IOT AND CLOUD COMPUTING LAB

By,

Asst. Prof. S.Asra

M.E(Computer science & Engineering)

B.E(Computer Science & Engineering)

Diploma(Computer Science & Engineering)

IOT AND CLOUD COMPUTING LAB

Course	B.Tech.-VI-Sem.	L	T	P	C
Course Code	22CSPC64	-	-	2	1

Course Outcomes (COs) & CO-PO Mapping (3-Strong; 2-Medium; 1-Weak Correlation)

COs	Upon completion of course the students will be able to	PO4	PO5	PO9	PSO2
CO1	identify various IoT devices	3	3	3	3
CO2	use IoT devices in various applications	3	3	3	3
CO3	develop automation work-flow in IoT enabled cloud environment	3	3	3	3
CO4	take part in practicing and monitoring remotely	3	3	3	3
CO5	make use of various IoT protocols in cloud	3	3	3	3

List of Experiments

Week	Title/Experiment
1	Install necessary software for Arduino and Raspberry Pi.
2	Familiarization with Arduino and Raspberry Pi board.
3	Write a program to transfer sensor data to a Smartphone using Bluetooth on Arduino.
4	Write a program to implement RFID using Arduino.
5	Write a Program to monitor temperature and humidity using Arduino and Raspberry Pi.
6	Write a Program to interface IR sensorswith Arduino using IoT Cloud Application.
7	Write a Program to upload temperature and humidity data to the cloud using an Arduino or Raspberry Pi.
8	Write a program to retrieve temperature and humidity data from the cloud using Arduino and Raspberry Pi.
9	Write a program to create a TCP server on cloud using Arduino and respond with humidity data to the TCP client when requested.
10	Write a program to create a UDP server on cloud using Arduino and respond with humidity data to the UDP client when requested.

References

1. IoT and Cloud Computing Lab Manual, Department of CSE, CMRIT, Hyd.

Micro-Projects: Student should submit a report on one of the following/any other micro-project(s) approved by the lab faculty before commencement of lab internal examination.

1. Air Pollution Meter.
2. Smart Garbage Collector.
3. Weather monitoring system.
4. Baggage Tracker.
5. Circuit Breakage Detection.
6. Anti-Theft Flooring System.
7. IoT Based Smart Street Light.
8. IoT based Gas Leakage Monitoring system.
9. IoT Based Smart Irrigation System.
10. IoT Based Water Level Monitoring System.

WEEK-6

Aim: Write a program to interface IR Sensor with Arduino using IoT Cloud application.

IOT-Cloud Application:

THINGSPEAK

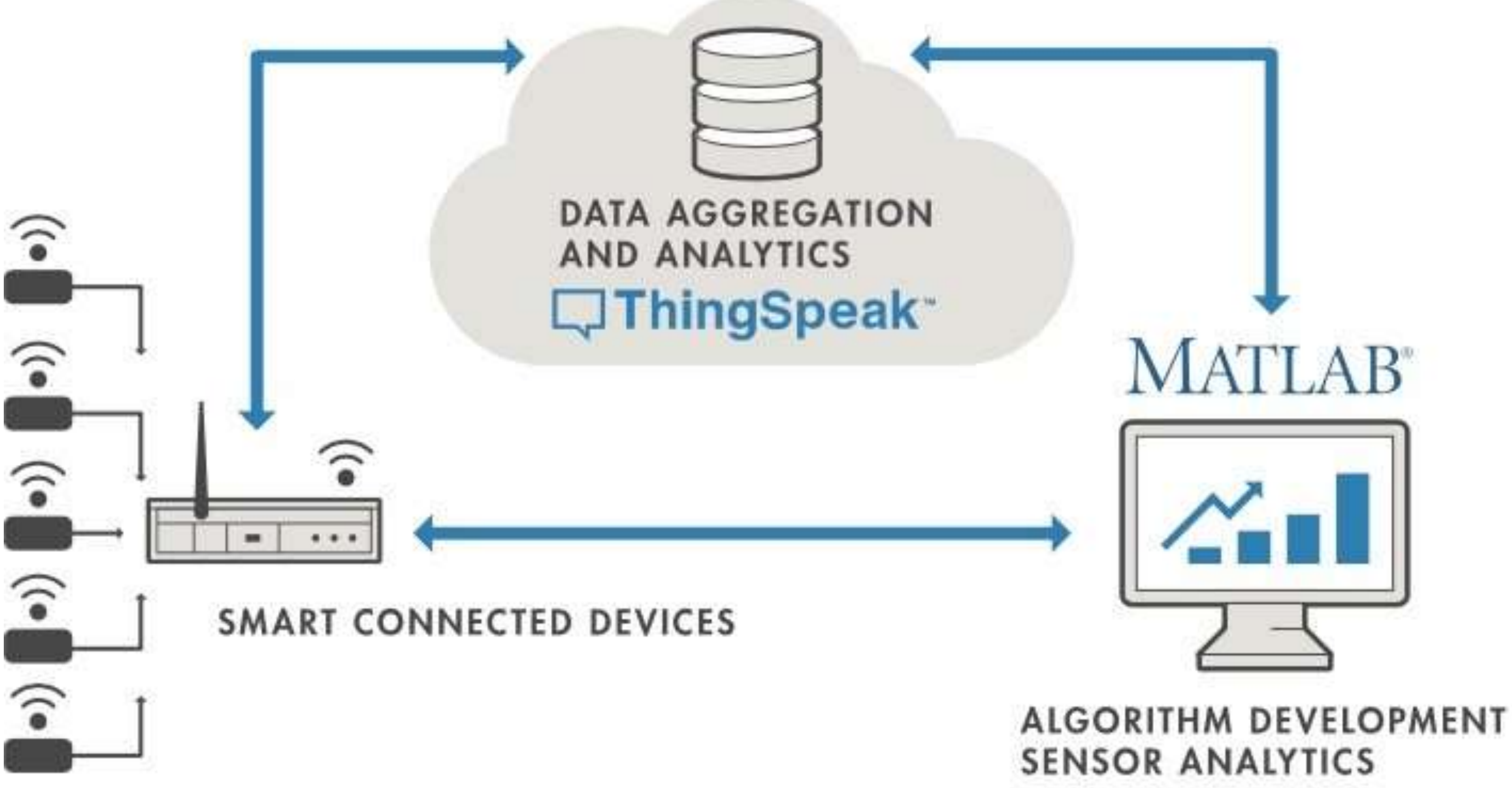
ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our Privacy

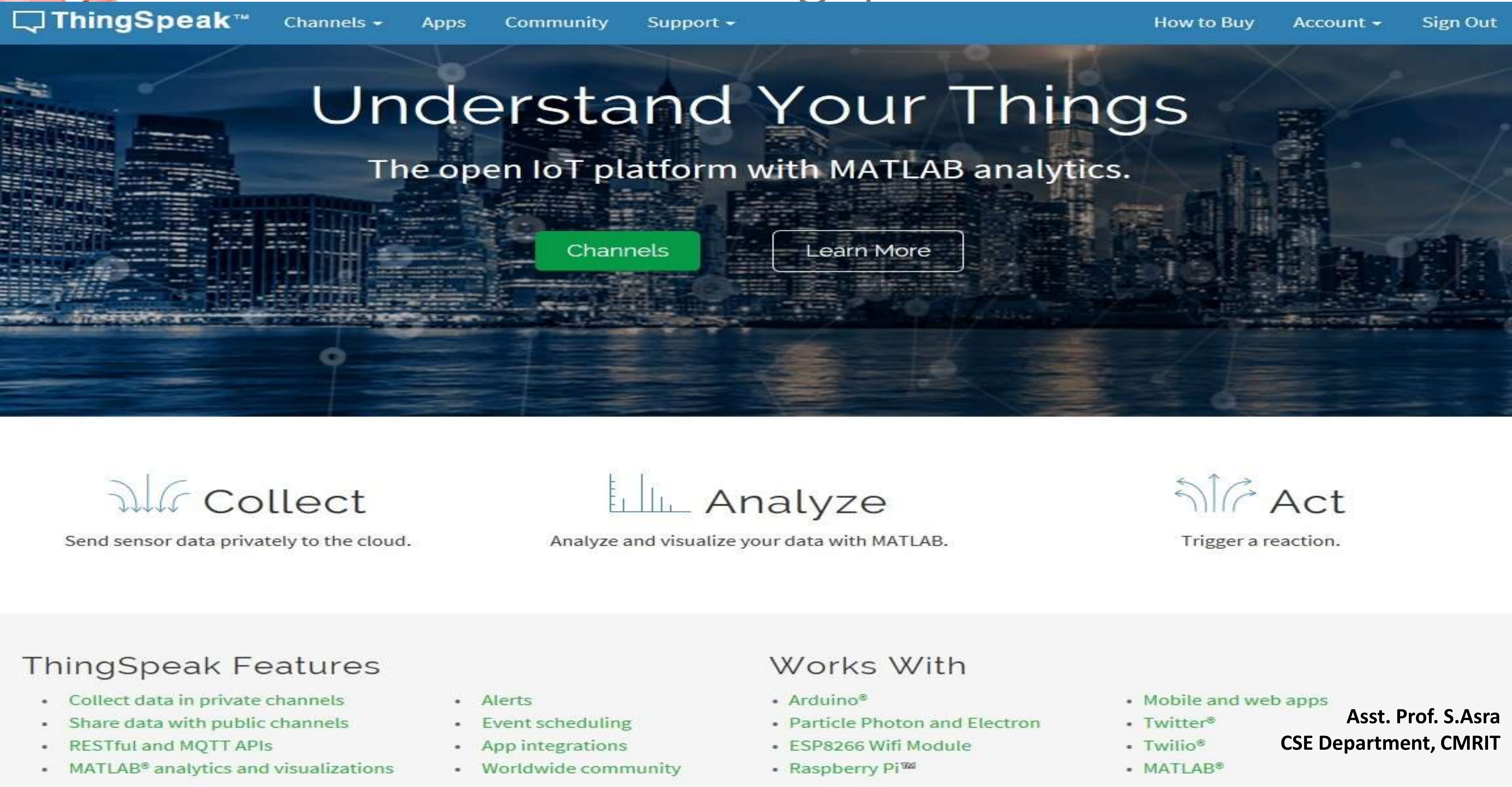
THINGSPEAK

- ThingSpeak is a Web Service (REST API) 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.



Asst. Prof. S.Asra
CSE Department, CMRIT

What is ThingSpeak?



Understand Your Things

The open IoT platform with MATLAB analytics.

Collect

Send sensor data privately to the cloud.

Analyze

Analyze and visualize your data with MATLAB.

Act

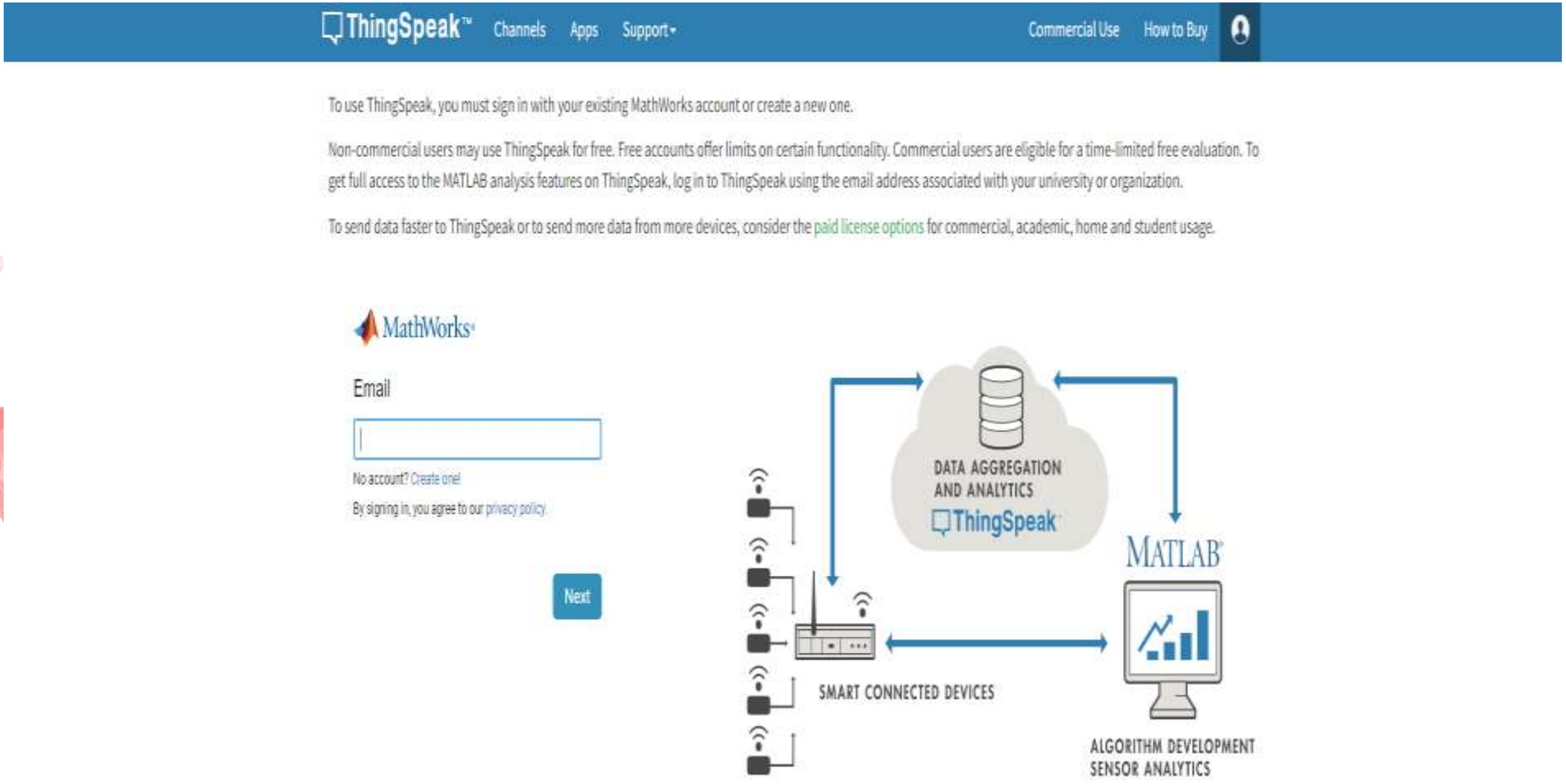
Trigger a reaction.

