8th EnvImeko - IMEKO TC19 Symposium on Environmental Instrumentation and Measurements
October 29-30, 2019, Sfax, Tunisia

IoT Real-Time Remote Environment Monitoring System

Amir Hammami
Faculty of Business Administration, Computer Science Department
Afif Branch, Shaqra University, Kingdom of Saudi Arabia
Ecole Nationale d'Ingénieurs de Gabès, Tunisia
Amir.hammami@su.edu.sa

***Abstract*— In this research paper, we propose a computer system whose main objective is to connect different sensors to a web server containing a database. The monitoring system can help to view in real time environment data like temperature and humidity. By using sensor connected to the Raspberry Pi Minicomputer which stores them on remote server and makes them available to Internet users. Collected environment data are sent to ThingSpeak web server using simple Internet connection. A supervisor through a dashboard monitors various data. The system can help to resolve emergencies. In this way, it simultaneously improves the quality of the environment through constant attention. This project enters in the concept of Internet of Things which consists of integrating the real-world into the Internet.**

***Index Terms*—Environment Monitoring, Sensors, GPS, Internet of Things, Raspberry Pi.**

I. Introduction

Nowadays Internet of Things (IoT) gained a great attention from researchers, since it becomes an important technology that promises a smart human being life, by allowing communications between objects, machines and everything together with peoples. IoT represents a system which consists of things in the real world, and sensors attached to or combined to these things, connected to the Internet via wired and wireless network structure. IoT promises many applications in human life, making life easier, safe and smart. There are many applications such as smart cities, homes, transportation, energy, smart environment and E-Health Real Time Heart Beat Monitoring and GPS Tracking based on Internet of Things [1].

The main purpose of this project focuses on connecting real-world objects to the internet through different technologies. We can connect temperature, rain, humidity and soil sensors to servers scattered on the internet. It is therefore possible to store different types of information (temperature, humidity, ...) then analyze them later and make appropriate decisions and interventions. Real data, collected from the various sensors, inform the users about the evolution of their environment in real-time. Beyond the simple provision of information, the aggregation of these data from heterogeneous sources enables to quantify the surrounding environment in order to identify trends and enrich uses. After collecting data and transmitting them to remote database, the supervisor monitors these data and observes evolution. If data exceeds a certain threshold, alerts can be evoked.

The main advantage of our solution is the real-time sending of all this information to the Internet. It is possible to put several autonomous devices in different places that send the data continuously. Real-time monitoring becomes possible through this platform.

This paper is structured as following: section 2 presents several related works dealing with the problematic of the collection data and publishing them on the Internet. Section 3 describes the overall system architecture and present the offered services and features. It also introduces hardware components that makes up our system. A description of the software part will be presented in section 4, before concluding the paper and citing the perspectives in section 5.

II. Related Works

Data monitoring system has been widely used in many fields especially in case of weather station. Authors in [2] present an automated weather station for real time and local measurements. Their system continuously measures several weather factors. The results are displayed in an iOS and Android application through blynk platform. Two parts compose the system which are located indoor and outdoor. Another weather station is presented in [3]. The hardware module is based on Arduino board and Zigbee wireless technology. It measures the meteorological data including air temperature and barometric pressure.

III. System Features

The Environment data monitoring System proposed in this paper has the following features:

- Acquisition of a geographic coordinates (latitude, longitude, date, time) in real time using GPS Module,
- Measure of temperature, humidity, pressure and gas,
- Transmission of collected data (GPS location and other values) to a web server using Wi-Fi module after a specified time interval (Push Mode),
- A supervisor monitors the data and can visualize positions on Google map in real-time using Smartphone or Laptop (Pull Mode),

- With the data collected and analyzed, we can make decision about this environment and perform intervention.

The system follows a layered architecture that can be mapped to the Internet of Things Layers. In particular, the Application Layer concerns the analysis layer based on web technologies. This architecture implements the previous listed features.

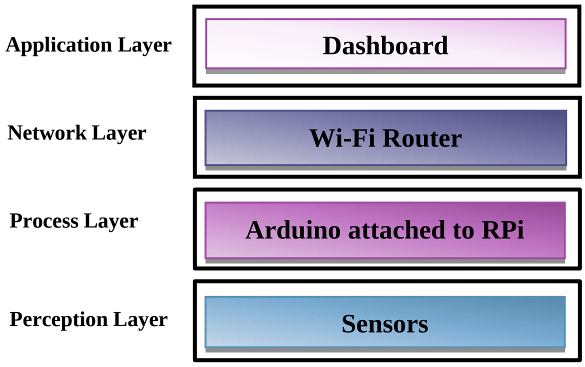

Fig. 1. Layered Architecture

From this architecture we identify four major steps to enable real-time data monitoring:

1. Capture environment data like temperature and humidity. This is done using sensors connected to Raspberry Pi board,
2. Have a Python program running in the Raspberry Pi that reads values and stores them into ThingSpeak server,
3. In the end, we can point the internet browser to a ThingSpeak website and get a plot of the collected data.

