USE OF A FREE OPEN-SOURCE IOT PLATFORM IN A CASE STUDY

Hongbo Zhou
Department of Computer Science
Slippery Rock University of Pennsylvania
hongbo.zhou@sru.edu

ABSTRACT
An IoT course for Computer Science major introduces students to basic electricity and electronics, the architecture and programming of single-board microcontroller systems (e.g., Arduino and ESP32) and single-board computer systems (e.g., Raspberry Pi). However, there is no sufficient time to cover every detail in an IoT system. A free open-source IoT platform is utilized in a case study to illustrate the interaction among the sensor, microcontroller, IoT communication protocol, database server, and web server, and to prepare the students for a complete IoT project.

KEY WORDS
IoT platform, free open-source, HTTP, DHT11, ESP32

1. Introduction

The Internet of Things (IoT) is an emerging technology that manages devices with sensors installed in gadgets, appliances, machines, vehicles, etc. [1] The sensor data can be utilized to monitor the state of the environment, user, or device. The actionable knowledge extracted from sensor data results in improvement in efficiency, safety, and cost reduction. Therefore, it has a variety of applications in Industry 4.0, smart home, smart transportation, robotics, and healthcare.

Since the audience of the IoT course is from Computer Science field, they lack the experience in electricity and digital electronics. The computer organization and architecture course in the major covers CPU control, pipelining, instruction set, memory hierarchy, and assembly language. There is currently no Electrical and Computer Engineering program on campus to share the course. Therefore, the contents of the course focus on introduction to basic electricity and electronics, building prototypes with a breadboard and off-the-shelf electronic parts, and programming with single-board microcontroller systems and single-board computer systems. The topic on collection, storage, management, and presentation of IoT data merits its own course in one semester.

To help students get a full picture of the IoT system, a free open-source IoT platform is utilized in a case study. The students created their accounts on the platform, built an IoT prototype with an ESP32WROVER board and a DHT11 sensor to transmit temperature data to the platform, and implemented sensor data visualization with a variety of widgets. The case study illustrates the interaction among the sensor, microcontroller, IoT communication protocol, database server, and web server, and prepares the student for a complete IoT project.

2. Hardware

The hardware components used in the case study are mainly an ESP32WROVER board and a DHT11 temperature/humidity sensor.

2.1 ESP32WROVER

ESP32WROVER is one of ESP32 development boards. It is a low-cost single-board microcontroller system manufactured by Espressif Systems, as Fig. 1 [2]. It comes with two CPU cores with 80 to 240 MHz clock frequency, 4MB flash memory, and 8MB PSRAM. It supports Wi-Fi, Bluetooth, and BLE. The 38 pins support PWM, SPI, I2C, and I2S. The 2 Megapixel onboard camera can be used for face detection and other edge AI applications. Because it can be powered and controlled with a micro USB port, it is ideal for development in a PC lab.

Figure 1. ESP32WROVER board

In addition to ESP32 SDK, Expressif Systems released the Arduino core for ESP32 boards at https://github.com/espressif/arduino-esp32. Because the students in the IoT course have been using Arduino IDE for development on an Arduino Uno board in the first half of the semester, they have no difficulty to switch to programming on ESP32WROVER board. Once the Arduino core for ESP32 was installed, students can click Tools menu and choose "ESP32 Wrover Module" as the platform, as Fig 2. They can create a program (called a sketch in Arduino IDE) as one for Arduino Uno, compile it, and upload the binary code to the board. Some third-party libraries for Arduino can be included in a sketch for ESP32 boards as well.