ThingSpeak Features

- Collect data in private channels
- Share data with public channels
- RESTful and MQTT APIs
- MATLAB® analytics and visualizations
- Alerts
- Event scheduling
- App integrations
- Worldwide community

Works With

- Arduino®
- Particle Photon and Electron
- ESP8266 Wifi Module
- Raspberry Pi™
- Mobile and web apps
- Twitter®
- Twilio®
- MATLAB®



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

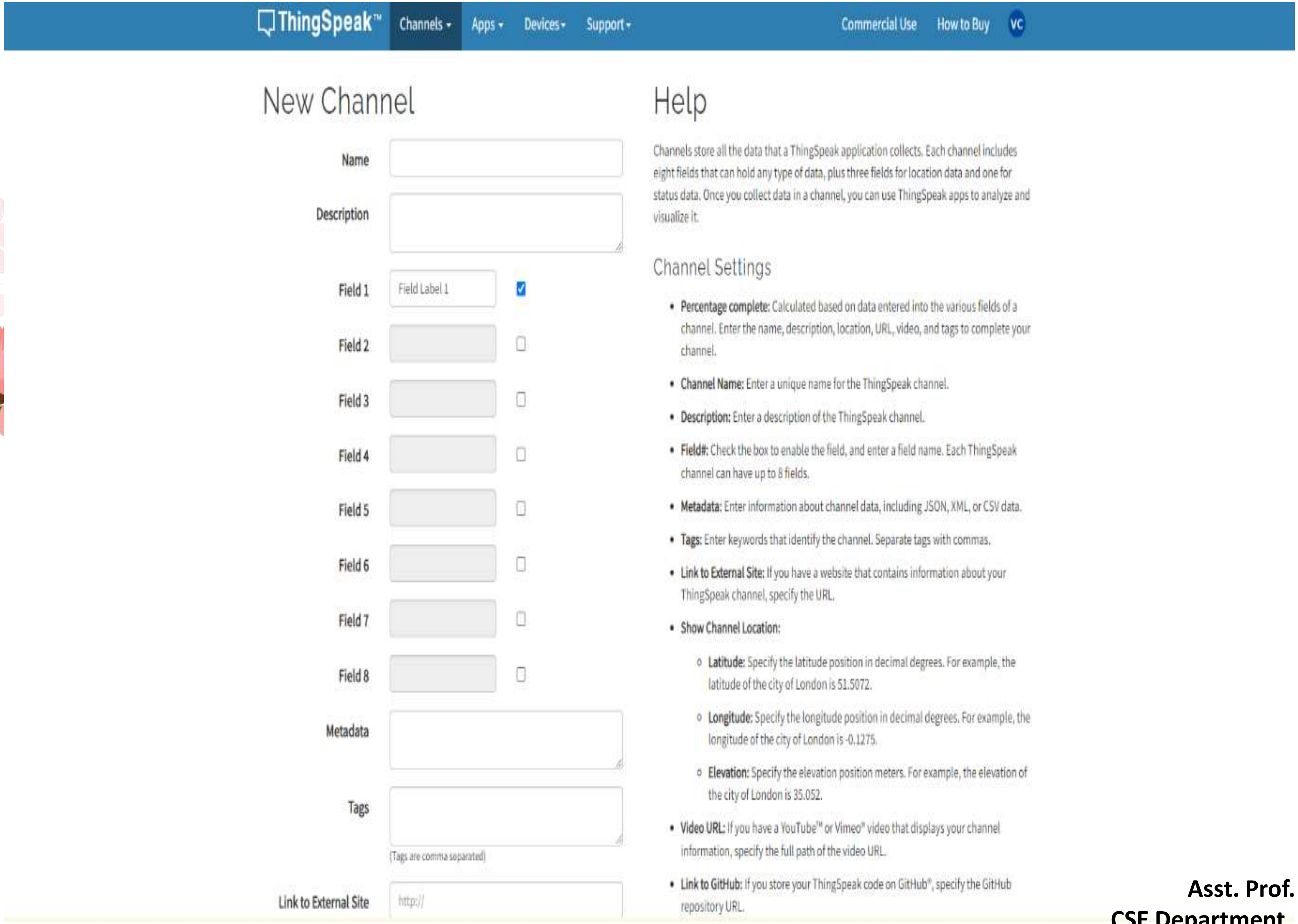
Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the paid license options for commercial, academic, home and student usage.

Email

No account? Create one!

By signing in, you agree to our privacy policy.



New Channel

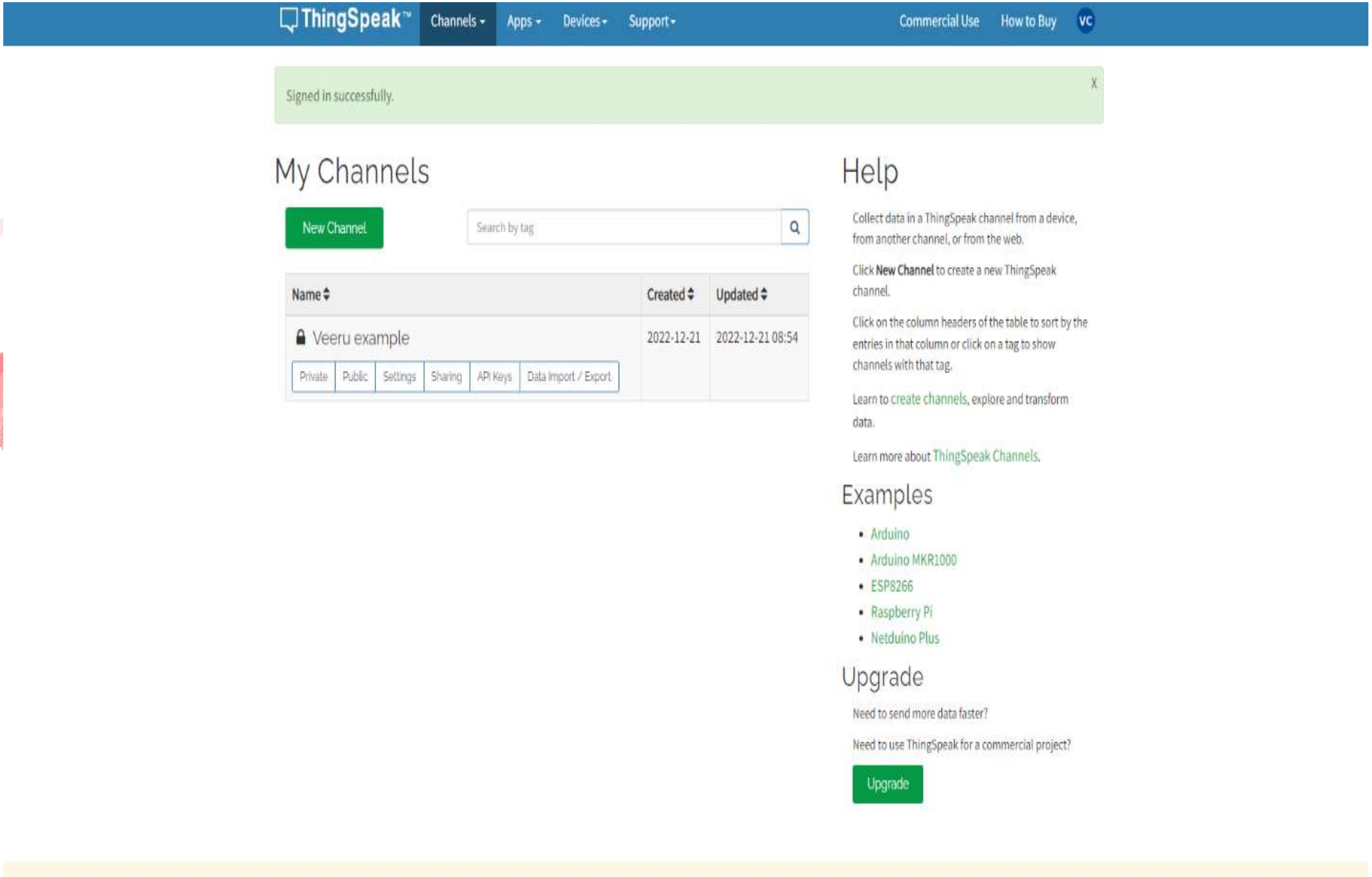
Name		
Description		
Field 1	Field Label 1	☑
Field 2		□
Field 3		□
Field 4		□
Field 5		□
Field 6		□
Field 7		□
Field 8		□
Metadata		
Tags	(Tags are comma separated)	
Link to External Site	http://	

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- **Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- **Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.



Signed in successfully.

My Channels

Name	Created	Updated
Veeru example	2022-12-21	2022-12-21 08:54

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

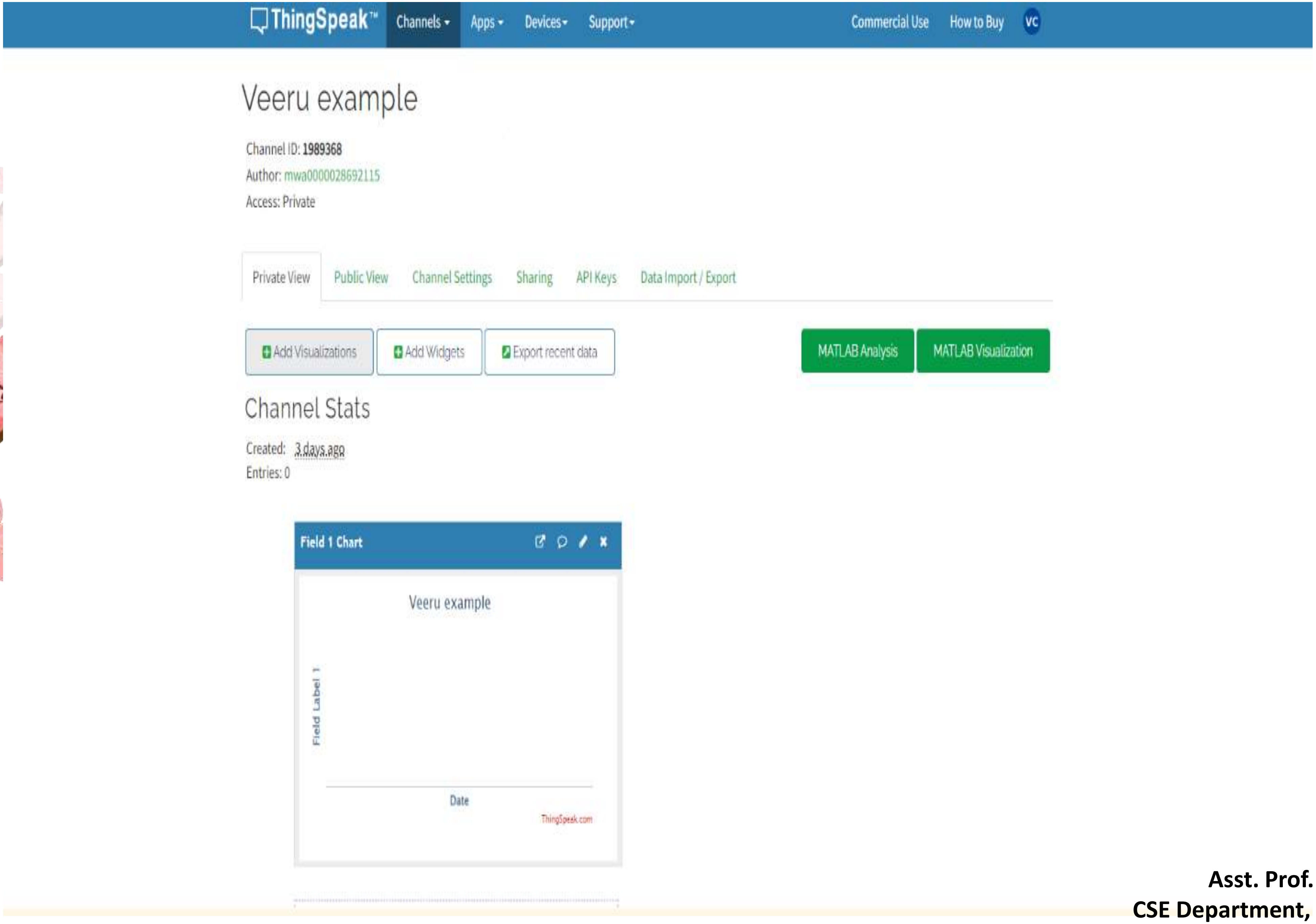
Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial project?



