



Building Leaders for Advancing Science and Technology (BLAST) 2021 (Summary)

Internet of Things Hands-on Activities with Arduino

Dr. Murat Kuzlu
Department of Engineering Technology

Outline

- Internet
- Internet of Things (IoT)
- Arduino
- ThingSpeak
- Hands-on Activities



<https://medium.com/@otavioguastamacchia/creating-a-simple-iot-case-8102f22908a7>

What is the Internet?



<https://www.connexusuk.com/high-speed-internet/>



- The Internet is a global **web of computers connected to each by communication lines** (mostly phone lines).

- If you look at a map of big cities, smaller towns, and scattered houses, each is connected together with roads, railways, etc. This is similar to the Internet, except with the Internet, wires connect computers.

The Internet is a superhighway.

<http://mediatechnologyeducation.pbworks.com/w/page/20693030/The%20Information%20Superhighway>

Why Internet is Important

- Data, data, data!
- Modern organizations rely on the efficient transmission of data
- Enables distributed systems, **real-time communication**, electronic commerce, social media, and the Web

<https://makeawebsitehub.com/social-media-sites/>



<https://www.edx.org/course/social-media-how-media-got-social>



Ref: Taylor M. Wells: College of Business Administration, California State University, Sacramento
Copyright © 2015 John Wiley & Sons, Inc. All rights reserved

Trends – WoT and IoT

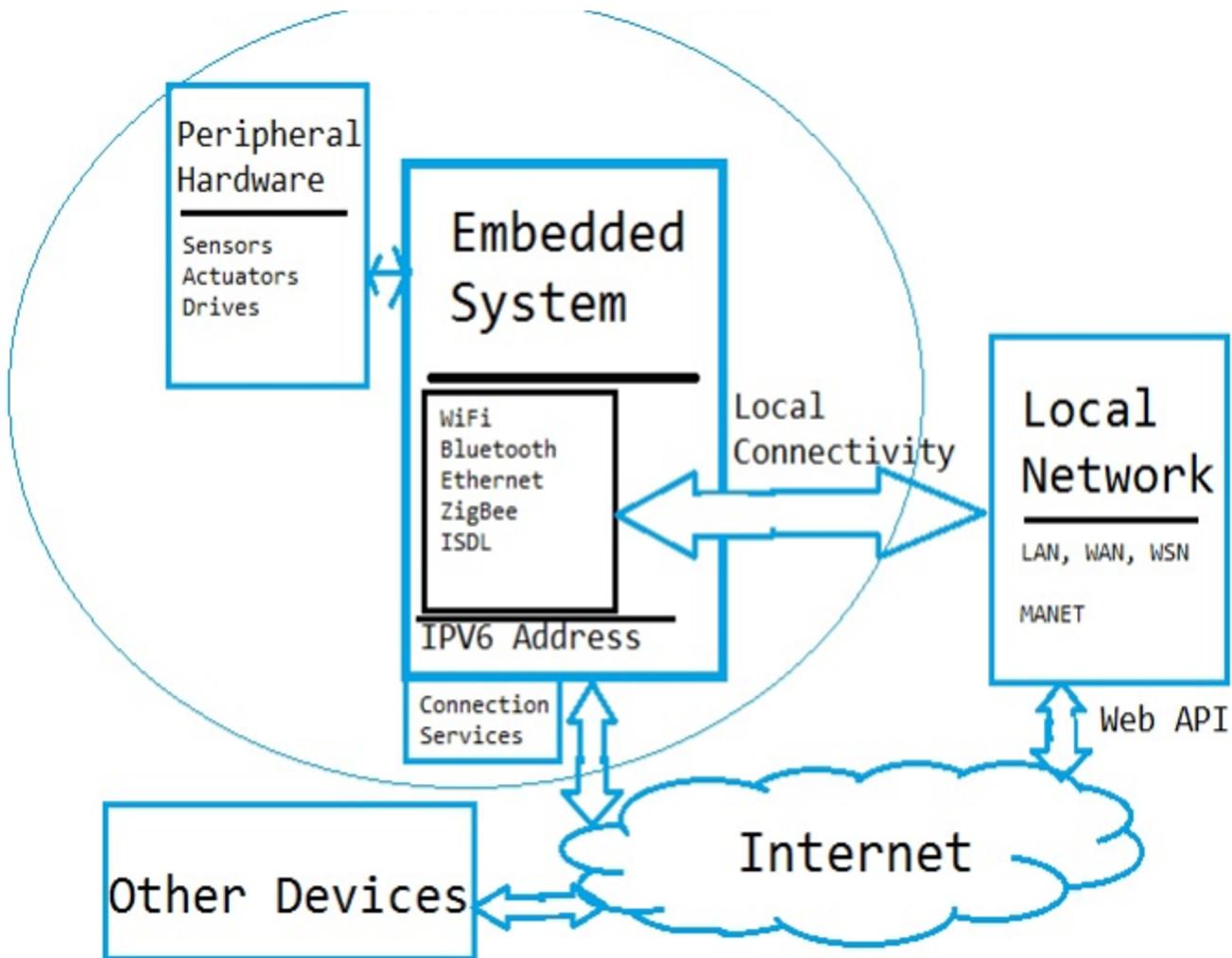
- The Web of Things
 - Everything connects to the network!
 - e.g., cars, refrigerators, thermostats, shoes, doors, etc.
 - Networks need to support the increased demands of these devices
- The Internet of Thing (IoT)
 - *The network of physical objects*—devices, vehicles, buildings and other items-- *embedded with electronics, software, sensors, and network connectivity*—that enables these objects *to collect and exchange data.*"



