmitha and G Sowmyabala, International journal of computer applications, vol. 97-No. 3, July 2014(0975-8887).
5. Setting Up Your Raspberry Pi For Python GPIO Projects, http://www.thirdeyevis.com/pi-page-1.php
6. https://chrisbaume.wordpress.com/category/technology/raspberry-pi/
7. https://www.raspberrypi.org/learning/web-server-wordpress/worksheet.md
8. https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md
9. http://www.instructables.com/id/Arduino-Weather-Station
10. http://iot-projects.com/index.php?id=diy_hms_raspberry_pi
11. http://www.suntimebox.com/raspberry-pi-tutorial-course/week- 3/day-5/
12. https://iada.nl/en/blog/article/temperature-monitoring-raspberry-pi
13. http://www.instructables.com/id/Raspberry-Pi-Temperature-Humidity-Network-Monitor/step1/Physical-assembly/
14. https://www.raspberrypi.org/forums/
15. https://www.google.co.in/search?sclient=psy-
16. http://www.columbiaweather.com/products/weather-stations