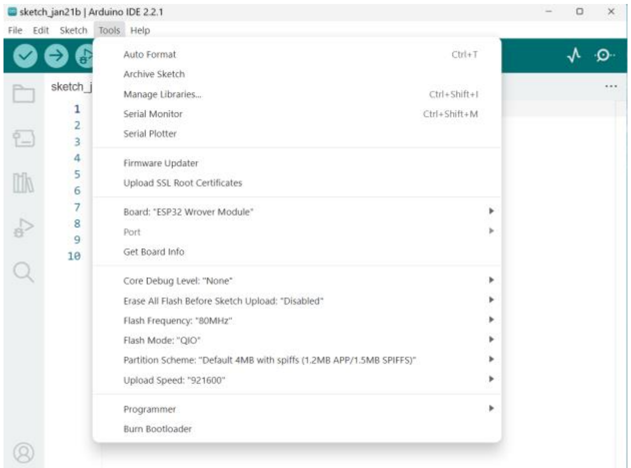

Figure 2. Support of ESP32WROVER in Arduino IDE

2.2 DHT11

DHT11 is a low-cost digital temperature and humidity sensor, as Fig. 3. Its operating range is 0 – 50 °C and 20 – 90% relative humidity, and therefore is well suited for indoor use [3].

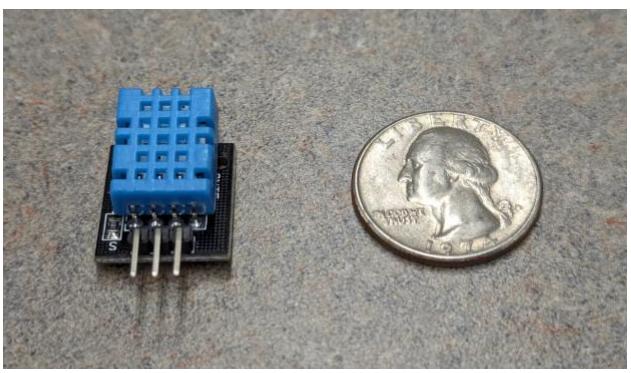

Figure 3. DHT11 temperature/humidity sensor

3. IoT Platform

An IoT platform is a software suite either installed on the site or running in a cloud service that manages sensors and actuators [4]. With the IoT platform, the user can register the devices and automate collection and management of sensor data, and issuance of commands or alerts.

The IoT platform is a sophisticated system, comprising the following technology and software components:

(1) Gateway software that receives sensor data and saves them to a database

(2) A database system that manages sensor data

(3) A web application that retrieves sensor data from the database for visualization

(4) Additional packages, such as data mining and machine learning modules to extract actional knowledge from the sensor data. For instance, in predictive maintenance, sensor data can be utilized to predict when a part is going to fail.

The implementation of IoT platform requires knowledge in network protocols, network programming, database management and programming, full-stack development, web graphics, etc. Even the students took courses on client-side and server-side web programming and database before, an informal survey conducted in the class in Spring 2023 found that none had heard of or used any IoT platform.

The same scenario applies to many STEM courses. With the proliferation of interdisciplinary areas, an instructor is facing the dilemma that there is no sufficient time to cover a broad list of topics and to dive deep into details in the same semester. Therefore, a hands-on case study combining multiple topics is desirable in effective teaching. In the IoT course, a case study based on a free open-source IoT platform is necessary to give the students the first-hand experience to build a complete IoT project.

Among several options, ThingsBoard website has the following advantages [5]:

(1) It offers a free cloud service to register devices to manage IoT data.

(2) It provides both a commercial edition and a free open-source community edition that can be installed in a local server.

(3) It supports a variety of communication methods and database systems.

(4) The web interface is user-friendly and accessible from both desktops and mobile devices.

(5) Its rich and excellent documentation makes it easy for the students to get started.

4. IoT Project

The project consists of the following steps:

(1) Build the circuit with ESP32WROVER and DHT11

(2) Implement the function to collect sensor data in a sketch

(3) Implement the function in the sketch to transmit the sensor data to the IoT platform

(4) The sensor data are saved into the database of the IoT platform

(5) The sensor data are retrieved for visualization on the IoT platform

Step (1) and (2) can be done with the sensor and microcontroller board. Steps (3) to (5) need support from ThingsBoard cloud service.