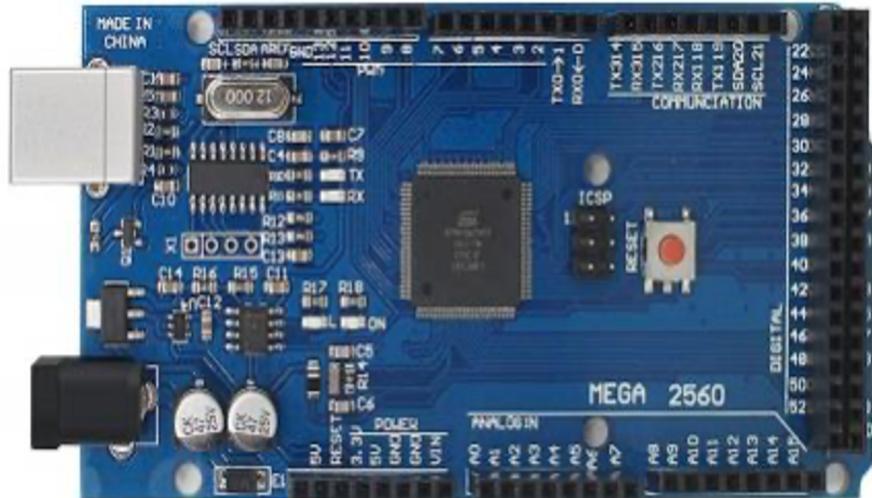
The Internet of Things (IoT)

IoT

- *Network of physical objects*
- *Embedded System*
- *Network connectivity*
- *Collect and exchange data*