Veeru example

Channel ID: **1989368**

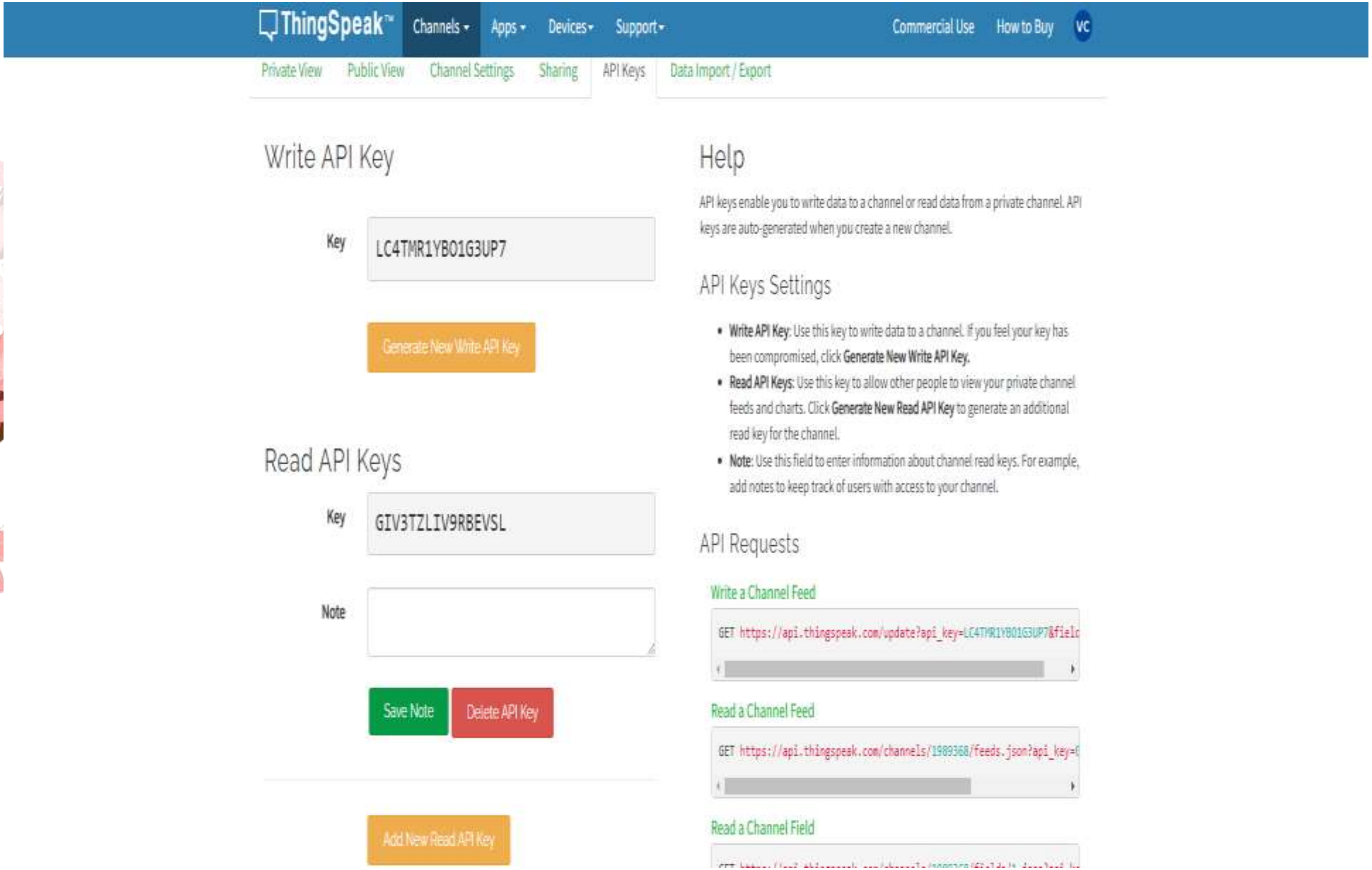
Author: mwa0000028692115

Access: Private

Channel Stats

Created: 3 days ago

Entries: 0



Write API Key

Key: LC4TMR1YBO1G3UP7

Read API Keys

Key: GIV3TZLIV9RBEVSL

Note:

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=LC4TMR1YBO1G3UP7&field
```

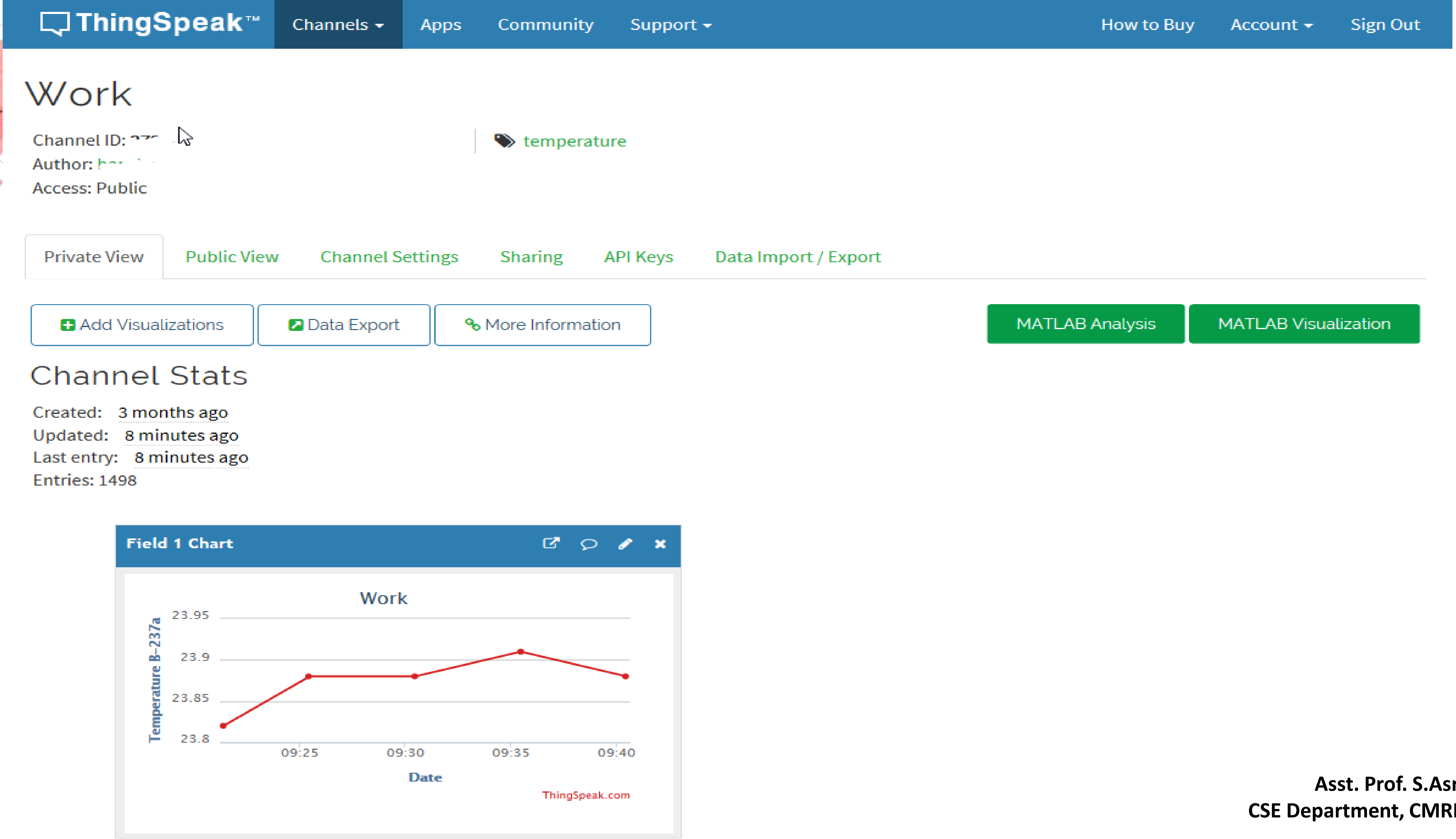
Read a Channel Feed

```
GET https://api.thingspeak.com/channels/1989368/feeds.json?api_key=G
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels/1989368/fields/1.json?api_k
```

Asst. Prof. S.Asra
CSE Department, CMRIT

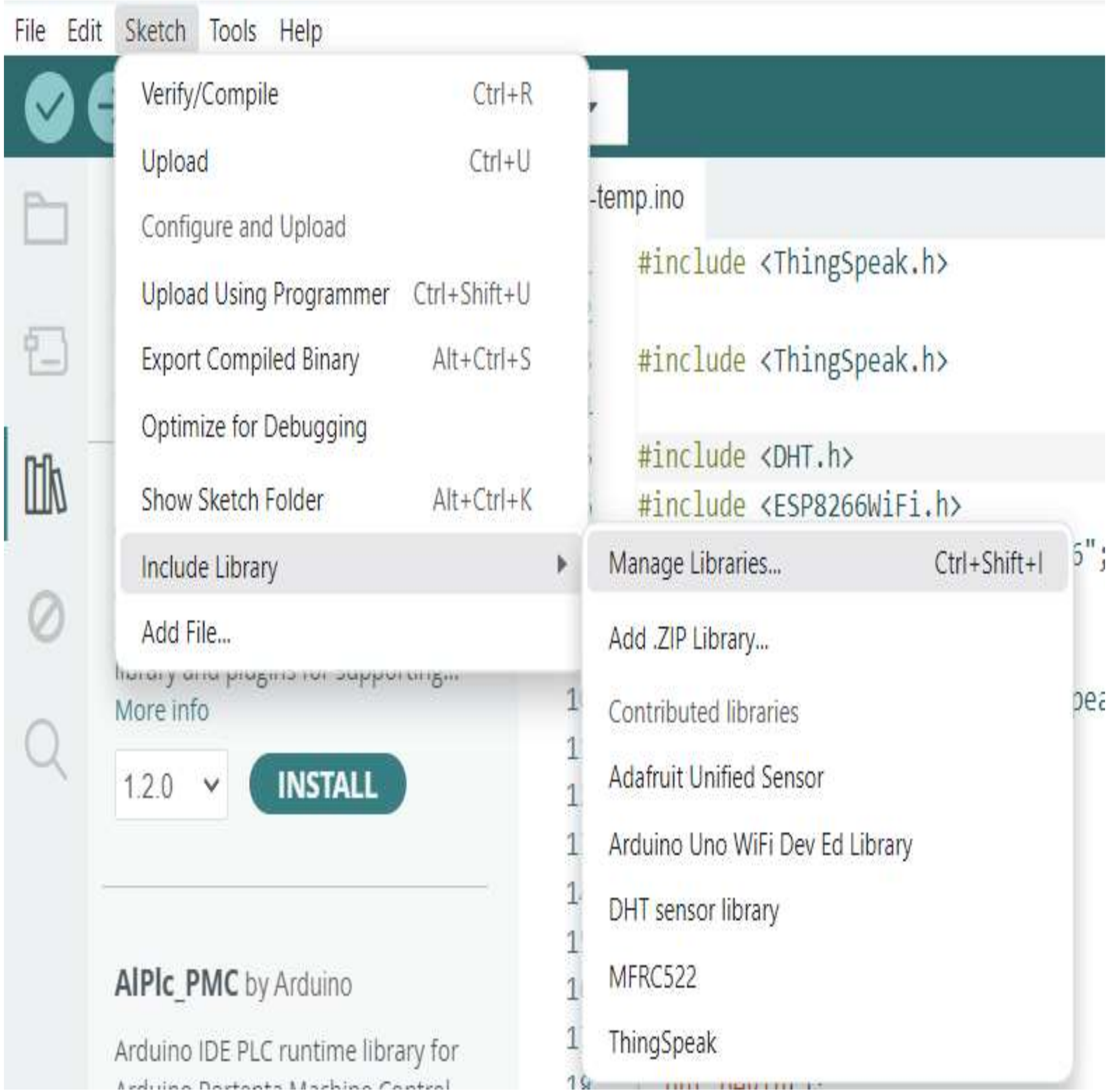


Work

Channel ID: [redacted]
Author: [redacted]
Access: Public

Channel Stats

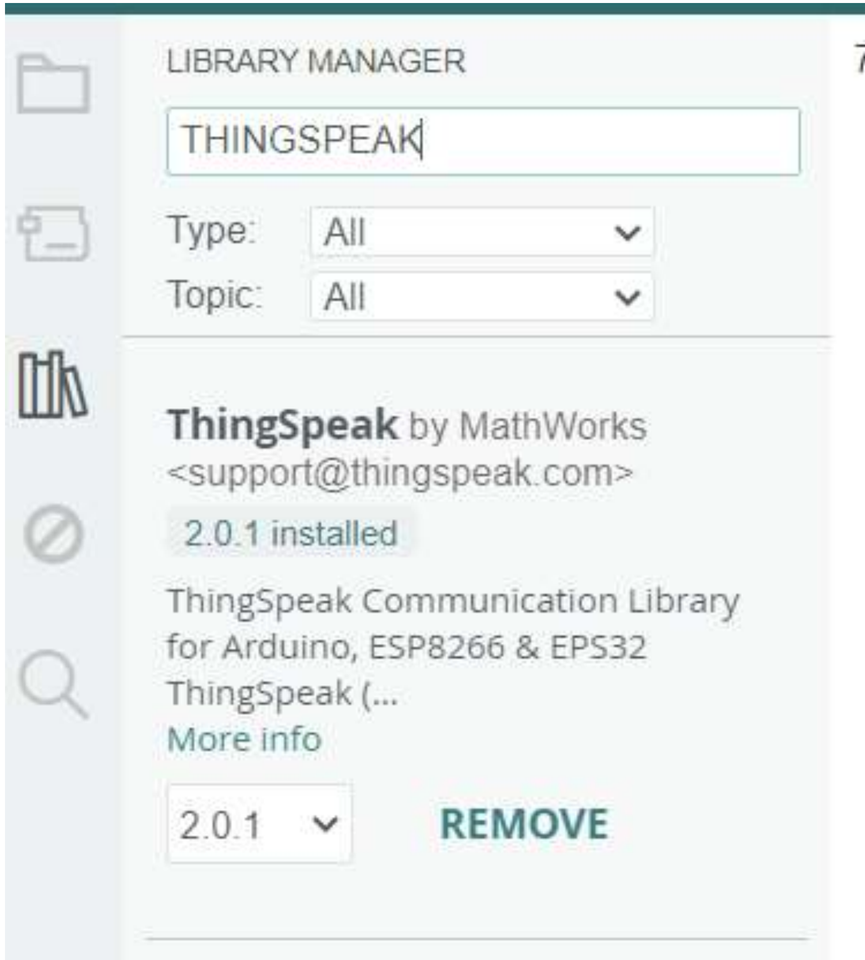
Created: 3 months ago
Updated: 8 minutes ago
Last entry: 8 minutes ago
Entries: 1498