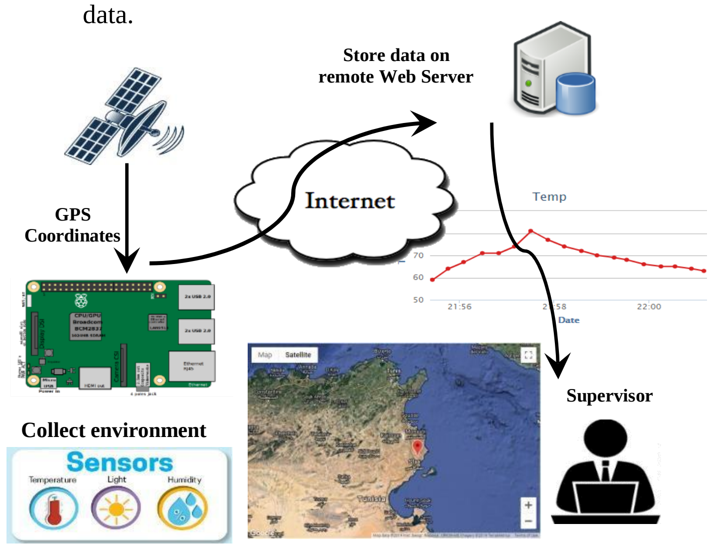

Fig. 2. Global Architecture

The associated devices for the different layers and their functionalities are given below:

A. Perception Layer

Sensors convert various forms of physical energy into electrical energy, allowing microcontrollers to read changes in the physical world. Many sensors work by converting the energy they read into a changing electrical resistance by using a variably resistive material at their heart. For example, Photoresistors or light dependent resistors change their resistance when exposed to a change in light energy. In order to read changes in resistance, you typically place these sensors in a voltage divider circuit, which converts the resistance change into a changing voltage.

1) *GPS Module:* GPS [4] (Global Positioning System) is a satellite navigation system to locate the position of an object in the earth. It provides information such as the location coordinates (latitude, longitude, altitude, date, time, speed, and so on). NMEA (National Marine Electronics Association) protocol is a standard of data in the navigation system. The GPS receiver collects these data from satellite. Adafruit offers several GPS offerings: - GPS only module[1], - Data logger shield[2] which includes micro SD card slot and prototyping area, - Wearable GPS module[3], the GPS receive module alone[4] and Ultimate GPS HAT for Raspberry Pi Shield[5]. In our study, we used the last Shield plugged on top of Raspberry Pi board as shown in the figure below:

Fig. 3. Adafruit Ultimate GPS HAT

Fig. 4. Adafruit Ultimate GPS HAT plugged in Raspberry Pi

[1] https://www.adafruit.com/product/746
[2] https://www.adafruit.com/product/1272
[3] https://www.adafruit.com/product/1059
[4] https://www.adafruit.com/product/790
[5] https://www.adafruit.com/product/2324

Persons who are interested with this GPS shield car learn how to use it following this link [5] from Adafruit Learn Center.

2) *Temperature, Humidity, Pressure and Gas Sensor:* The BME680 from Bosch gives us all the environmental sensing we want in one small package. This little sensor contains temperature, humidity, barometric pressure, and VOC gas sensing capabilities.

Fig. 5. BME680 Sensor

Like the BME280 & BMP280, this precision sensor can measure humidity with ±3% accuracy, barometric pressure with ±1 hPa absolute accuracy, and temperature with ±1.0°C accuracy. Because pressure changes with altitude, and the pressure measurements are so good, we can also use it as an altimeter with ±1 meter or better accuracy.

Fig 6. shows how to connect this sensor to Raspberry Pi MiniComputer using small breadboard.

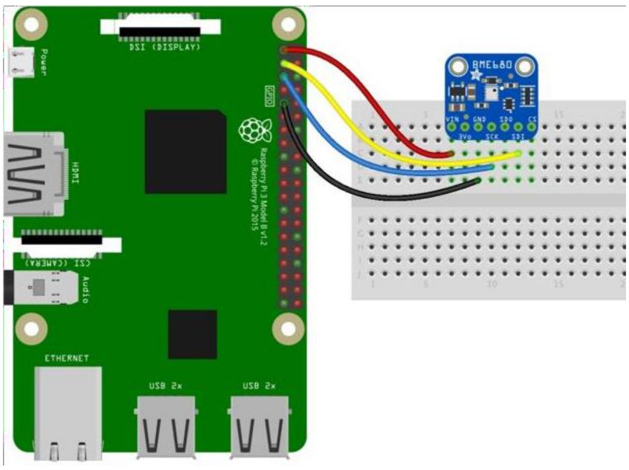

Fig. 6. Circuit Diagram for BME680 Sensor with Raspberry Pi

More practical details with this sensor, can be found following this link [6] from Adafruit Learn Center.