Arduino



Mega2560 R3 ATmega2560-16AU
CH340 Development Board



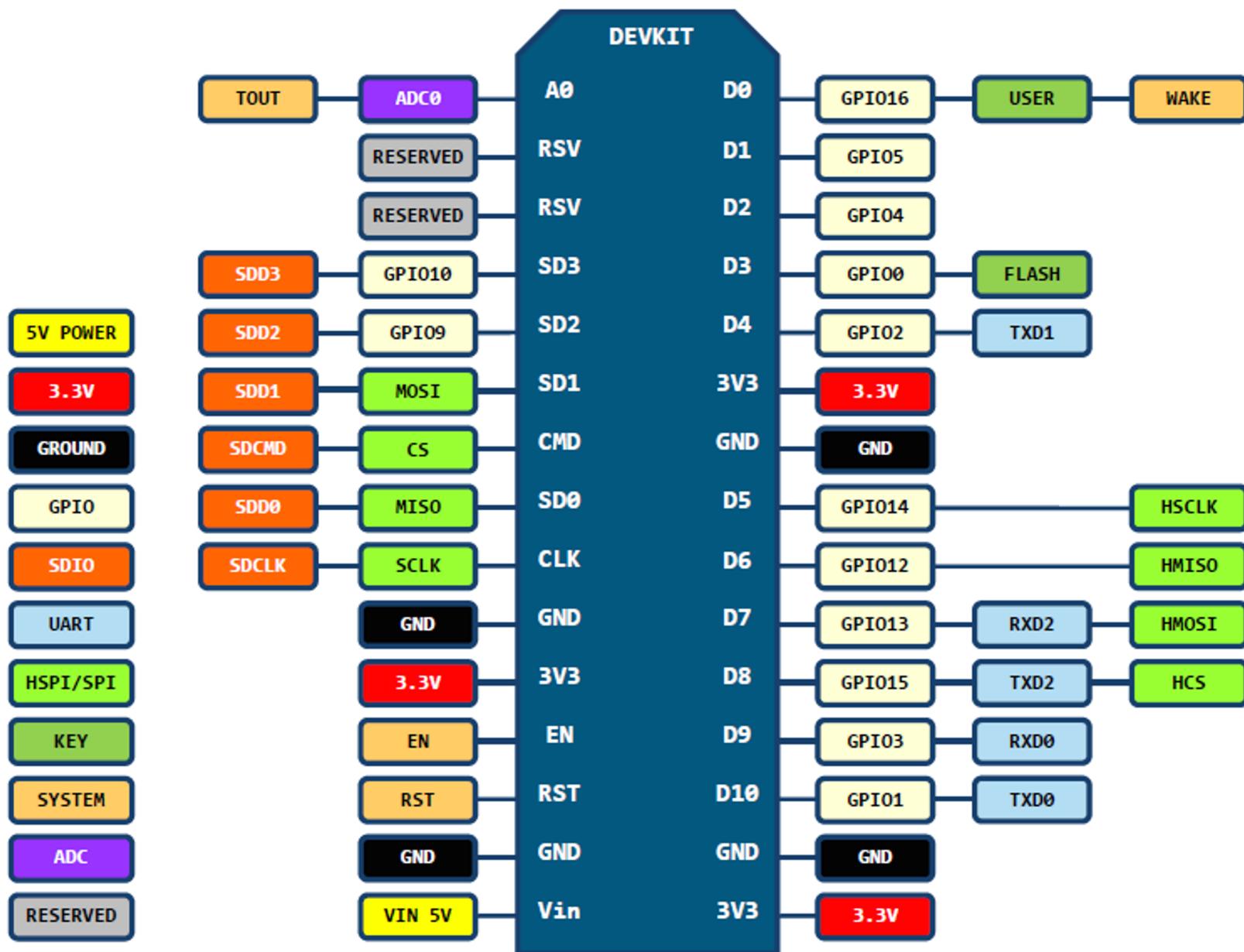
ESP8266 NodeMCU LUA
CP2102 ESP-12E

NodeMCU ESP8266 ESP-12E



- Voltage: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Integrated TCP/IP protocol stack.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5
- Good tutorial:
https://www.handsontec.com/pdf_learn/esp8266-V10.pdf

NodeMCU ESP8266 ESP-12E Pinout

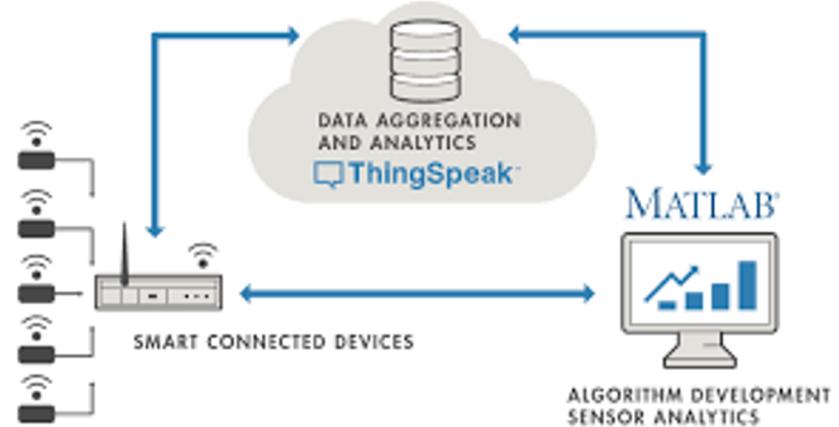


ThingSpeak

- ThingSpeak is an IoT analytics platform service that lets you collect and store sensor data in the cloud and develop Internet of Things applications.
- It works with Arduino, Raspberry Pi and MATLAB (premade libraries and APIs exists).
- But it should work with all kind of Programming Languages, since it uses a REST API and HTTP.



