

ECE 442/510
Internet of Things and Cyber Physical Systems

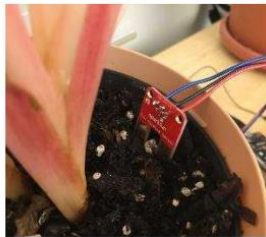
**Lecture 5: Temperature Sensing System using
Bluetooth (Lab 3)**
Summer 2022

ThingSpeak

- ThingSpeak (<http://www.thingspeak.com>)
 - The Free Open IoT Platform with MATLAB Analytics

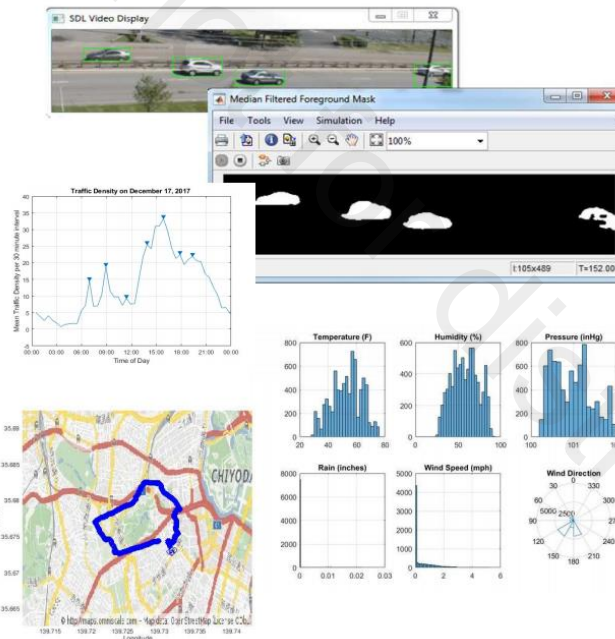
Collect

Send sensor data privately to the cloud



Analyze

Analyze and visualize your data with MATLAB



Act

Trigger a reaction (Tweet, alarm, motor, LED, etc.)

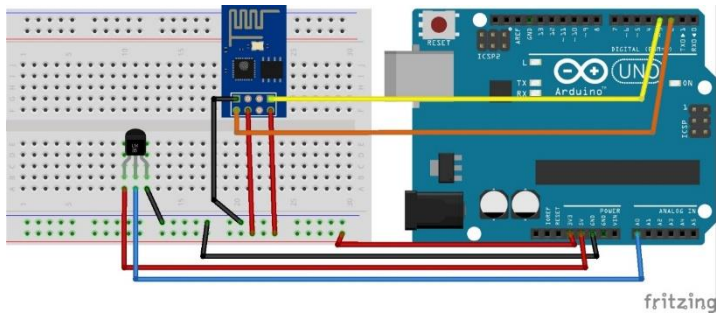


IoT using Arduino and ThingSpeak

- IoT Project Idea – Temperature Monitoring System

Collect

Arduino with Wi-Fi and temperature sensor



Temperature
Sensor

Wi-Fi Module

Analyze

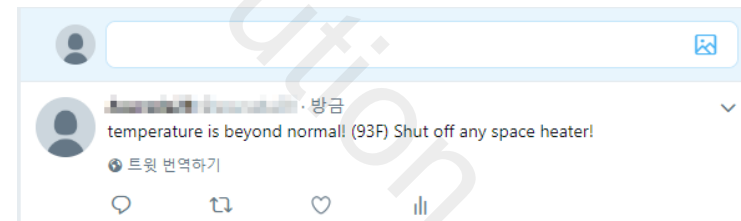
Watch out for temperature above threshold



Act

Send an alarm

Broadcast message through Social Media



ThingSpeak Concepts

- **Channel**
 - A place to store and retrieve data that is organized into fields
- **Field**
 - Location in a channel where a variable is stored
- **Write API Key**
 - API key that allows writing data to a channel
- **Read API Key**
 - API key that allows reading data from a channel
- **MATLAB Analytics**
 - Run MATLAB directly on thingspeak.com for analysis and visualization of your data
- **Apps**
 - Add-on functionality to ThingSpeak including API to other services