4.1 The Circuit

DHT11 sensor has a simple interface. There are three pins for signal output, power input, and ground. They are connected to a GPIO pin, power output, and ground pin on ESP32WROVER board, as Fig. 4.

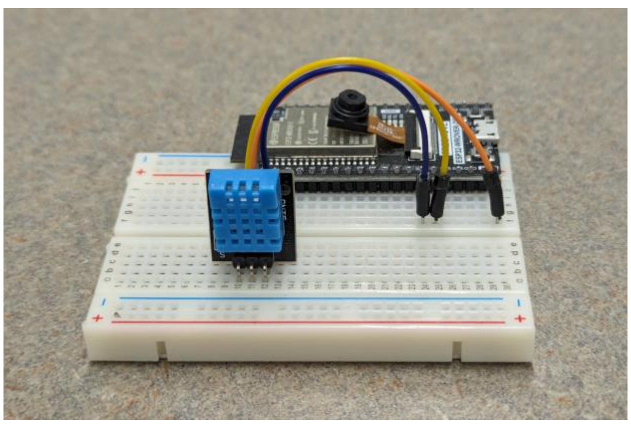

Figure 4. The circuit

4.2 Sensor Data Collection

DHTStable library for Arduino [6] is utilized in the sketch for ESP32WROVER board. Fig. 5 is the sample code of data collection. Fig. 6 shows the output from the board.

```
#include <DHTStable.h>
#define DHTPIN 13
DHTStable dht;
…
void loop() {
  int chk = dht.read11(DHTPIN);
  float t = dht.getTemperature();
  Serial.print("Temp: ");
  Serial.println(t);
  …
}
```

Figure 5. Sample code for data collection

Figure 6. Output of temperature data

4.3 Sensor Data Transmission

By following the tutorial [7], the students created their free accounts at ThingsBoard cloud service, and then registered a device entity for DHT11/ESP32WROVER system, as Fig. 7.

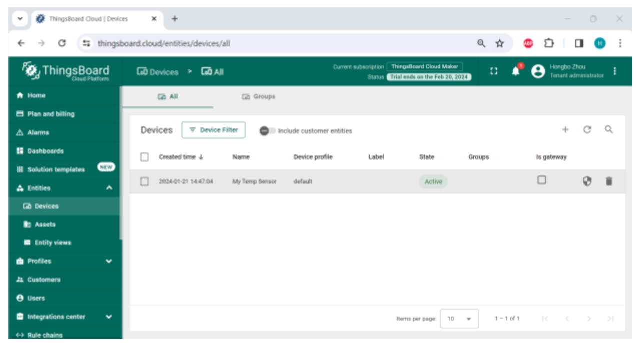

Figure 7. A device entity registered on ThingsBoard

An access token is generated for the device entity, which is used in HTTP POST message to upload sensor data in JSON format and is associated with the device.

After Arduino Wi-Fi library is called to connect the ESP32WROVER board to a wireless network set up in the lab, the sketch utilizes ArduinoHttpClient library [8] to transmit temperature data to ThingsBoard platform. Fig. 8 shows the status of data transmission, and Fig. 9 is the sample code.

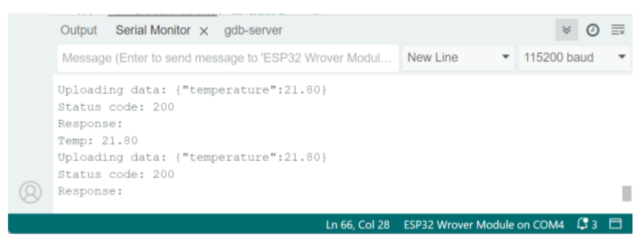

Figure 8. Status of data transmission

```
char serverAddr[] = "thingsboard.cloud";
int port = 80;

WiFiClient wifi;
HttpClient client = HttpClient(wifi,
serverAddr, port);
…
void loop() {
  …
  String contentType =
"application/json";
  String data = "{\"temperature\":" +
String(t) + "}";
  Serial.print("Uploading data: ");
  Serial.println(data);
  client.post("/api/v1/PqMCQ1MQokiZStjqH
bXU/telemetry", contentType, data);

  unsigned int statusCode =
client.responseStatusCode();
  Serial.print("Status code: ");
  Serial.println(statusCode);
  String response =
client.responseBody();
  Serial.print("Response: ");
  Serial.println(response);
  …
```