- <https://thingspeak.com>



What is ThingSpeak?

- **What is ThingSpeak:**
ThingSpeak is an IoT analytics platform service that lets you collect and store sensor data in the cloud and develop Internet of Things applications.
- **ThingSpeak Channel:**
'ThingSpeak Channel' is the core element of the ThingSpeak platform. This channel is used to store the real-time data, or the data transferred through various sensors and embedded systems. Data stored at the channel is further used for analysis and visualization.
- Software Requirement: Internet
- Hardware Requirement: Arduino.



This section of the website highlights three main components: "Collect" (send sensor data privately to the cloud), "Analyze" (analyze and visualize data with MATLAB), and "Act" (trigger a reaction). Below these are sections for "ThingSpeak Features" (private/public channels, RESTful MQTT APIs, MATLAB analytics) and "Works With" (Arduino, Particle Photon, Raspberry Pi, etc.).

Creating A Channel

- Before creating a channel, you need to sign into things speak. You can easily sign in either using your either ThingSpeak account or mathswork account, or create a new mathswork account via following link:
- https://thingspeak.com/users/sign_up
- Login Page
- Email: mkuzlu@hotmail.com
- Password: ODU_Blast2000

Create MathWorks Account

Email Address

Missing required information

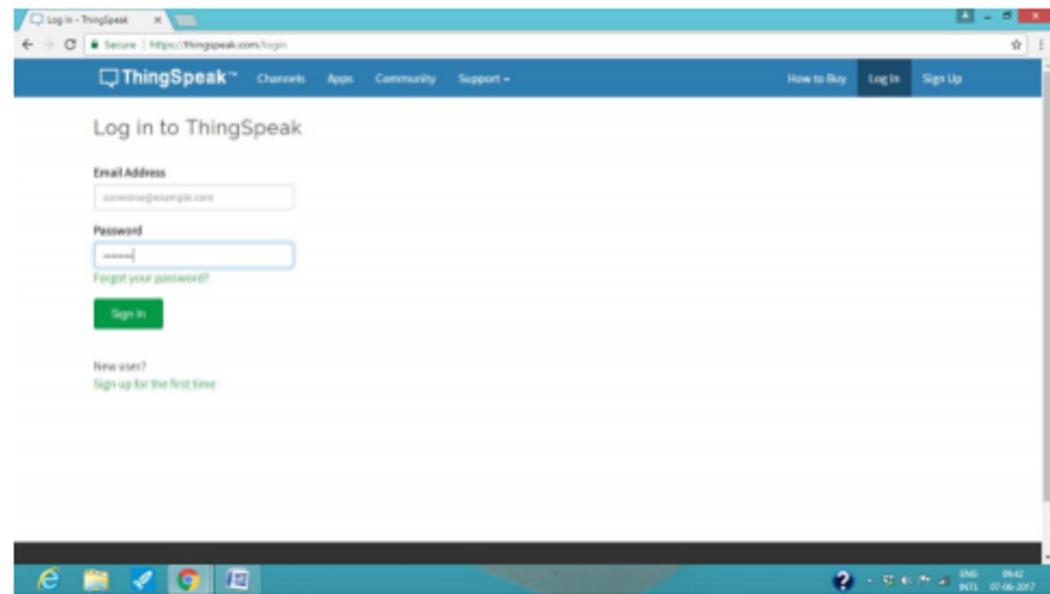
To access your organization's MATLAB license, use your school or work email.

