Internet of Things (IoT) Based Data Logging and Live Graph Monitoring Using Adafruit io Server

Khin Chaw Su Hlaing  
Yangon Technology University  
khinchawsuhaing1976@gmail.com

Me Me Aye  
Yangon Technology University  
memeaye75@gmail.com

Set Swe  
Yangon Technology University  
setsetswe27@gmail.com

Abstract

Physical data are determined, monitored and logged in various methods. Internet of things (IoT) based data logging system is more reliable to use than others because of the lack of labors and less expenditure. In this research, the physical data such as temperature and humidity are measured using DHT 11 sensor ESP8266 NodeMCU. The process is implemented by using NodeMCU Wi-Fi module which is a microcontroller integrated with Wi-Fi and Bluetooth capability. As DHT 11 produces 40-bit digital output including both temperature and humidity data, it is connected to one of the digital pin of microcontroller. NodeMCU ESP8266 Wi-Fi module is contributed in data collection and data sending to “Adafruit io” via internet router. Adafruit io provides very good tool for IoT based researches. By using Adafruit site, data can be monitored and control the system over the Internet. Adafruit io ‘Collects’ the data from the sensors, ‘Analyze and Visualize’ the data and ‘Acts’ by triggering a reaction. This research can be used in the environmental science as well as in remote weather station.

1. Introduction

This IoT based data logging system is basic operation of automatic weather station (AWS) which is an automated generation of traditional weather station, either to save human resources or to measure from the remote areas. Generally, the AWS consists of a weather proof container which contains the meteorological sensors, data processor, Wi-Fi module and rechargeable battery. The system may report in real time via internet and save data for later recovery. Nowadays, the solar panels are used for power supply and smartphone technology have made possible to have wireless station.

In this research, the temperature and humidity are measured by DHT11 sensor. These values are processed by ESP8266 NodeMCU chip (further abbreviated as NodeMCU) which is the system on chip (SOC) with Wi-Fi and Bluetooth capability. This device is particularly interesting because it is recognized and supported by the Arduino programming environment (Arduino IDE). Because of this sensors originally designed to be used with an Arduino can be used as well with a NodeMCU.

The ESP 8266 receives the temperature and humidity from the DHT 1 sensor and then, it connects to the Adafruit io server via Wi-Fi. The sensor data are sent to the “feeds” of Adafruit io through message queuing telemetry transport (MQTT) which is an open, light weight, publish-subscribe network protocol.[1] It can communicate between remote locations with lossless and bi-directional connections. Web server is configured with the Adafruit io logging parameters such as: - start/stop sending data, refresh time, application programming interface (API) key.

Adafruit io is a cloud service which can connect the NodeMCU through internet and read data from sensor remotely from anywhere with internet access.[2] Adafruit io dash board page is created by signing up. The temperature and the humidity are inserted in “feeds” on dash board. “AIO key” is obtained by clicking “view AIO key” on “setting”. This key is pasted in the program. Sensor data are transmitted by the NodeMCU via Wi-Fi to a router connected to the internet. A server at Adafruit io (https://Adafruit io.com) provides a timestamp and stores the data.

2. Materials and Methods

In this research, the temperature and humidity are logged over internet and Google sheets so that data can be monitored from anywhere in the world over internet. Materials used in the temperature and humidity monitoring system are DHT11 sensor and NodeMCU Wi-Fi capability module as the essential components. NodeMCU is functioned as sensor data processing and Wi-Fi station (STA) mode. The Internet of things (IoT) is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensor), these devices can communicate and over the Internet, and they can be remotely monitored and controlled. [4]

2.1. ESP8266 NodeMCU

NodeMCU is an open-source Lua based firmware and development board specially utilized for IoT based implementations. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.
ESP8266 NodeMCU is the microcontroller with integrated Wi-Fi and Bluetooth. ESP8266 chip composes 32-bit reduced instruction set controller (RISC) processor which operates with adjustable frequency 80Hz to 160Hz. The 128KB internal RAM and 4MB flash memory have already added in NodeMCU. ESP8266 can connect to Wi-Fi network or can interact with internet. Moreover, it can set up own network as server. CP 2102 USB - to - UART controller chip which included in ESP8266 board converts USB signal to serial data for communication with computer.

The general purpose input output (GPIO) pins in ESP8266 NodeMCU are as shown in figure 1. These pins are assigned to some peripheral capabilities including: 1) one analog-to-digital converter (ADC), 2) one universal synchronous receiver and transmitter (UART), 3) four pulse width modulation (PWM), 4) two serial peripheral interface (SPI), 5) one inter-integrated circuit (I2C) and 6) one inter-integrated sound(I2S). These pins are multiplexing features, that is; a single GPIO pin can multiplex as PWM / SPI / I2C.

ESP8266 board is compatible to Arduino IDE. NodeMCU can do Wi-Fi connection to server via internet.

2.2. DHT 11

DHT 11 is a Sensor which can produce the digital output. The DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms. Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first.

Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of “8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data”.

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again. Overall communication process is shown in Figure 3.

