

## **Internet of Things (IoT) Laboratory**

### **Dr. Jai P. Agrawal, Purdue University Northwest**

Jai P. Agrawal is a professor in electrical and computer engineering technology at Purdue University, Calumet. He received his Ph.D. in electrical engineering from University of Illinois, Chicago, in 1991, dissertation in power electronics. He also received M.S. and B.S. degrees in electrical engineering from Indian Institute of Technology, Kanpur, India, in 1970 and 1968, respectively. His expertise includes analog and digital electronics design, power electronics, nanophotonics, and optical/wireless networking systems. He has designed several models of high frequency oscilloscopes and other electronic test and measuring instruments as an entrepreneur. He has delivered invited short courses in Penang, Malaysia and Singapore. He is also the author of a textbook in power electronics, published by Prentice-Hall, Inc. His professional career is equally divided in academia and industry. He has authored several research papers in IEEE journals and conferences. His current research is focused on renewable energy technology, smart energy grid.

### **Prof. Omer Farook, Purdue University Northwest**

Omer Farook is a member of the faculty of Electrical and Computer Engineering Technology at Purdue University, Northwest. Farook received the diploma of licentiate in mechanical engineering and B.S.M.E. in 1970 and 1972, respectively. He further received B.S.E.E. and M.S.E.E. in 1978 and 1983, respectively, from Illinois Institute of Technology. Farook's current interests are in the areas of embedded system design, hardware-software interfacing, digital communication, networking, image processing, and biometrics, C++, Python, PHP and Java languages. He has a keen interest in pedagogy and instruction delivery methods related to distance learning. He has a deep commitment to social justice and in achieving economic and educational equity.

### **Mr. Zachary John Anderson**

### **Mr. Dustin Glynn Walker, Purdue University Northwest**

# INTERNET OF THINGS (IOT) LABORATORY

## Abstract

Internet of Things technology is the preferred choice of modern engineers in the industry. Bringing this technology in the undergraduate education has become a coveted and imperative objective of engineering educators. The paper presents the results of a project to develop laboratory exercises using Internet of Thing (IoT) in the Electrical and Computer Engineering Technology major. This project, uses an ESP8266 NodeMCU v1.0 board for connection to the Internet. It employs Arduino programming for the design of several laboratory exercises. These labs include: Introduction to the IoT technology, Basics of the development board ESP8266 NodeMCU V1.0, Data Grab, Phone controlled LED, OLED Clock, ESP8266 Wi-Fi Soft Access Point, Voting Machine, Plane Spotter, Temperature Monitor, Object tracking, Traffic Monitor and Weather Station.

## 1. INTRODUCTION

Internet is a network of live persons. Internet of Things or IoT is a network of things and persons. IoT brings the things alive and there is interacting among themselves and with persons lively. Internet of Things is the network of devices such as vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data<sup>[1][2][3]</sup>. With the help of embedded technology, these things can communicate and interact over the Internet, and they can be remotely monitored and controlled. In the Internet of things, the precise geographic location and also the dimensions of a thing is critical. Therefore, sensors, transducers, locating devices and networks play very important role in IoT.

There are many areas of applications of the Internet of Things like consumer, health, industrial, transportation, security, entertainment and many other<sup>[1][2][3][4][7]</sup>.

Furthermore, IoT enables the technology of Laboratory on the Web. This technology when fully brought in the academic arena, will affect the education very significantly. It is imperative to bring the IoT technology in the Engineering curriculum as soon as possible. This paper presents the laboratory exercises for an undergraduate engineering technology course, designed on the MATLAB and an embedded platform using ESP8256 a Wi-Fi enabled microcontroller.

Interaction in IoT among things and persons involves reading and writing on the web and the embedded systems to interface with. The concept is shown in Fig. 1, where there are two Arduino microcontrollers for interfacing to Things (devices or systems). The Arduinos communicate wirelessly through Internet.