Location

▼

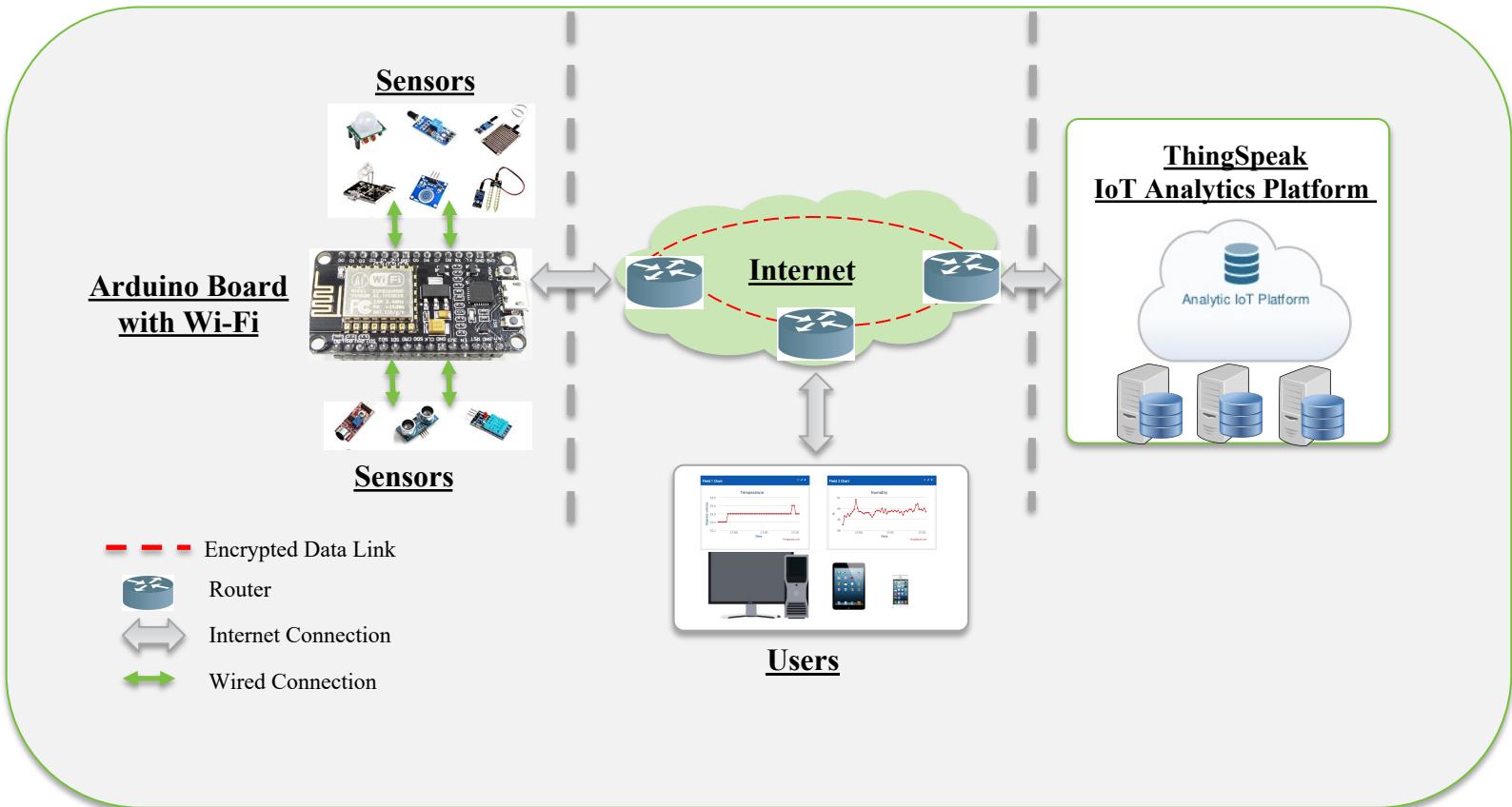
First Name

Last Name

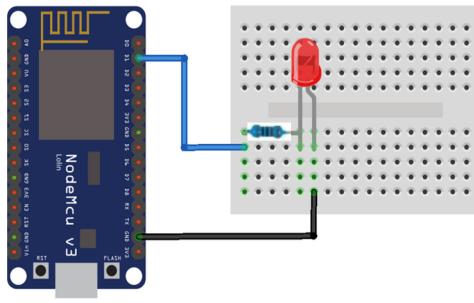


The screenshot shows a web browser window titled "Log In - ThingSpeak". The URL in the address bar is "https://thingspeak.com/login". The page has a blue header with the "ThingSpeak" logo and navigation links for "Channels", "Apps", "Community", and "Support". On the right side of the header, there are buttons for "How to Buy", "Log In", and "Sign Up". Below the header, the main content area is titled "Log in to ThingSpeak". It contains two input fields: "Email Address" with the value "someone@example.com" and "Password" with the value "secret". Below the password field is a link "Forgot your password?". At the bottom of the form is a green "Sign In" button. Further down, there is a "New user?" section with a "Sign up for the first time" link. The browser taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as "08-06-2017".