The data pin of DHT 11 is connected to D2 of NodeMCU12E as shown in Figure 4.
3. System Operation

System operation is divided into two sections; hardware preparation and software development.

3.1 Hardware Preparation

NodeMCU microcontroller accesses data from DHT11 and send data to Adafruit io server by means of Wi-Fi because NodeMCU possesses Wi-Fi module based on ESP8266. The fresh data with date/time can be received in Adafruit io server each two seconds. And then, these data can be viewed and stored in laptop, tablet and smart Android phone using internet access. The block diagram of IoT based data logging system is shown in Figure 5.

![Figure 5](image_url)

**Figure 5. Basic configuration of the proposed system**

Although ESP8266 NodeMCU is compatible to Arduino IDE the NodeMCU board is not included in the default Arduino IDE. After IDE window is open, the path of ESP8266 board packet http://arduino.esp8266.com/stable/package_esp8266com_index.json is pasted into the “Additional Board Manager URLs” Field of preference in IDE window as shown in the Figure 6. After URL path is defined, ESP8266 board can be installed using the board manager, as in Figure 7.[4]

![Figure 6](image_url)

**Figure 6 Including NodeMCU board in Arduino IDE**

3.2 Software Development

Firstly, the libraries of ESP8266Wi-Fi.h, Adafruit_MQTT.h, Adafruit_MQTT_Client.h and DHT used in this research are included in Arduino IDE. Service set identifier (SSID) and its password, Adafruit io user name and its AIO key are defined as:

```c
#define WLAN_SSID       "********"
#define WLAN_PASS       "**********"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883 // use 8883 for SSL
#define AIO_USERNAME    "***********"
#define AIO_KEY   "***********"
```

The data baud rate of serial monitor is defined as;

```c
Serial.begin(115200);
```

Microcontroller get the temperature and humidity data as follows;

```c
float h = dht.readHumidity();
float t = dht.readTemperature();
```

Setup the MQTT client class by passing in the Wi-Fi client and MQTT server and login details. Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

Setup a feed for publishing.

```c
Adafruit_MQTT_Publish temperature = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME="/feeds/temp");

Adafruit_MQTT_Publish humidity = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME="/feeds/hum");
```

MQTT (messageQueueing Telemetry Transport) is machine-to-machine(M2M)/internet of thing(IoT) connectivity protocol. It is useful for connection with remote area.[3,4,5]

The flow chart diagram of system operation is illustrated in Figure 8.
4. Results

Firstly, “Adafruit io” account is created with email address. Then, the circuit is connected to PC via serial communication. The program is uploaded to NodeMCU. If the system has connected to “Adafruit io” server via Wi-Fi, the uploading is done. The specified channel in Adafruit io window is chosen and data can be set in option. Then, if the specified channel is open, the data is plotted in graph on Adafruit io window. The serial monitor also shows the data simultaneously. The data are shown in Figure 9 to Figure 11. These data are also directly displayed on the smart phone with internet access.

5. Discussion

In this research, the web server in station (STA) mode is communicated to the more than five numbers of clients such PSs, phones and tablets.

The set of data display on both laptop and phone at the same time seen that the decimal values of temperature are different in little because the performances of devices such as PC, phone are different. GPIO 6, 7, 8, 11 and A0 do not work in digitalWrite command while GPIO 1, 3, 6, 7, 8, 11 and A0 do not work in digitalWrite command.

If the program cannot be uploaded to ESP8266 NodeMCU. The following steps will implement;
1) Disconnect ESP8266 from power
2) Make the upload mode by connecting GPIO0 to ground.
3) Then, supply power again to ESP8266 and upload.

6. Conclusion

DHT 22 sensor has better humidity measuring range, from 0 to 100% with 2 – 5% accuracy, while DHT 11 range from 20 to 80% with 5% accuracy. And also, DHT22 has better temperature measuring range, from -40 to +125°C with ±0.5°C accuracy, while DHT 11 range from 0 to 50°C with ±2°C accuracy. But, the sampling rate for the DHT11 is 1Hz or one reading every second, while the DHT22 sampling rate is 0.5Hz or one reading every two seconds.

The time for data transferring to and from the Adafruit io server is approximately 2 seconds so that the relatively faster than the “Thingspeaks” cloud. It has reliable data storage-the data can be stored for 30 days.

MQTT is one of the most commonly used protocols in IoT projects. In addition, it is designed as a lightweight messaging protocol that uses publish/subscribe operations to exchange data between clients and the server.

This is basic implementation for weather station which can be operated at a remote area without human resource. Data can be collected at the specified refresh time. If data are logged every ten minutes, six data points in an hour, 144 data points in a day and 1728 data points in a year. It is reasonable points to observe the data analysis in weather station and it is reduced the memory space.

For future work, the meteorological sensors and the environmental sensors which can detect temperature, humidity, atmospheric pressure, gas, water turbidity, wind speed, wind direction, rain gauge are added to this system. Data is plotted in excel graph and logged in hard disk for the latter review.

Acknowledgement

I thank all of the teachers who directly or indirectly helped us to complete this paper. I would like to thank all of my friends, Department of Physics, Yangon Technological University for their various kinds of help.

References