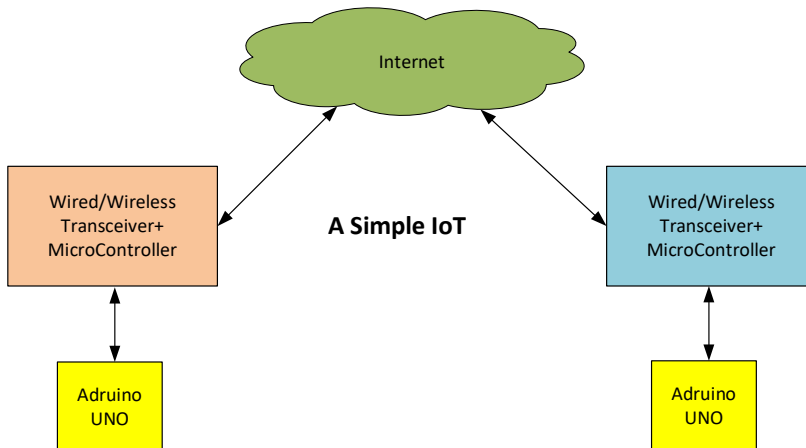


Fig. 1 A simple IoT system

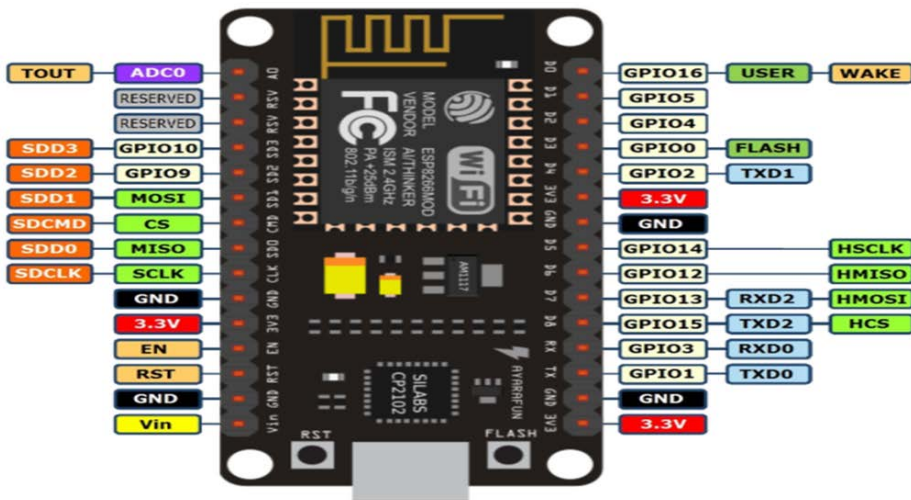


Fig. 2 The Wi-Fi module ESP8266

Internet is the main enabling technology in IoT. The first and last-mile connection to Internet may be either through the connected circuits or via wireless.

The IoT laboratory is designed using MATLAB, Arduino UNO and ESP8266, a low-power, highly-integrated Wi-Fi solution for IoT applications, with highly-integrated on-chip features, offers reliability, compactness and robustness. ESP8266 is integrated with a 32-bit Tensilica L106 32-bit RISC processor, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules. All of them are included in one small package, in Fig. 2. It is engineered for mobile devices, wearable electronics and IoT applications. It consumes low power and a maximum clock speed of 160 MHz.

Computer language C is used for programming. The labs were executed by two senior design students who are the co-authors in this paper.

## 2. LABORATORY EXERCISES

Laboratory exercises are developed with the objective of introducing them in a future course ECET 49900: Internet of Things. In this section we have used Arduino Uno boards as the Things. The labs begin with the concepts of interfacing Arduino Uno board to MATLAB and long distance (through Internet) connection between two Arduino Uno things. The laboratory exercise #3 introduces the Wi-Fi enabled ESP8366 IoT unit. The following laboratory exercises relate to establishing a soft access point for IoT, controlling a LED via IoT, grabbing data from a website with permission, displaying the world clock for a specific location, monitoring and logging the temperature and humidity data at a given location, tracking the distance of an object from an ultrasonic sensor, spotting a plane somewhere and collecting weather map data for a specific location.

**Lab 1 a) - Arduino-MATLAB Interface:** The first lab uses a serial USB interface between the microcontroller Arduino Uno and the MATLAB, see Fig. 3. In this program, MATLAB sends a byte long code on the serial link to the Arduino. The Arduino turns on a LED and returns the message “job done”.