Figure 9. Sample code for data transmission

4.4 Sensor Data Storage and Visualization

On receipt of the sensor data, ThingsBoard takes care of saving them in a database. To visualize the data, a user needs to create a dashboard with widgets, as Fig. 10. In Fig. 11, a timeseries line chart and an analog gauge widget are included to present the same temperature data in real time.

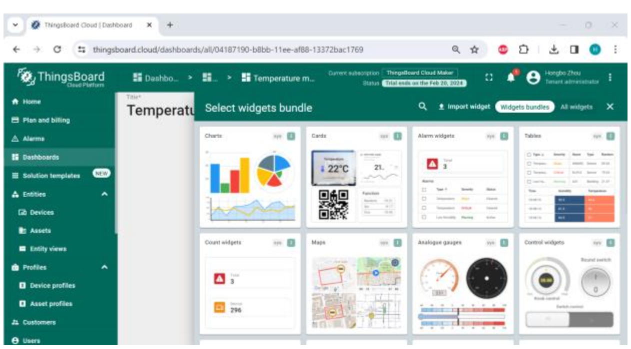

Figure 10. Widgets for sensor data visualization

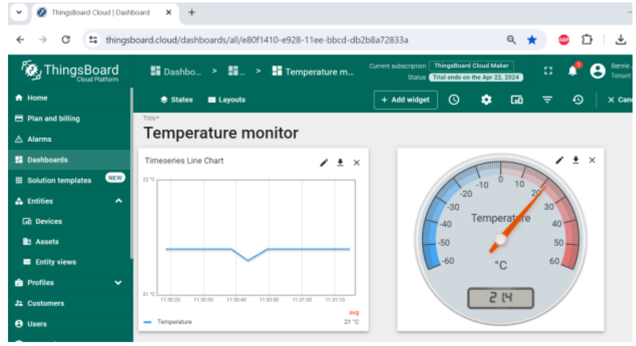

Figure 11. Visualization of real-time temperature data

5. Conclusion

Although ThingsBoard is a free open-source system, it offers a complete set of features and functions in an IoT platform. For instance, alarm rules can be created to raise an alarm if the sensor data are outside of a normal range.

The use of a free open-source IoT platform in the case study allows the students to go through the steps to build a complete IoT project, and to gain first-hand experience with all the essential components in an IoT platform. Therefore, it is a valuable tool in effective teaching in the IoT course.

The project can be expanded for collection, storage, visualization, and analysis of other time series data. For instance, A MQ gas sensor can be utilized to monitor indoor air quality or concentration of harmful gases. An accelerometer attached to an engine, a motor, or CNC (Computer Numerical Control) machine can measure the vibration of parts. Therefore, it will also benefit learning in time series analysis in a Machine Learning course.

6. References

[1] Internet of things, Wikipedia, https://en.wikipedia.org/wiki/Internet_of_things, accessed on January 8, 2024

[2] ESP32-WROVE Datasheet, version 2,7, 2023, https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf, accessed on January 8, 2024

[3] DHT11 Humidity & Temperature Sensor, https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf, accessed on January 8, 2024

[4] IoT Platforms, https://www.gartner.com/en/information-

technology/glossary/iot-platforms, accessed on January 9, 2024

[5] ThingsBoard: Open-source IoT Platform, https://thingsboard.io/

[6] DHTStable library for Arduino, https://github.com/RobTillaart/DHTstable, accessed on January 10, 2024

[7] Getting Started with ThingsBoard, https://thingsboard.io/docs/getting-started-guides/helloworld/, accessed on January 10, 2024