Architecture of an IoT-based System



Hands-on Activities

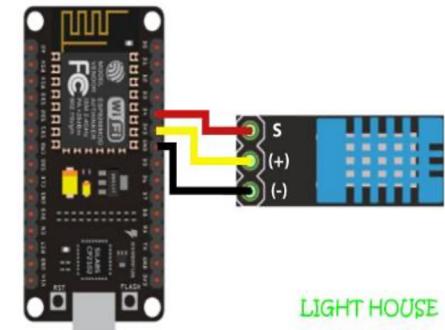


```
#define LED D1 // Led in NodeMCU at pin GPIO5 (D1).
```

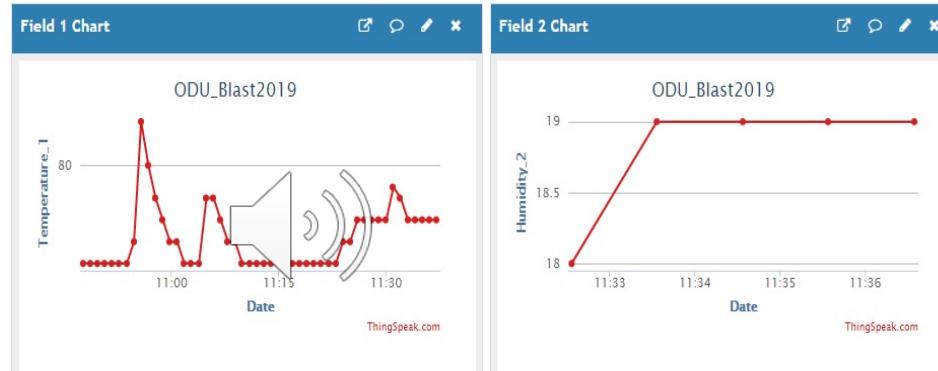
```
void setup()
{
pinMode(LED, OUTPUT); // set the digital pin as output.

}

void loop()
{
digitalWrite(LED, HIGH); // turn the LED off.
delay(1000); // wait for 1 second.
digitalWrite(LED, LOW); // turn the LED on.
delay(1000); // wait for 1 second.
}
```



LIGHT HOUSE



https://github.com/muratkuzlu/ODU_BLAST2021

Hands-on Activity - IV

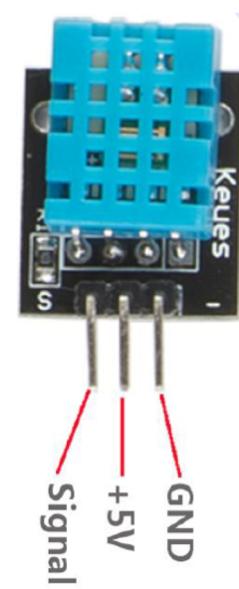
NodeMCU

Temperature & Humidity Sensor DHT11

The DHT11 sensor can detect temperature (C and F) & humidity.

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface.

It has everything it requires built into it, so it will work very well with the NodeMCU. This sensor is used in conjunction with the DHT11 Library.



Wiring a switch with Arduino

We will learn how to set up the DHT11 Humidity and Temperature sensor on your NodeMCU. And learn about how the Humidity sensor works, and how to check output readings from the Serial monitor.