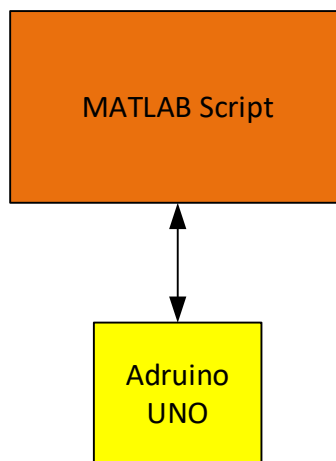


Fig. 3 Arduino-MATLAB Interface

### Lab 1 b) -Long-Distance Serial Link Between Two Arduino Devices

In this part, MATLAB engine on the transmitter side TX is connected via wire-link to another MATLAB engine on the receiver side RX. MATLAB TX prompts the Arduino TX to supply the instruction code to blink an LED on the long-distance

receiver side Arduino RX. The Arduino TX lights up an LED to indicate that it has received the prompt. It sends the required code to the MATLAB TX and turns off the light. The instruction code thus obtained is transmitted to the receiver via a communication channel. The MATLAB RX side sends the code to the Arduino RX attached to it and blinks a LED on pin 8 once and returns the message, “job completed.” to the MATLAB TX.

The two Arduinos are implemented on a single Arduino Uno board, see Fig. 4. The two MATLAB TX and RX are combined into one MATLAB program. The Internet link is a direct wire connection in the MATLAB Script. The project steps must follow the following sequence:

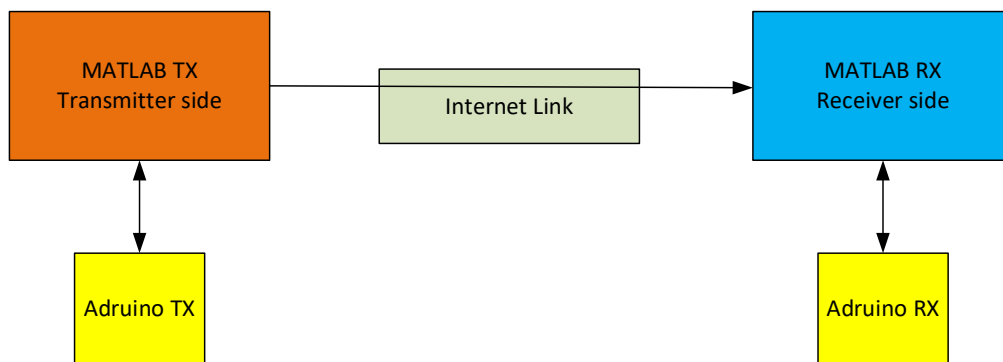


Fig. 4 Arduino to Arduino Communication

**Lab 2- IoT on the MATLAB Platform:** This lab collects data from a weather map website and displays it on the computer. The current temperature and condition show along with the next three days and the date and time.

Write in a file **IoT\_1.txt** a single character ‘3’. We will read from this website as shown below:

**MATLAB Code: Readweb.m**

```

%MATUINO
%courtesy, http://matlabarduinorobotics.blogspot.com/
%initiate the api and open the URL where the instruction for LED on the UNO is
stored.
api = 'http://ecet.purduecal.edu/~jpagrawa/ecet303/';
url = [api 'labs/IoT_1.txt'];
byte_inst = webread(url); %read the content of the URL
whos('byte_inst')
byte_inst(1) %read the old byte instruction, if available
  
```

>> Readweb

Name	Size	Bytes	Class	Attributes
byte_inst	1x1	2	char	

ans = 3

The **webread** function is harmless but the **webwrite** function faces hurdles of firewall and permissions to write to a website. The **webwrite** function is available only in MATLAB R2015a.

The MATLAB code for writing to a website is shown below, which can be used for interacting with a Thing in IoT operation, though not tested: For doing this, you need to open an account on an open source <https://thingspeak.com/>, an open data platform for Internet of Things (IoT) and obtain the following IDs and keys, which will be used in the following code.