```
#include <ThingSpeak.h>

#include <ThingSpeak.h>

#include <DHT.h>
#include <ESP8266WiFi.h>
```



THINGSPEAK

Type: All

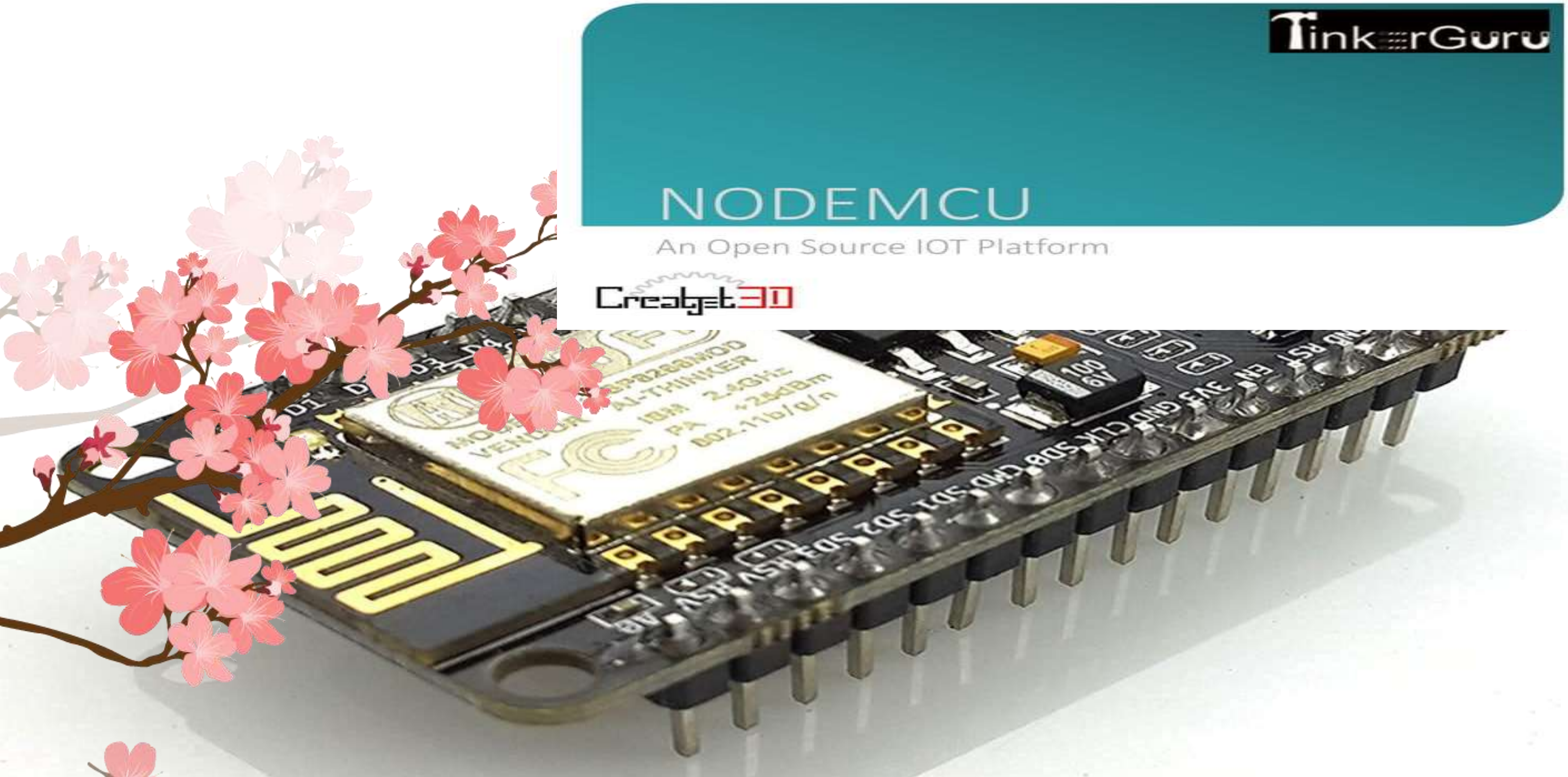
Topic: All

ThingSpeak by MathWorks
<support@thingspeak.com>

2.0.1 installed

ThingSpeak Communication Library for Arduino, ESP8266 & EPS32 ThingSpeak (...
More info

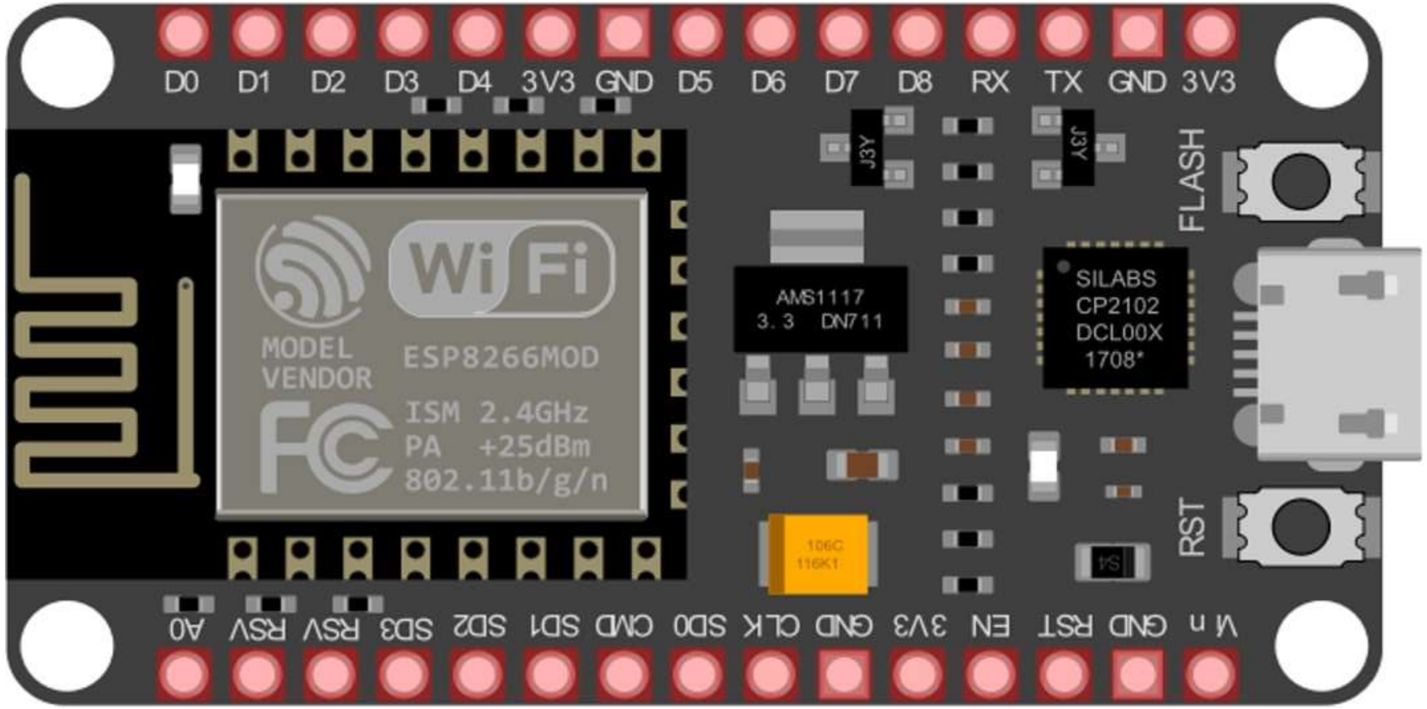
2.0.1



Asst. Prof. S.Asra
CSE Department, CMRIT

NODEMCU | Overview

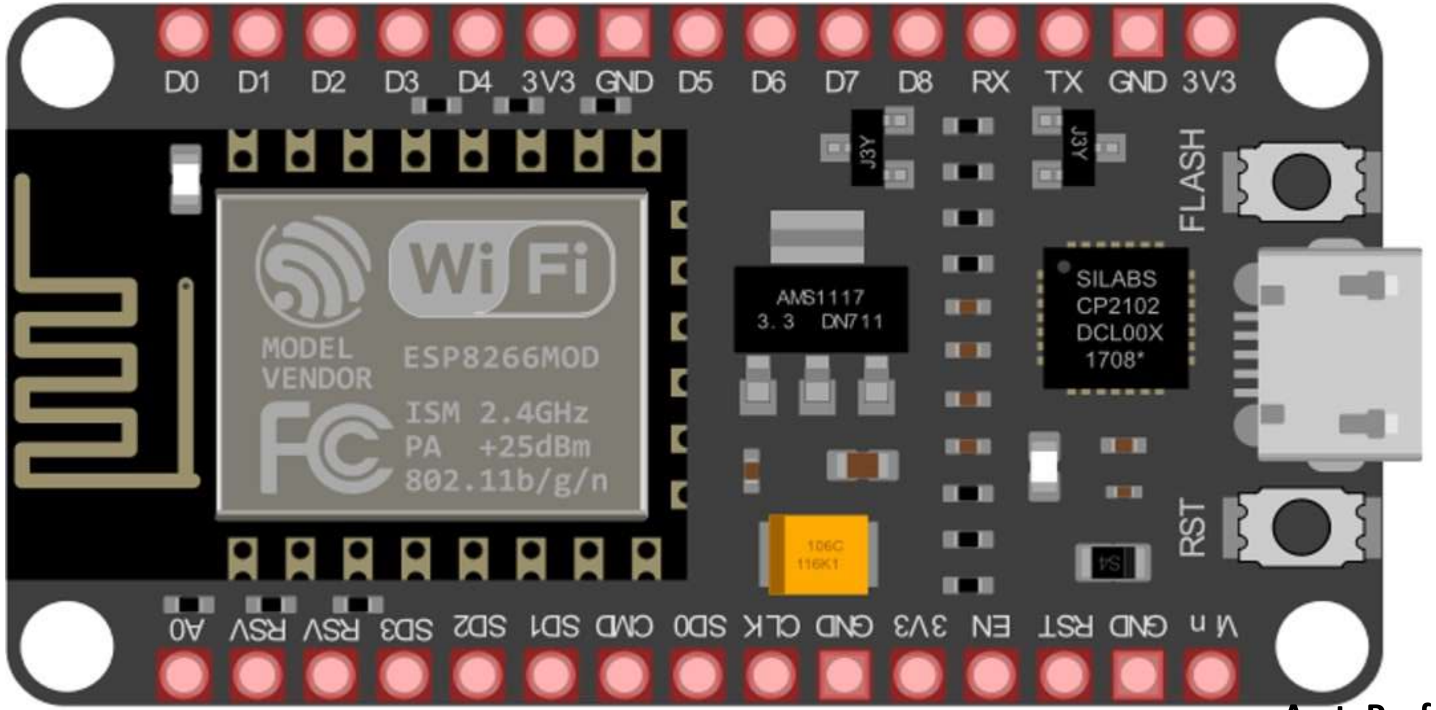
- The NodeMcu is an open-source firmware and development kit that helps you to Prototype your IOT product within a few Lua script lines.
- Power:
  - Input Voltage : 3.3V
  - DC Current : 250mA
- Memory:
  - RAM 32Kb
  - DRAM 80Kb
  - Flash 200Kb



NODEMCU | What is it?

The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266.

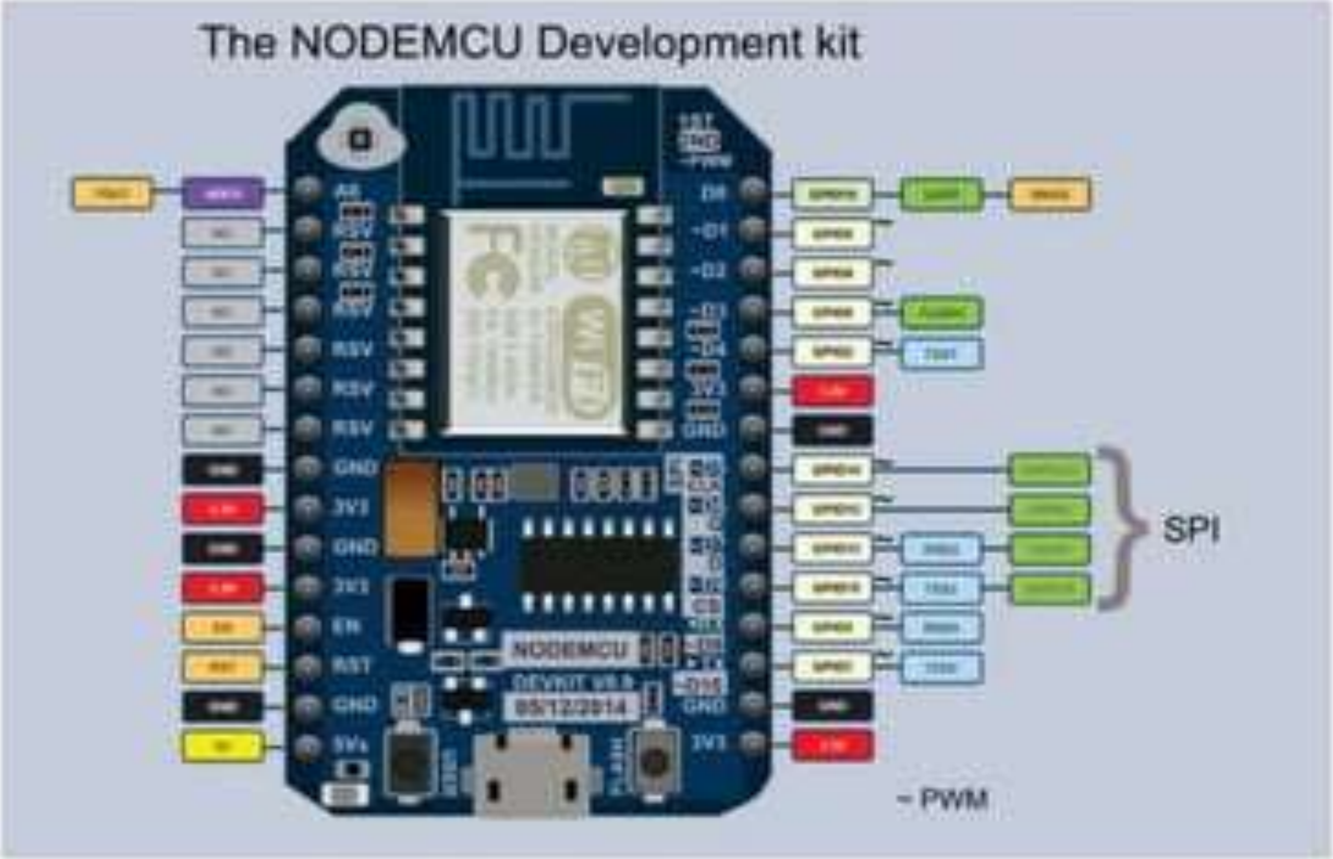
- An Arduino-like device
- Main component: ESP8266
- With programmable pins
- And built-in wifi
- Power via USB
- Low cost



Asst. Prof. S.Asra
CSE Department, CMRIT

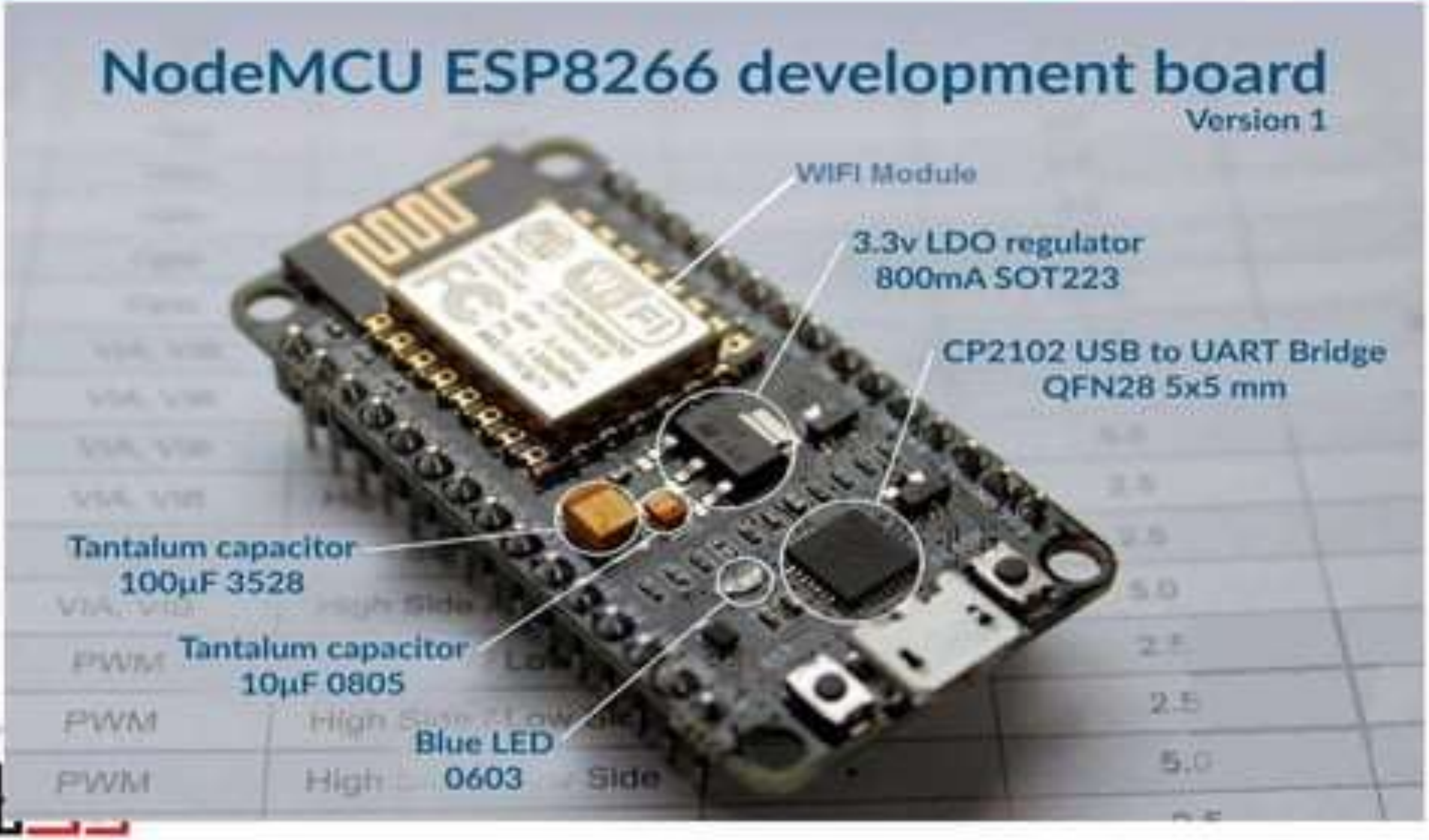
NODEMCU | What you can do with it?

- Program it via C or LUA
- Access it via wifi (ex. HTTP)
- Connect pins to any device (in or out)



The NODEMCU Development kit

NODEMCU | MAIN COMPONENT

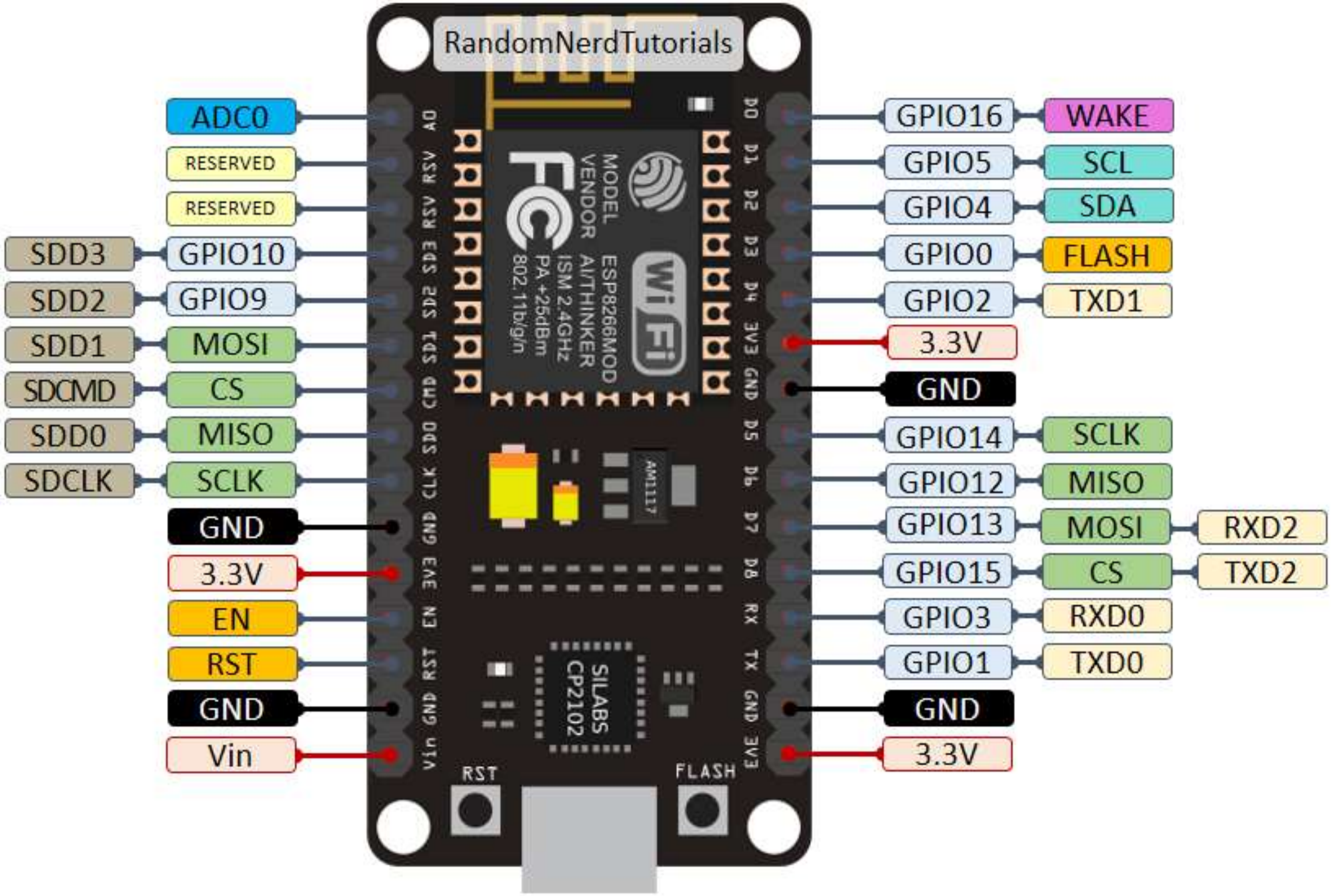


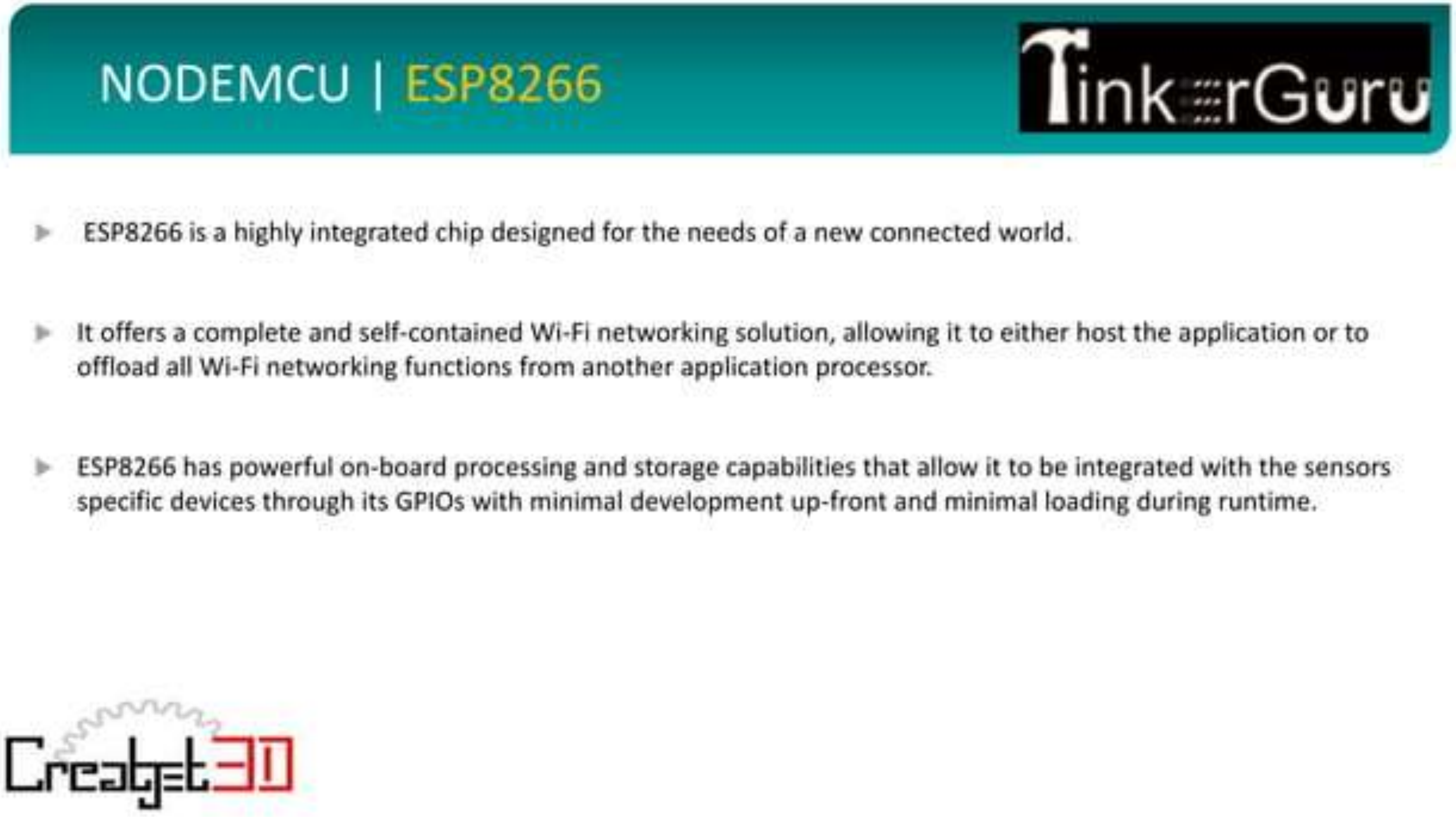
NodeMCU ESP8266 development board

Version 1

NODEMCU | PIN DESCRIPTION

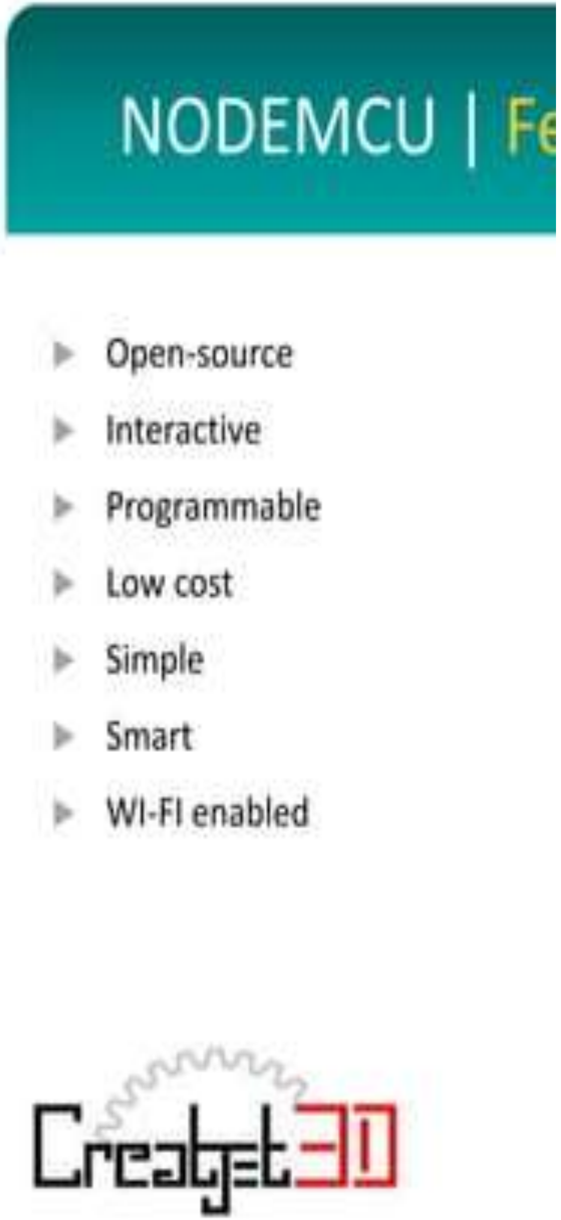