Hardware

- NodeMCU
- DHT11 Humidity and Temperature sensor
- Breadboard
- Jumper Wires (Optional)
- Micro USB Cable

Software

- [Arduino IDE\(version 1.6.4+\)](#)
- [https://github.com/muratkuzlu/ODU_BLAST2021/
tree/main/Projects/4_DHT11](https://github.com/muratkuzlu/ODU_BLAST2021/tree/main/Projects/4_DHT11)

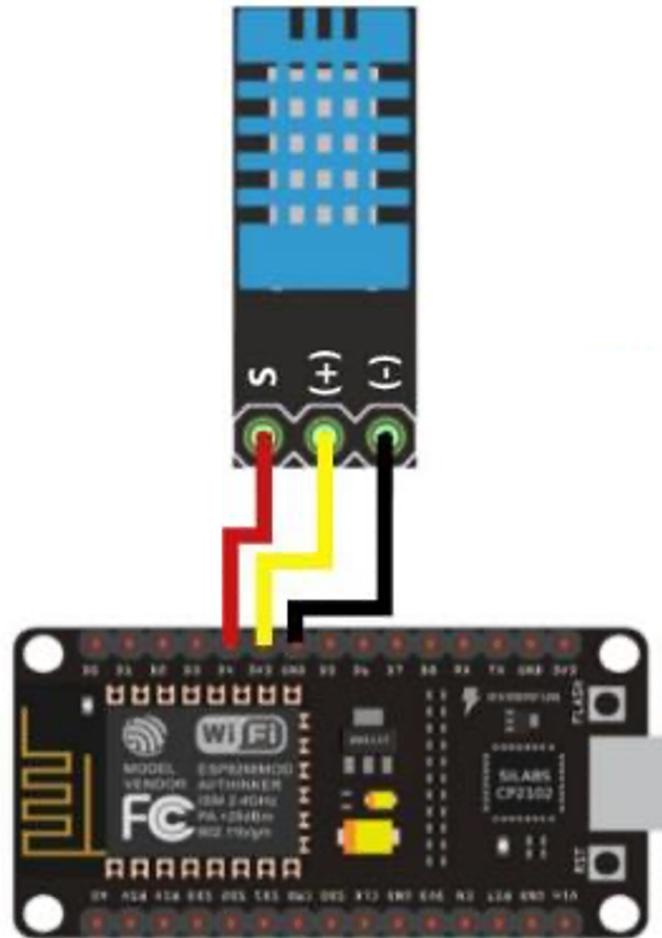
Digital input with Arduino

Set up

Wiring the **DHT11** to the NodeMCU is really easy, but the connections are different depending on which type you have either 3-pins or 4-pins

The **wiring connections** are made as follows:

- **Pin 1** of the DHT11 goes into Digital Pin **D4** of the NodeMCU.
- **Pin 2** of the DHT11 goes into **+3.3v** of the NodeMCU.
- **Pin 3** of the DHT11 goes into Ground Pin (**GND**) of the NodeMCU.



Digital input with Arduino

Code

→ Copy the following code to the IDE

Add the DHTesp.h library

Define Variables

Initialize variables.
Runs once

Used to actively control
the Arduino board. Run
repeatedly

```
#include "DHTesp.h"  
  
int temperature, humidity, k=0, l=0;  
  
#define DHTPIN D4      //pin where the dht11 is connected  
DHTesp dht;  
  
void setup()  
{  
    Serial.begin(115200);  
    delay(10);  
    //dht.begin();  
    dht.setup(DHTPIN, DHTesp::DHT11); // data pin 4  
}  
  
void loop()  
{  
    static boolean data_state = false;  
    float humidity = dht.getHumidity();  
    float temperature = dht.getTemperature();  
    temperature = CelsiusToFahrenheit(temperature);  
    Serial.print("Temperature Value is :");  
    Serial.print(temperature);  
    Serial.println("F");  
    Serial.print("Humidity Value is :");  
    Serial.print(humidity);  
    Serial.println("%");  
    delay(5000);  
}
```

Functions

//Functions

```
float FahrenheitToCelsius(float fahrenheit)
{
    float celsius;
    celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
    return celsius;
}
```

```
float CelsiusToFahrenheit(float celsius)
{
    float fahrenheit;
    fahrenheit = (celsius * 9.0) / 5.0 + 32;
    return fahrenheit;
}
```

NOTE:

When you check the serial monitor make sure the baud rate and the serial begin number in your code is the same.