ThingSpeak Specs

ThingSpeak Specs

	FREE For small non-commercial projects and for evaluation of the service	ACADEMIC For academic use by faculty, staff, or researchers at degree-granting institutions ⁽¹⁾
Scalable for larger projects	× No. Annual usage is capped.	✓
Number of messages	3 million/year (~8,200/day) ⁽²⁾	33 million/year per unit (~90,000/day per unit) ⁽²⁾
Message update interval limit	Every 15 seconds	Every second
MATLAB Compute Timeout	20 seconds	60 seconds
Number of simultaneous MQTT subscriptions	Limited to 3	50 per unit
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support	Forum	Standard MathWorks support

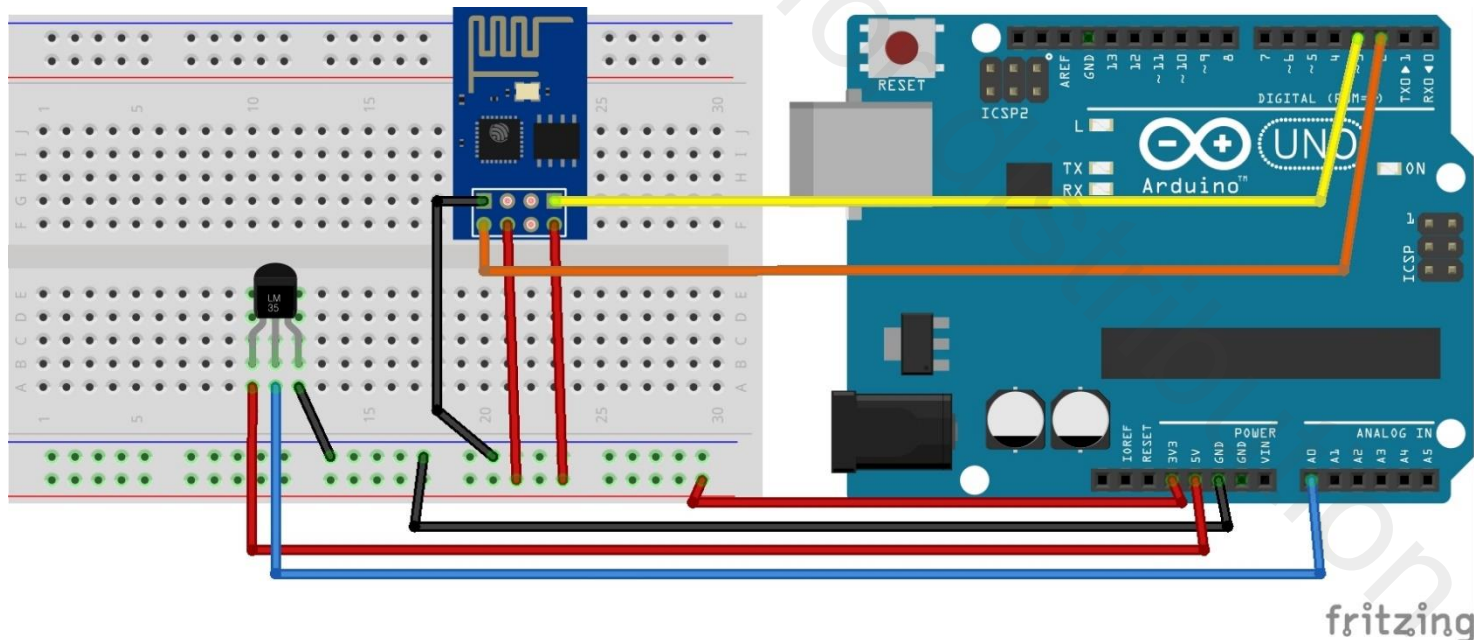
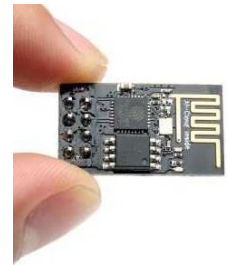
Toolboxes supported: <https://www.mathworks.com/help/thingspeak/matlab-toolbox-access.html>