TinkerGuru





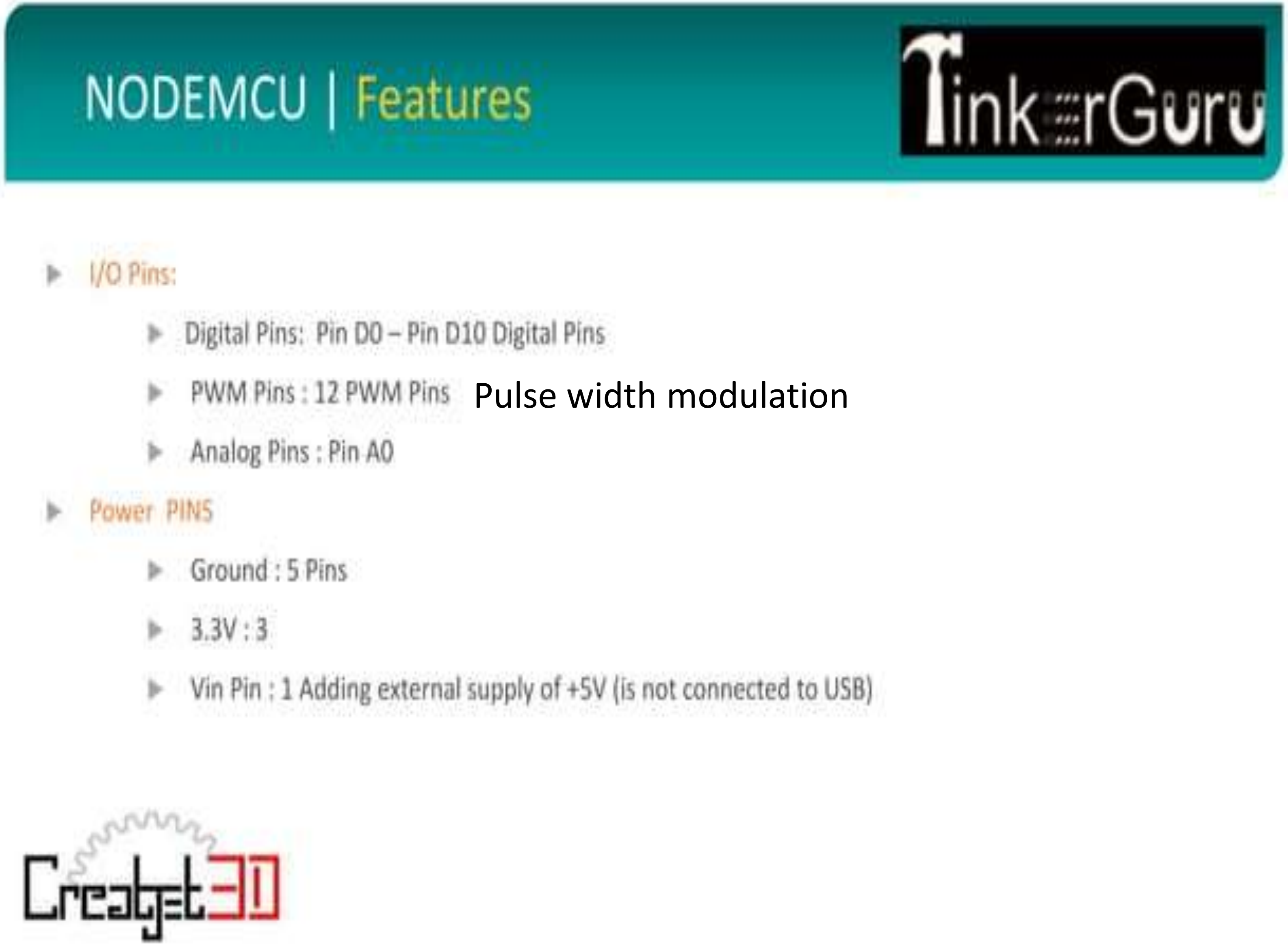
NODEMCU | ESP8266

- ESP8266 is a highly integrated chip designed for the needs of a new connected world.
- It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.
- ESP8266 has powerful on-board processing and storage capabilities that allow it to be integrated with the sensors specific devices through its GPIOs with minimal development up-front and minimal loading during runtime.



NODEMCU | Fe

- Open-source
- Interactive
- Programmable
- Low cost
- Simple
- Smart
- WI-FI enabled

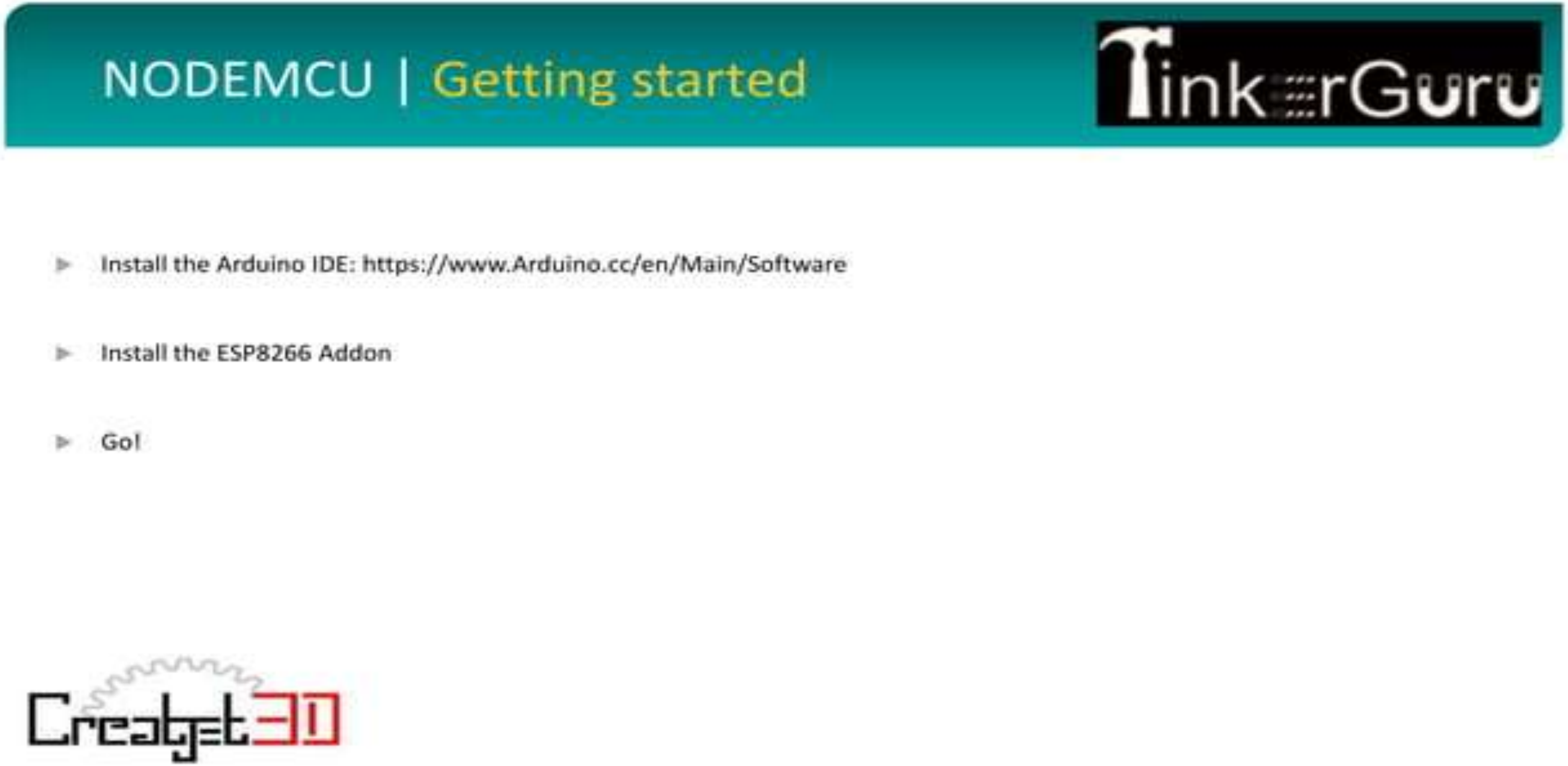


NODEMCU | Features

TinkerGuru

- I/O Pins:
  - Digital Pins: Pin D0 – Pin D10 Digital Pins
  - PWM Pins : 12 PWM Pins
  - Analog Pins : Pin A0
- Power PINS
  - Ground : 5 Pins
  - 3.3V : 3
  - Vin Pin : 1 Adding external supply of +5V (is not connected to USB)

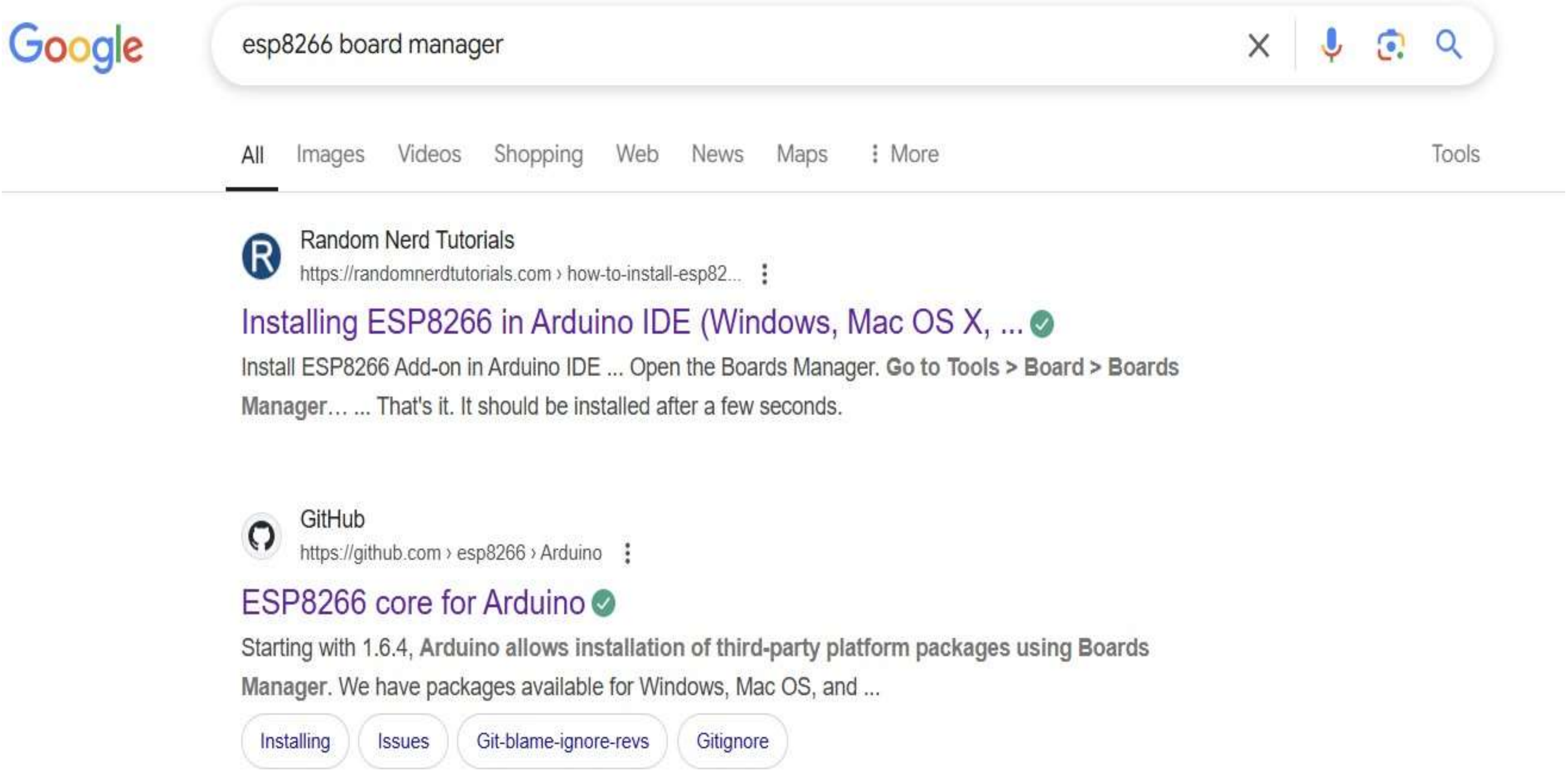
Pulse width modulation



NODEMCU | Getting started

- Install the Arduino IDE: https://www.Arduino.cc/en/Main/Software
- Install the ESP8266 Addon
- Go!

Asst. Prof. S.Asra
CSE Department, CMRIT



esp8266 board manager

Random Nerd Tutorials
https://randomnerdtutorials.com › how-to-install-esp82...

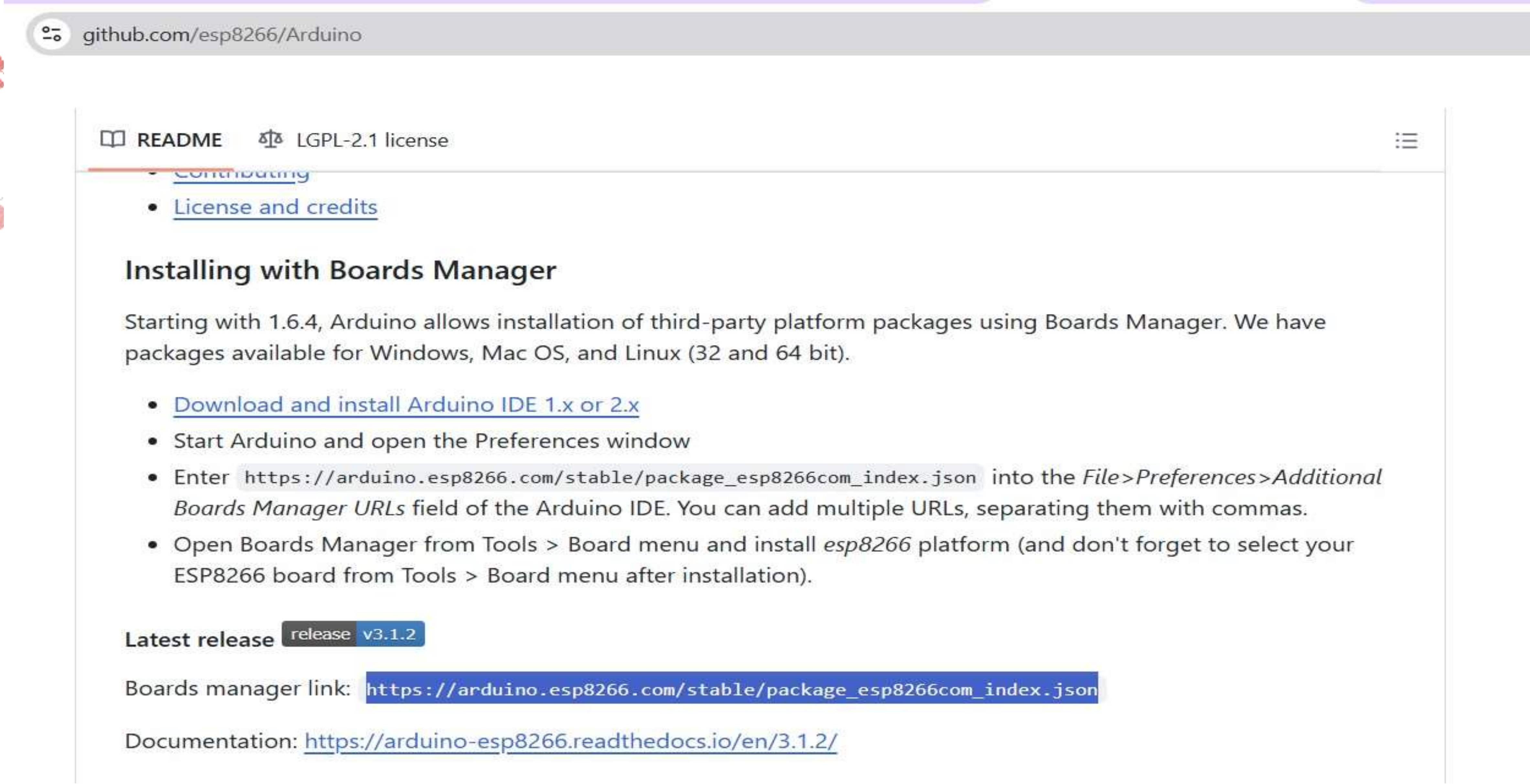
Installing ESP8266 in Arduino IDE (Windows, Mac OS X, ...

Install ESP8266 Add-on in Arduino IDE ... Open the Boards Manager. **Go to Tools > Board > Boards Manager**… ... That's it. It should be installed after a few seconds.