ChannelID  
api\_key  
writeApiKey

**MATLAB Code: write2web.m**

```
%MATUINO
%courtesy, http://matlabarduinorobotics.blogspot.com/
thingSpeakURL = 'http://api.thingspeak.com/';
thingSpeakWriteURL = [thingSpeakURL 'update'];
writeApiKey = 'your key';
data = 8;           %new data to be uploaded to the URL
%data = num2str(data);
data = ['api_key=',writeApiKey,'&field1=',data];
response = webwrite(thingSpeakWriteURL,data)
%Check back what was uploaded
channelID = num2str(your channelID);
thingSpeakReadURL = [thingSpeakURL 'channels/' channelID '/fields/field1/last'];
data = webread(thingSpeakReadURL,'your api_key',writeApiKey)
```

**Lab 3- Basics of ESP8266:** This lab presents the basics of the ESP8266 architecture, setting up the board and looking for WiFi to use for the WiFi chip. It uses an Integrated Development Environment (IDE) which involves adding library files that are necessary to run all exercises.

**Lab 4- Soft Access Point:** This lab creates a soft access point. The soft access point is a network that can be connected with a cell phone or a laptop.

**Lab 5- Data Grab:** Connect the chip ESP8266 to a network and run a C code that grabs random data from a webpage that is already saved and displays it on the serial monitor in Arduino. After the code runs the chip connects to the network which in this case is PNW-Media (or any other available network).

```
const char* ssid = "PNW-Media";
```

```
const char* password = "";
```

The following data was grabbed from a website [www.ramiro.info](http://www.ramiro.info) in the city of Indianapolis, IN 59590,:

```
Connecting..{
  "id": 3,
  "name": "Clementine Bauch",
  "username": "Samantha",
  "email": "Nathan@yesenia.net",
  "address": {
    "street": "Douglas Extension",
    "suite": "Suite 847",
    "city": "McKenziehaven",
    "zipcode": "59590-4157",
    "geo": {
      "lat": "-68.6102",
      "lng": "-47.0653"
    }
  },
  "phone": "1-463-123-4447",
  "website": "ramiro.info",
  "company": {
    "name": "Romaguera-Jacobson",
    "catchPhrase": "Face to face bifurcated inter",
    "bs": "e-enable strategic applications"
  }
}
```

The lab also shows how to a) select the data that one wishes to view, b) change users, and c) view their information compared to or along with other users.

### Lab 6- Hello Server:

In this lab, the student creates a web server that displays a text that can be read by signing into the page. The text can be changed in the code as needed.

**Lab 7- Control a LED:** This lab creates a web server that displays two buttons that can be used to control two LEDs connected to the board, see Fig. 5.

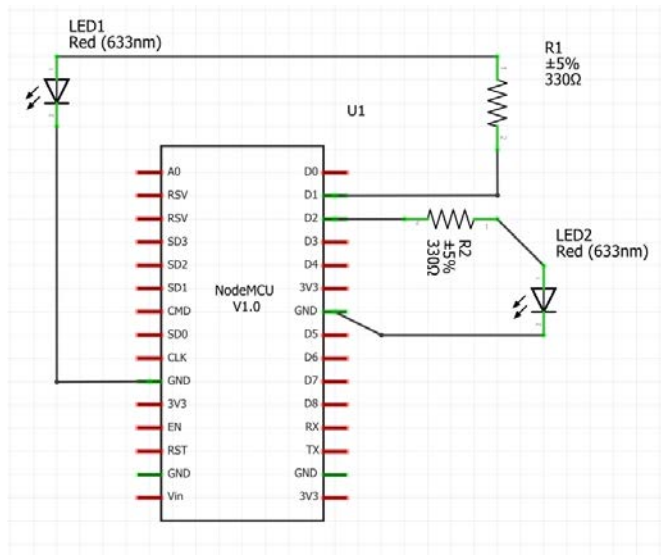


Fig. 5 Controlling LEDs

```

#include <ESP8266WiFi.h>
const char* ssid = "PNW-Medai";
const char* password = "";
// Set web server port number to 80
WiFiServer server(80);
// Variable to store the HTTP request
String header;
...

void loop(){
  WiFiClient client = server.available();
  // Listen for incoming clients
  if (client) {
    Serial.println("New Client.");
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        header += c;
        if (c == '\n') {
          // if the byte is a newline character
          // if the current line is blank, you got two newline characters in a row.
          // that's the end of the client HTTP request, so send a response:
        }
      }
    }
  }
}

```

**Lab 8- World Clock:** This lab displays the time and date for any given region on an OLED screen, ESP32 OLED Driver for SSD1306 Display, see Fig. 6. The time is received from an online server and displayed.