References for ThingSpeak

- Deep Learning with MATLAB resources:
<https://www.mathworks.com/discovery/deep-learning.html>
- Free 2-hour Online Deep Learning Training:
<https://matlabacademy.mathworks.com/>
- Deep Learning Blog: <https://blogs.mathworks.com/deep-learning/>
- MathWorks Open IoT Platform – ThingSpeak:
<https://thingspeak.com/>
- IoT Blog: <https://blogs.mathworks.com/iot/>
- Maker Community:
<https://www.mathworks.com/matlabcentral/maker.html>

IoT using Arduino and ThingSpeak

- Enable Wi-Fi on Arduino UNO
 - ESP8266 Wi-Fi module serially connected to the board (Tx/Rx)
 - 802.11 b/g/n with 2.4 GHz support
- Connect a sensor to the Arduino UNO
 - Temperature sensor (TMP36) to Analog Input
 - Operating range -40C to +125C



IoT using Arduino and ThingSpeak

- Create Channel on ThingSpeak


ThingSpeak™

Channels ▾ Apps ▾ Community Support ▾

Commercial Use How to Buy Account ▾ Sign Out

IoT Arduino WiFi Temperature Readings

Channel ID: 512170 | IoT Arduino WiFi Temperature Readings

Author: 

Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Channel Settings

Percentage complete 50%

Channel ID 512170

Name

Description

Field 1 ☒

Field 2 ☐

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

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

- **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.
- **Latitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.
- **Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- **Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.

IoT using Arduino and ThingSpeak

- API Key for exchanging data with Arduino over Wi-Fi

The screenshot shows the ThingSpeak website interface for a specific channel. At the top is a navigation bar with links like 'Channels', 'Apps', 'Community', 'Support', 'Commercial Use', 'How to Buy', 'Account', and 'Sign Out'. Below this, the channel name 'IoT Arduino WiFi Temperature Readings' is displayed, along with its ID '512170', author, and access status 'Private'. A series of tabs allows switching between 'Private View', 'Public View', 'Channel Settings', 'Sharing', 'API Keys' (which is selected), and 'Data Import / Export'. The 'API Keys' section is divided into two main areas: 'Write API Key' and 'Read API Keys'. The 'Write API Key' area shows a key '33MD1UXSSE5ZYM6F' and a 'Generate New Write API Key' button. The 'Read API Keys' area shows a key 'CYUFSJJYR37LJ64P', a 'Note' field, 'Save Note' and 'Delete API Key' buttons, and a 'Generate New Read API Key' button. To the right of these settings is a 'Help' section explaining API keys and their settings, followed by 'API Requests' which lists several GET API endpoints for updating feeds, getting feeds, fields, and status updates, each with a corresponding URL and a 'Learn More' link at the bottom.

ThingSpeak™ Channels ▾ Apps ▾ Community Support ▾ Commercial Use How to Buy Account ▾ Sign Out

IoT Arduino WiFi Temperature Readings

Channel ID: 512170 | IoT Arduino WiFi Temperature Readings

Author: Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key 33MD1UXSSE5ZYM6F

Generate New Write API Key

Read API Keys

Key CYUFSJJYR37LJ64P

Note

Save Note Delete API Key

Generate New Read API Key

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

Update a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=33MD1UXSSE5ZYM6F&field=
```

Get a Channel Feed

```
GET https://api.thingspeak.com/channels/512170/feeds.json?api_key=C
```

Get a Channel Field