GitHub
https://github.com › esp8266 › Arduino

ESP8266 core for Arduino

Starting with 1.6.4, **Arduino allows installation of third-party platform packages using Boards Manager**. We have packages available for Windows, Mac OS, and ...



- Contributing
- License and credits

Installing with Boards Manager

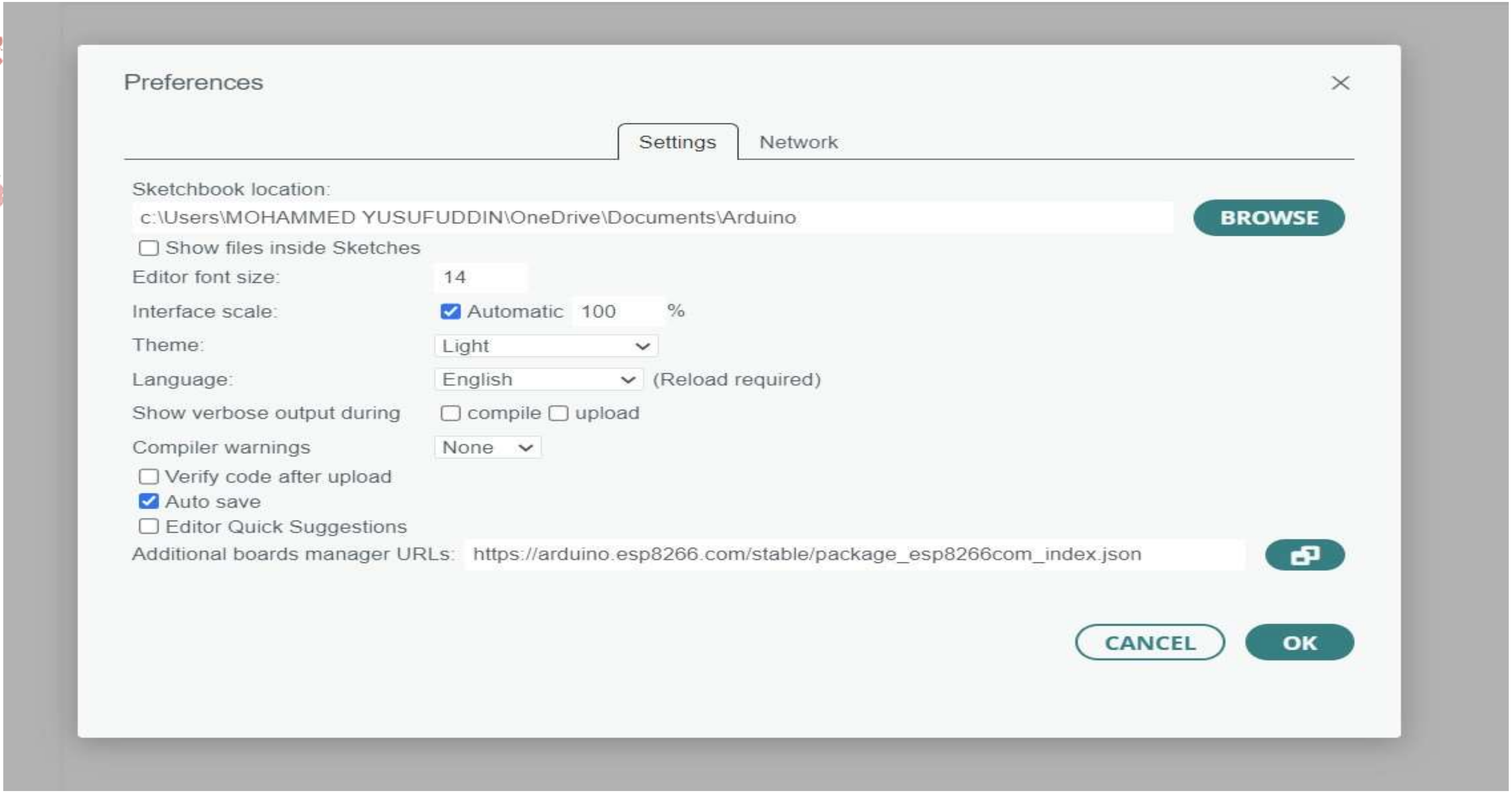
Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).

- Download and install Arduino IDE 1.x or 2.x
- Start Arduino and open the Preferences window
- Enter `https://arduino.esp8266.com/stable/package_esp8266com_index.json` into the *File>Preferences>Additional Boards Manager URLs* field of the Arduino IDE. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install *esp8266* platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

Latest release release v3.1.2

Boards manager link: `https://arduino.esp8266.com/stable/package_esp8266com_index.json`

Documentation: https://arduino-esp8266.readthedocs.io/en/3.1.2/



Sketchbook location:
c:\Users\MOHAMMED YUSUFUDDIN\OneDrive\Documents\Arduino

□ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light

Language: English (Reload required)

Show verbose output during □ compile □ upload

Compiler warnings None

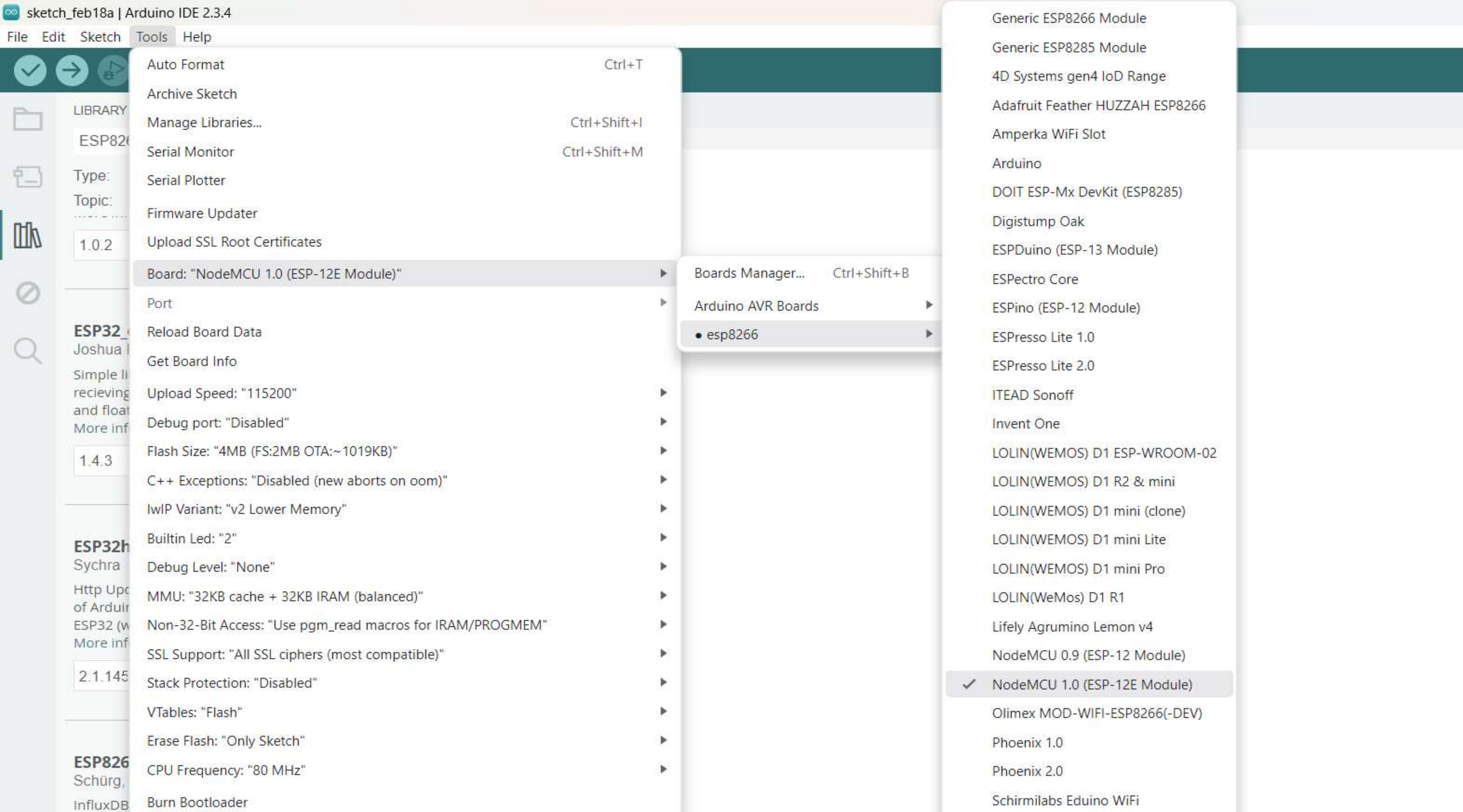
□ Verify code after upload

☑ Auto save

□ Editor Quick Suggestions

Additional boards manager URLs: https://arduino.esp8266.com/stable/package_esp8266com_index.json

Asst. Prof. S.Asra
CSE Department, CMRIT