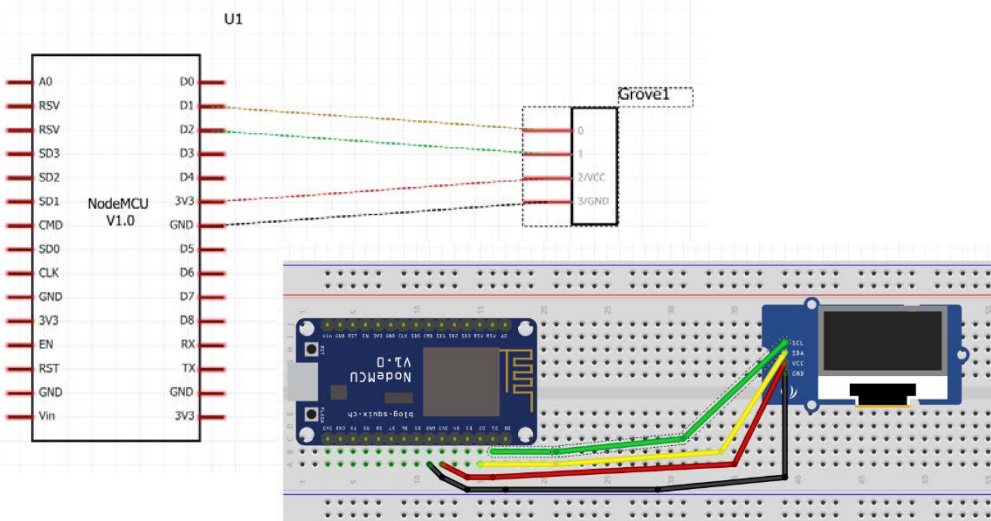


Fig. 6 Displaying the World clock



**Lab 9- Temperature Monitor:** This lab grabs data from the DHT temperature sensor and displays and stores the temperature and humidity on ThingSpeak, Fig. 7a) and b).