B. Process Layer

In order to collect data and send values to Internet, we need a specific hardware with a processing logic to perform these tasks. Raspberry Pi is the ideal board for IoT projects. The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. It does not include peripherals (such as keyboards and mouse) or cases. However, some accessories have been included in several official and unofficial bundles. Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU). Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+ or 1.5 GHz for the Pi 4; on-board memory ranges from 256 MB to 1 GB with up to 4 GB available on the Pi 4 random-access memory (RAM). The boards have one to five USB ports. For video output, HDMI and composite video are supported. Lower-level output is provided by a number of GPIO pins, which support common protocols like I²C. The B- models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth. Prices range from US$5 to $55 [7].

At the time writing this paper, the current version is Raspberry Pi 4 Model B released in June 2019 with a 1.5 GHz 64-bit quad core ARM Cortex-A72 processor (Fig 7.).

Fig. 7. Raspberry Pi 4 Model B

Like any other computer, RPi uses an Operating System. The Linux option called Raspbian based on Debian version is a great match for RPi because it's free, open-source and there are many libraries, frameworks and programs available on the Internet that we can download, install and use with RPi.

C. Network Layer: Wi-Fi Router with SIM Card

Raspberry Pi will push environment data values to a remote server. The suitable solution consists of using a Wi-Fi router which has two interfaces: (i) an external interface and (ii) an internal interface. The external interface is based on GSM/GPRS standard. It allows the router to be connected to Internet using the GSM network operator of the country. GSM (Global System for Mobile communication) is digital cellular system used for mobile devices. It is an international standard for mobile which is widely used for long distance communication. GPRS (General Packet Radio Services) is a packet based wireless communication service that works with data rate of 56-114 kbps. GSM already has GPRS facilities inbuilt. A SIM card (pre-paid) is required to implement the data monitoring system.

The internal interface is based on Wi-Fi local network. It allows Raspberry Pi to be connected to the Wi-Fi hotspot and thus to Internet. The Wi-Fi hotspot acts as a bridge between Raspberry Pi and the Internet.

In this project we used HUAWEI E5330 as a mobile Wi-Fi device. It is produced by Huawei technologies and it provides Internet access through any W-Fi device. The device supports GSM/GPRS standard. It is possible to plug a pre-paid SIM card and connect to the GSM network operator of the country. By using this device, we create our own secure mobile Wi-Fi hotspot. This device is fast, reliable and easy to use.

Fig. 8. Huawei E5330 mobile Wi-Fi device

D. Application Layer: Webpage based dashboard

We have to make difference between two locations: Where the system is located and from where the supervisor monitors data. The first location denotes where the system is placed. The system is an autonomous box with power supply that can be placed anywhere in condition that the GSM network is enabled. The reason of this constraint is to enable Internet connections.

After transmitting values to a remote server, a supervisor can visualize data on charts. A detailed description is given in the next section.

IV. Software Implementation

In the software implementation we distinguish four parts:

A. Collect environment data

We need to install Adafruit_Blinka library that provides support in python. This may also require to enable I2C on Raspberry Pi board and verify that we are running Python 3. From the command line, we run the following command:

```
sudo pip3 install adafruit-circuitpython-bme680
```

we run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
import board
import busio
import adafruit_bme680
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_bme680.Adafruit_BME680_I2C(i2c)
```

Now we're ready to read values from the sensor using any of these properties:

- temperature - The sensor temperature in degrees Celsius.
- gas - The resistance (in Ohms) of the gas sensor. This is proportional to the amount of VOC particles in the air.
- humidity - The percent humidity as a value from 0 to 100%.
- pressure - The pressure in hPa.
- altitude - The altitude in meters.

```
print('Temperature: {}
       degrees C'.format(sensor.temperature))
print('Gas: {} ohms'.format(sensor.gas))
print('Humidity: {}%'.format(sensor.humidity))
print('Pressure: {}hPa'.format(sensor.pressure))
```

B. GPS Latitude and Longitude

We followed the Adafruit guide but noticed that we could get debug output from the serial port without first running stty to configure it, whereas if we ran this to set the options specified we did not. So in short, to test we are getting data from the GPS module simply use:

```
cat /dev/ttyS0
```

Data output from the GPS receiver are formatted as standard “NMEA sentences”. Pynmea is a library used to parse collected data and extract the current latitude and longitude.

C. Python Daemon

Where to put the data became the next question. A customized solution consists of building a private web server. This web server contains dynamic webpages that interact with a database. A programming script like PHP can be used to store data on tables and then perform selection queries. Using a plotting JavaScript library like Flot allows visualizing data on charts.

Another solution which is a simplified approach consists of sending all collected data to a predefined remote server. Such servers exist on Internet and are free for use for some period of time. We can cite Google Sheets, Dweet.io and ThingSpeak.

In our project we decided to leverage services offered by ThingSpeak. It has an API and let us post and review the data from anywhere. ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. The real-time environment data are visualized in the form of charts in ThingSpeak. The “ThingSpeak channel” is the core element. It wills stores the data that we send to this remote server.

On Raspberry Pi we run a Python script. This program is going to read data from Arduino and then publish the values of that data to our channel on ThingSpeak.

D. ThinkSpeak Server

On ThingSpeak website we should create a free account or login to an existing account. We create a new channel with many fields according to environment data. The order of the fields is important later when we post data. We will also need to write API key for the channel as it will be required to post data. Lots of experiments have been done. If we check our channel on ThingSpeak we obtain the following charts:

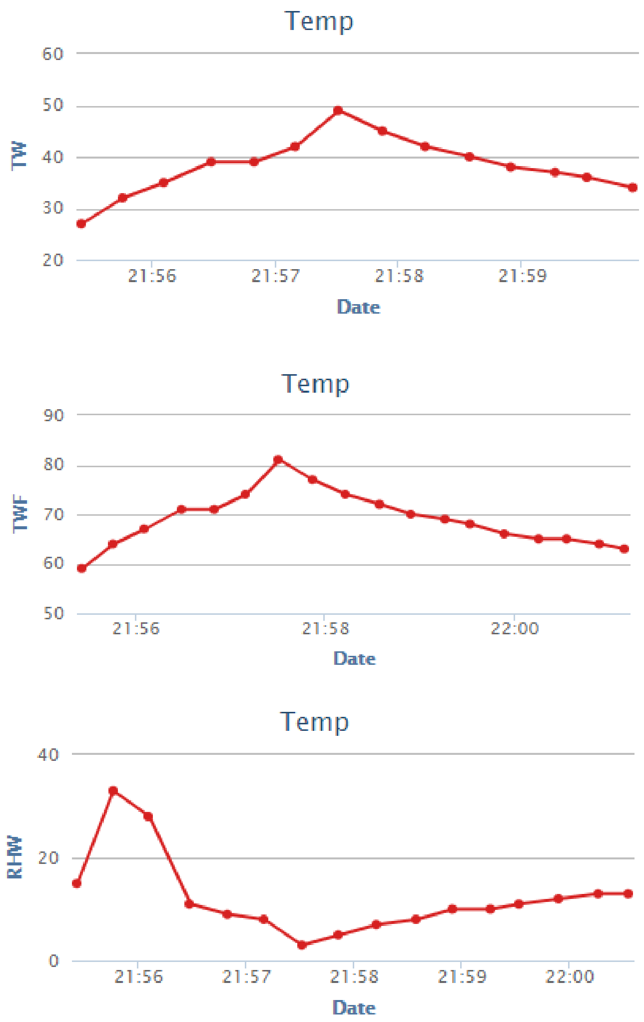

Fig. 9. Experiments

#Convert from Celius to Farenheit: TWF = 9/5*TW+32

V. Conclusion

The experimentation was a success and from this experimentation we can understand how to measure the temperature and humidity data of a space in a particular place. By using this IoT system in a cloud storage, we can evaluate and monitor the various data by our mobile phone or via PC's.

References

[1] A. Hammami, "Heart Beat Monitoring and GPS Tracking based on Internet of Things," Proceedings of the 3rd International Conference on Smart City Applications (SCA'18), Tetouan, Morocco — October 10 - 11, 2018.

[2] M. Farhat et al., "A Low Cost Automated Weather Station for Real Time Local Measurements", Journal of Engineering, Technology and Applied Science Research, Vol. 7, No. 3, 2017, pp. 1615-1618.

[3] H. S. Saini et al., "Arduino Based Automatic Wireless Weather Station with Remote Graphical Application and Alerts", 3rd IEEE International Conference on Signal Processing and Integrated Networks (SPIN-2016), February 2016.

[4] A. Hammami, "Person Tracking System based on Arduino Microcontroller and Web Technologies," 2018 International Conference on Smart Communications and Networking (SmartNets), 16-17 Nov. 2018

[5] Adafruit Learn Center, Ultimate GPS HAT for Raspberry Pi, https://learn.adafruit.com/adafruit-ultimate-gps-hat-for-raspberry-pi

[6] Adafruit Learn Center, BME680, https://learn.adafruit.com/adafruit-bme680-humidity-temperature-barometic-pressure-voc-gas

[7] Wikipedia, RPi, https://en.wikipedia.org/wiki/Raspberry