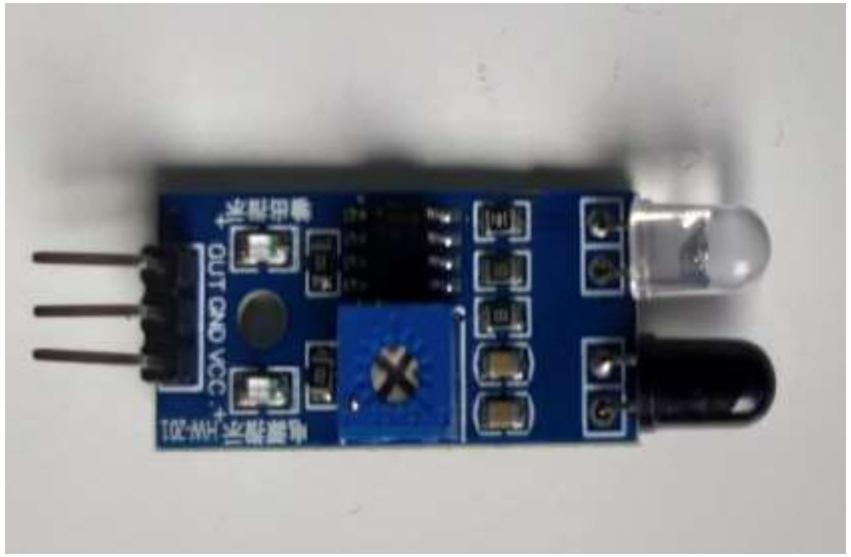


Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "NodeMCU 1.0 (ESP-12E Module)"
Port
Reload Board Data
Get Board Info
Upload Speed: "115200"
Debug port: "Disabled"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
C++ Exceptions: "Disabled (new aborts on oom)"
IwIP Variant: "v2 Lower Memory"
Builtin Led: "2"
Debug Level: "None"
MMU: "32KB cache + 32KB IRAM (balanced)"
Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM"
SSL Support: "All SSL ciphers (most compatible)"
Stack Protection: "Disabled"
VTables: "Flash"
Erase Flash: "Only Sketch"
CPU Frequency: "80 MHz"
Burn Bootloader

Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
● esp8266

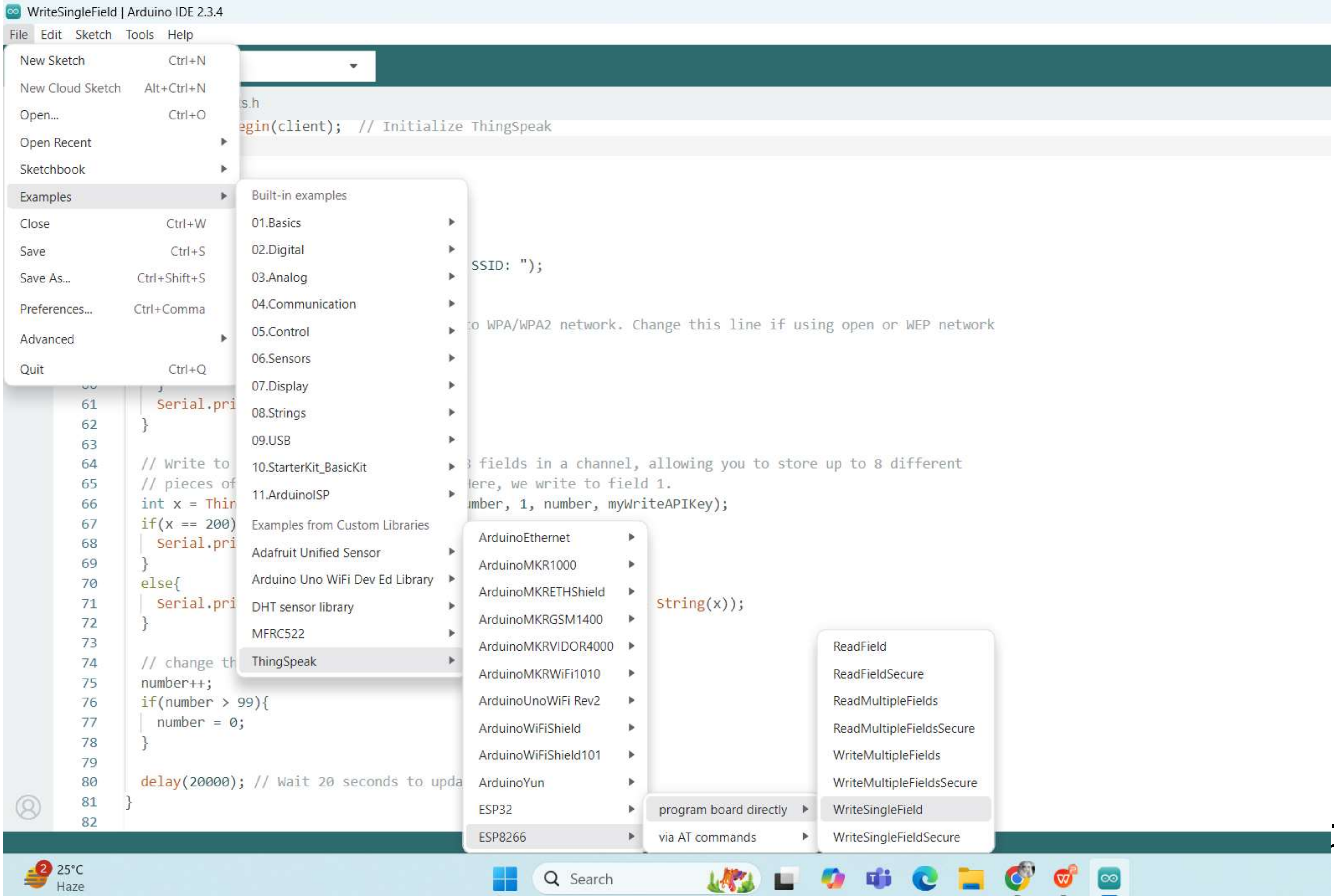
Generic ESP8266 Module
Generic ESP8285 Module
4D Systems gen4 IoD Range
Adafruit Feather HUZZAH ESP8266
Amperka WiFi Slot
Arduino
DOIT ESP-Mx DevKit (ESP8285)
Digistump Oak
ESPDuino (ESP-13 Module)
ESPectro Core
ESPino (ESP-12 Module)
ESPresso Lite 1.0
ESPresso Lite 2.0
ITEAD Sonoff
Invent One
LOLIN(WEMOS) D1 ESP-WROOM-02
LOLIN(WEMOS) D1 R2 & mini
LOLIN(WEMOS) D1 mini (clone)
LOLIN(WEMOS) D1 mini Lite
LOLIN(WEMOS) D1 mini Pro
LOLIN(WeMos) D1 R1
Lifely Agrumino Lemon v4
NodeMCU 0.9 (ESP-12 Module)
✓ NodeMCU 1.0 (ESP-12E Module)
Olimex MOD-WIFI-ESP8266(-DEV)
Phoenix 1.0
Phoenix 2.0
Schirmilabs Eduino WiFi

HARDWARE REQUIREMETS:



OUT- D3
GND- GND
VCC- 3.3V

Asst. Prof. S.Asra
CSE Department, CMRIT