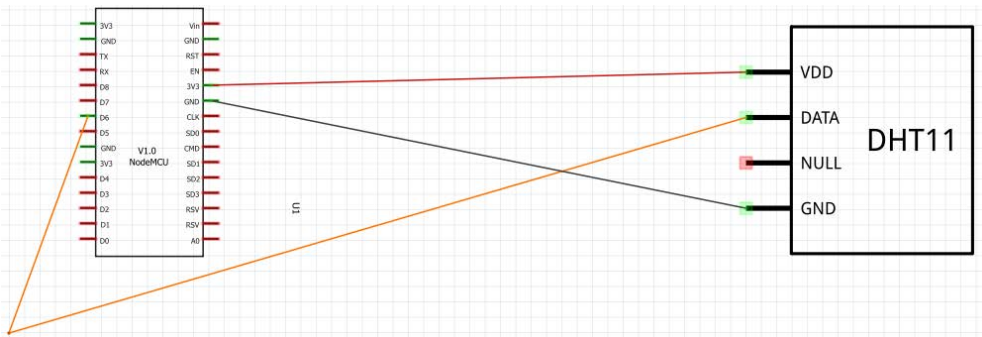


Fig. 7 a) Temperature Monitor schematic

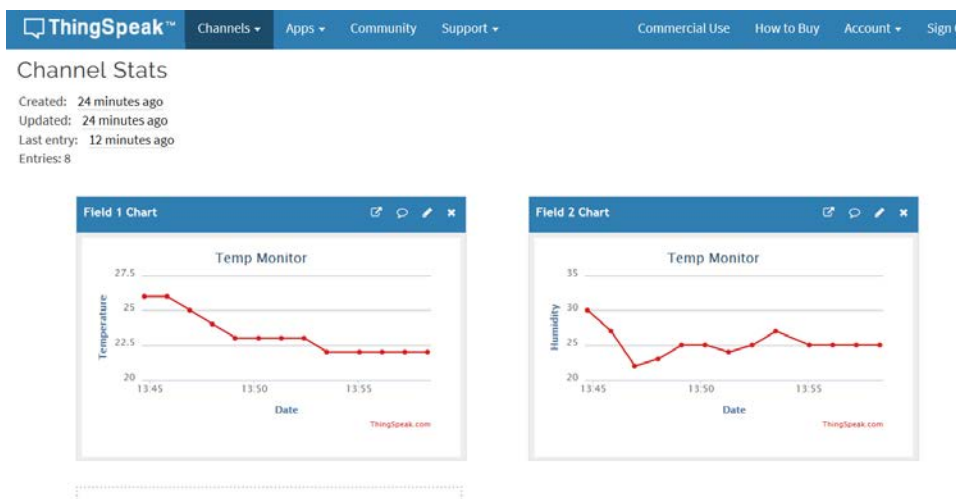


Fig. 7 b) Temperature Monitor display

**Lab 10- Object Tracking:** This lab uses an ultrasonic sensor to collect the distance to an object. The ultrasonic sensor is connected to the ESP8366 on the lab bench and a book is placed in front of the sensor at a distance of approximately 20 cm, as shown in Fig. 8 (a) An application Blynk is used to display the distance from the ultrasonic sensor, as seen in Fig. 8 (b). This application is downloaded on the cell phone.

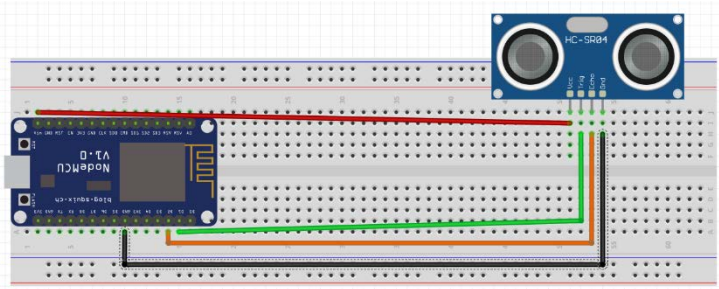
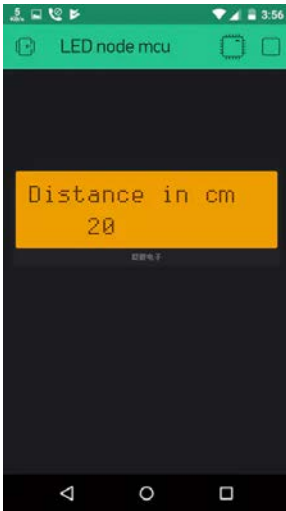


Fig. 8 (a) Object Tracking using ultrasonic sensor

Fig. 8 (b) Distance displayed on the Blynk

**Lab 11- Plane Spotter:** This lab uses an OLED to display data about planes that are flying over certain coordinates, see Fig. 9. The exercise uses the coordinates of planes in the Zurich, Switzerland area. The coordinates can be changed in the code only if the area you want to display is in this link<sup>[7]</sup>.

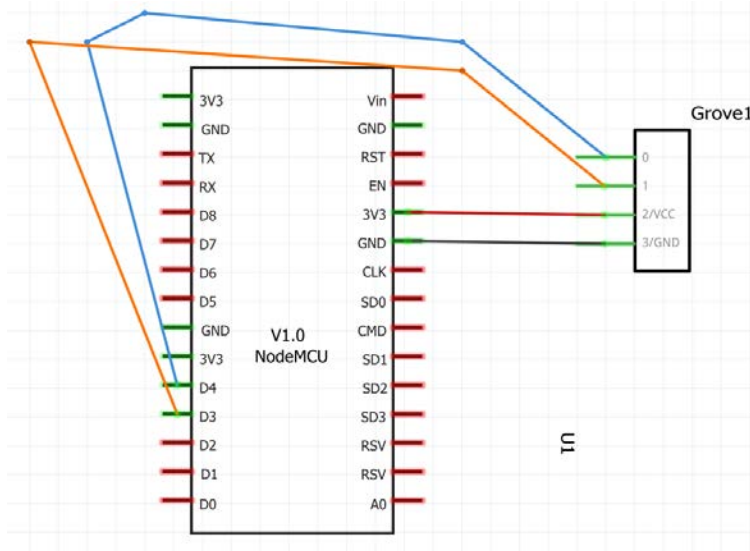


Fig. 9 Plane spotter

The code shows planes in the Zurich, Switzerland area. The coordinates can be changed in the code only if the area you want to display is in this link<sup>[4]</sup>.  
ESP32 OLED Driver for SSD1306 Display.