→ Upload

Digital input with Arduino and IoT - I

Now we are going to connect to IoT

→ Copy the following code to a new IDE sketch

```
#include "DHTesp.h"
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

const char *ssid = "AS2L-Room"; // replace with your wifi ssid and wpa2 key
const char *pass = "as2l214c";
const char* server = "api.thingspeak.com";
const char * myWriteAPIKey = "3M0SBN71PI6UD1A4"; // Enter your Write API key from ThingSpeak
unsigned long myChannelNumber = 803487;
uint8_t temperature, humidity, k=0, l=0;

#define DHTPIN D4      //pin where the dht11 is connected
DHTesp dht;

WiFiClient client;
```

The image shows two screenshots of the ThingSpeak API Keys Settings page. The top screenshot is titled 'Write API Key' and displays a single API key '3M0SBN71PI6UD1A4' with a 'Generate New Write API Key' button. The bottom screenshot is titled 'Channel Settings' and shows details for Channel ID 803487. It includes fields for 'Name' (ODU_Blast2019), 'Description' (Hands-on IOT Activities), and three data fields: 'Field 1' (Temperature), 'Field 2' (Humidity), and 'Field 3' (LED_Status), each with a checked checkbox.

Digital input with Arduino and IoT - II

```
void setup()
{
    Serial.begin(115200);
    delay(10);
    //dht.begin();
    dht.setup(DHTPIN, DHTesp::DHT11); // data pin 2
    Serial.println("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    // Print the IP address
    Serial.println(WiFi.localIP());
    ThingSpeak.begin(client);
}
```

Digital input with Arduino and IoT - III

```
void loop()
{
    static boolean data_state = false;
    float humidity = dht.getHumidity();
    float temperature = dht.getTemperature();
    temperature = CelsiusToFahrenheit(temperature);
    Serial.print("Temperature Value is :");
    Serial.print(temperature);
    Serial.println("F");
    Serial.print("Humidity Value is :");
    Serial.print(humidity);
    Serial.println("%");
    // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different pieces of information in a channel.
    Here, we write to field 1.
    if(temperature < 255)
    {
        k=temperature;
    }
    if(humidity < 255)
    {
        l=humidity;
    }
    if( data_state )
    {
        ThingSpeak.writeField(myChannelNumber, 1, k, myWriteAPIKey);
        data_state = false;
    }
    else
    {
        ThingSpeak.writeField(myChannelNumber, 2, l, myWriteAPIKey);
        data_state = true;
    }
    delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}
```

→ Upload

²⁴ delay(30000); // ThingSpeak will only accept updates every 15 seconds.

Digital output with Arduino and IoT -IV

→ Check ThingSpeak

ODU_Blast2000

Channel ID: 803487

Author: mkuzlu123

Access: Public

Hands-on IOT Activities

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

Add Visualizations

Add Widgets

Export recent data

MATLAB Analysis

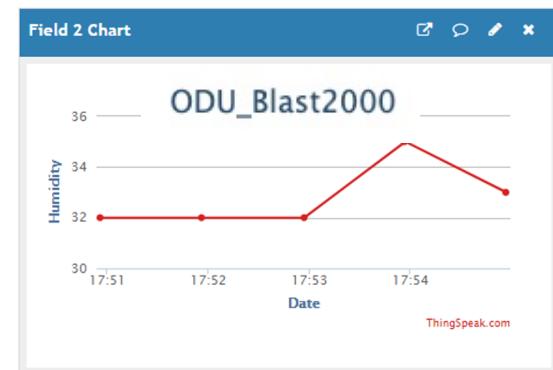
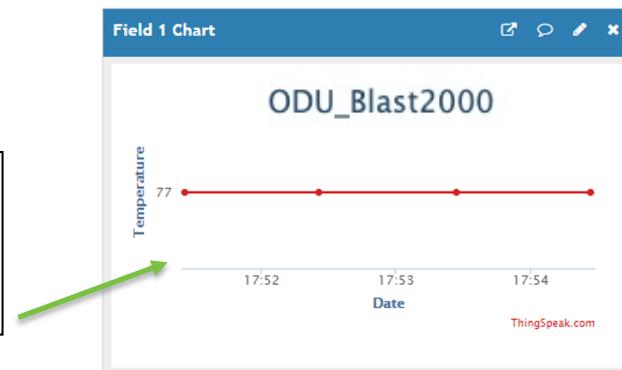
MATLAB Visualization

Channel Stats

Created: about 24 hours ago

Last entry: 5 minutes ago

Entries: 64



Changes in
Temperature and
Humidity



Dr. Murat Kuzlu
mkuzlu@odu.edu