```
s.h
egin(client);  // Initialize ThingSpeak

SSID: ");

o WPA/WPA2 network. Change this line if using open or WEP network

    Serial.pri
  }

  // Write to                      fields in a channel, allowing you to store up to 8 different
  // pieces of                     Here, we write to field 1.
  int x = Thin                     umber, 1, number, myWriteAPIKey);
  if(x == 200)
    Serial.pri
  }
  else{
    Serial.pri                                String(x));
  }

  // change th
  number++;
  if(number > 99){
    number = 0;
  }

  delay(20000); // Wait 20 seconds to upda
}
```

. Prof. S.Asra
ment, CMRIT

```
#include<ESP8266WiFi.h>
#include"secrets.h"
#include"ThingSpeak.h"// always include thingspeak header file after other header
files and custom macros
char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;
unsignedlong myChannelNumber = SECRET_CH_ID;
constchar * myWriteAPIKey = SECRET_WRITE_APIKEY;
int number = 0;
int IRPIN = D3;
voidsetup(){
  Serial.begin(115200);  // Initialize serial
  while(!Serial){
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialize ThingSpeak
}
```

```
voidloop(){
// Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using
open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  // 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.
  int x = ThingSpeak.writeField(myChannelNumber, 1, number, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
```

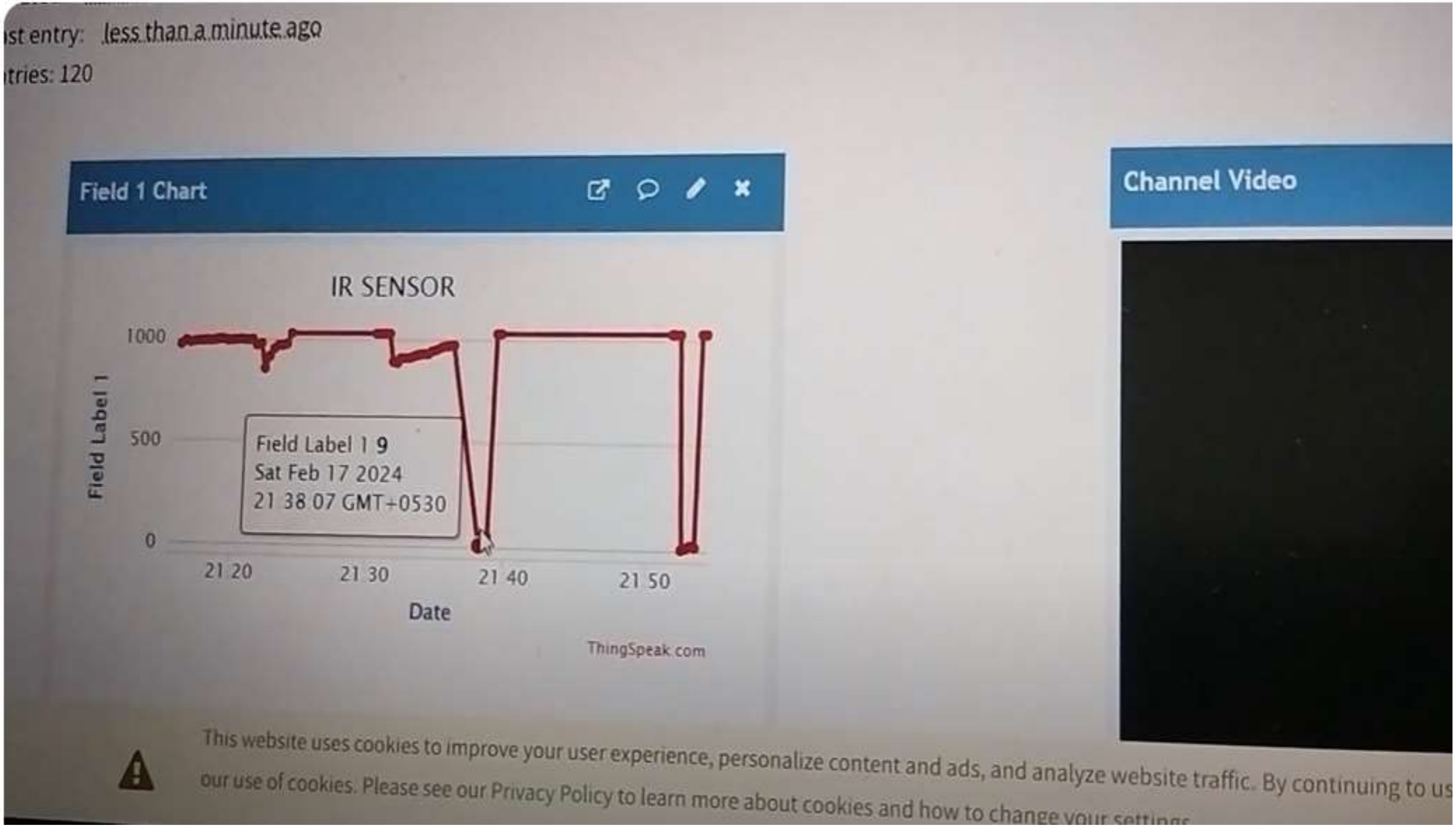
```
// change the value
  number++;
  if(number > 99){
    number = 0;
  }

  delay(20000); // Wait 20 seconds to update the
channel again
}
```

Secrets.h

```
#define SECRET_SSID "ACENAARTECHNOLOGY”
#define SECRET_PASS "987654321"
#define SECRET_CH_ID 1748773
#define SECRET_WRITE_APIKEY "IT8VWXOS2XKSYFR9"
```

OUTPUT

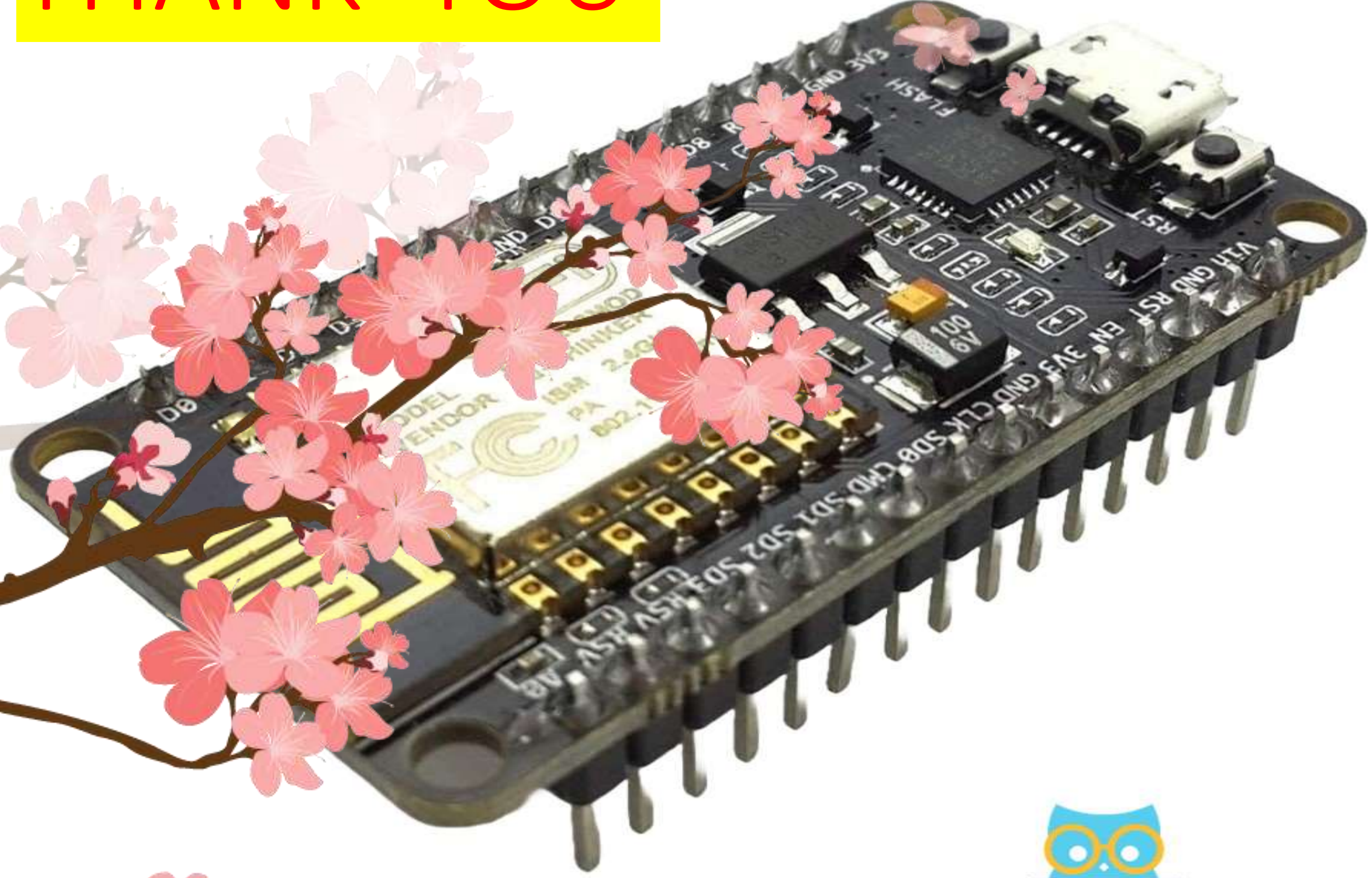


ıst entry: less than a minute ago

ıtries: 120

This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to us our use of cookies. Please see our Privacy Policy to learn more about cookies and how to change your settings

THANK YOU



newbiely.com