Copy the coordinates that you want to show the planes in and change the lat (Latitude) and lon (Longitude) in the code:

```
const String QUERY_STRING =  
"lat=47.424341887&lng=8.568778038&fDstL=0&fDstU=10&fAltL=0&fAltL=1  
500&fAltU=10000";
```

**Lab 12- Weather Station:** This lab collects data from a weather map website and displays it on the OLED. Following libraries are required to be installed. Then go to the website <https://openweathermap.org/appid> and sign up to get an API Key, which must be inserted in the code when asked.

```
String OPEN_WEATHER_MAP_APP_ID = "540eba102b6cd279a06e42e1b64ecd6";
```

The current temperature and condition shows along with the next three days and the date and time. The display is not shown here due to the restricted length of the paper.

**ESP8266 and ESP32 Oled Driver for SSD1306 display** by Daniel Eichhorn, Fabrice Weinberg Version 4.0.0 **INSTALLED**  
**A I2C display driver for SSD1306 oled displays connected to an ESP8266 or ESP32** A I2C display driver for SSD1306 oled displays connected to an ESP8266 or ESP32

**Json Streaming Parser** by Daniel Eichhorn Version 1.0.5 **INSTALLED**  
**A very memory efficient library to parse (large) JSON objects on small devices** A very memory efficient library to parse (large) JSON objects on small devices

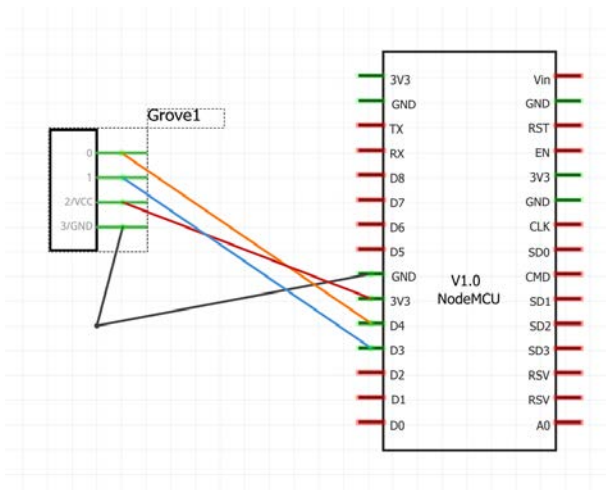
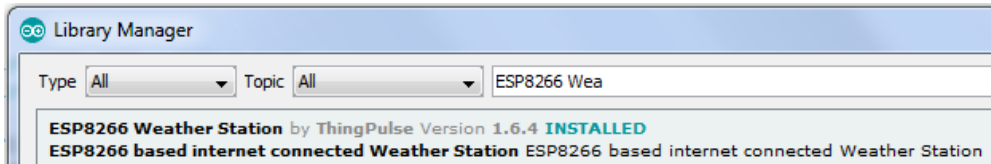


Fig. 10 Weather Station

### 3. CONCLUSION

This paper presents the laboratory exercises for a future course ECET 49900: Internet of Things. The laboratory exercises begin with the basic concepts of Internet of Thing (IoT.) A Wi-Fi enabled ESP8366 IoT unit is introduced. The laboratory exercises are designed to illustrate the capabilities and applications of IoT such as establishing a soft access point, controlling a LED via IoT, grabbing data from a website with permission, displaying the world clock for a specific location, monitoring and logging the temperature and humidity data at a given location, tracking the distance of an object from an ultrasonic sensor, and collecting weather map data for a specific location. These laboratory exercises were designed and demonstrated as part of the Senior Design Project. These exercises are not yet tried in the classroom.

#### Reference:

- [1] [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)
- [2] "Internet of Things (IoT)". gatewaytechnolabs.com.
- [3] "Internet of Things Global Standards Initiative". ITU. Retrieved 26 June 2015.
- [4] <http://blog.squix.org/weatherstation-guide>
- [5] <http://jsonplaceholder.typicode.com>
- [6] <https://www.hackster.io/helloworld1997/ultrasonic-sensor-with-blynk-and-nodemcu-50c074>
- [7] <http://global.adsbexchange.com/VirtualRadar/AircraftList.json?lat=47.437691&lng=8.568854&fDstL=0&fDstU=20&fAltL=0&fAltU=5000>
- [8] <http://www.mathworks.com/help/matlab/internet-file-access.html>
- [9] <http://www.mathworks.com/help/matlab/internet-file-access.html?refresh=true>