```
GET https://api.thingspeak.com/channels/512170/fields/1.json?api_key=
```

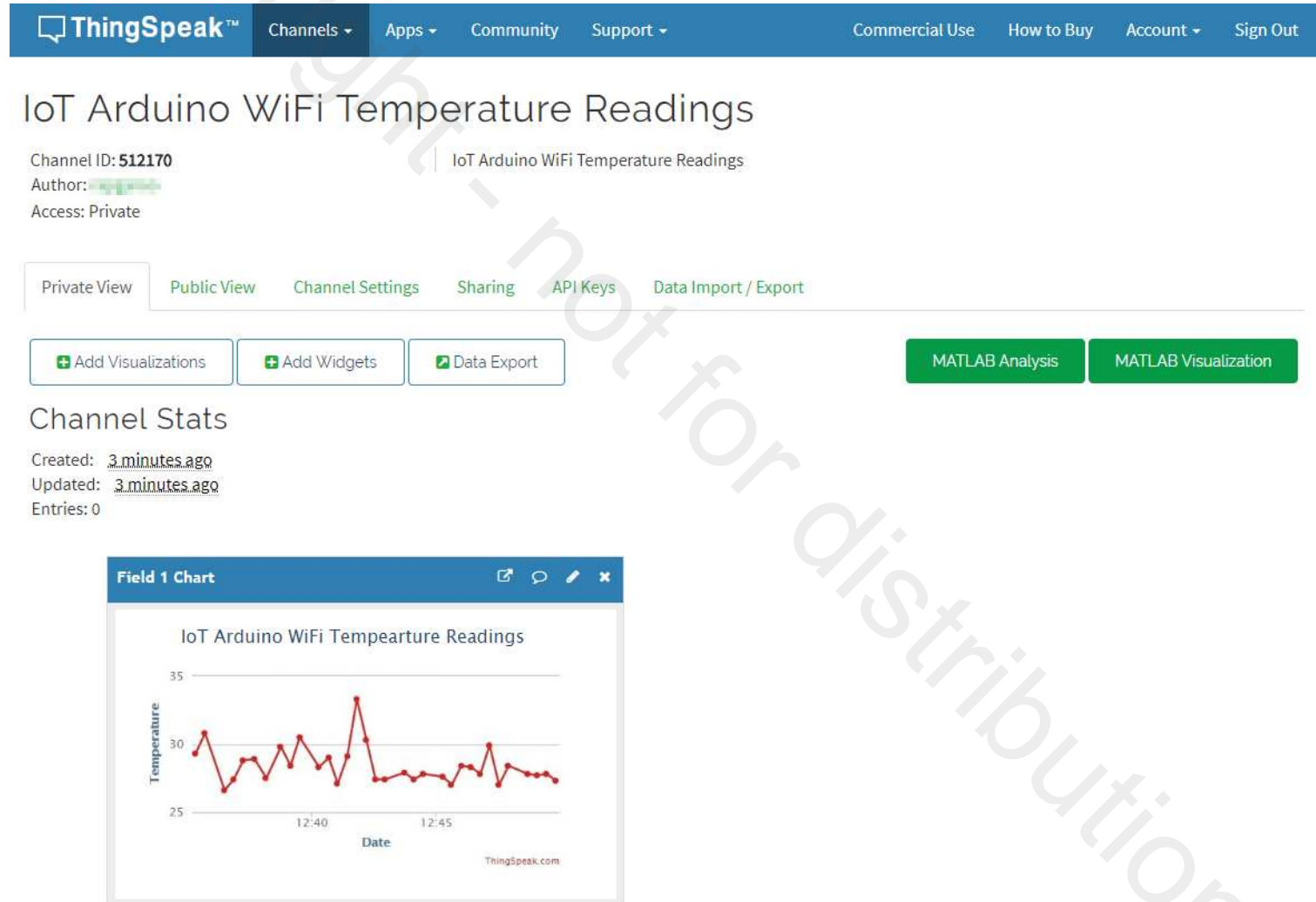
Get Channel Status Updates

```
GET https://api.thingspeak.com/channels/512170/status.json?api_key=C
```

[Learn More](#)

IoT using Arduino and ThingSpeak

- Collected data can be seen on ThingSpeak.com



Lab 3

Temperature Sensing System using Bluetooth

Lab explanation

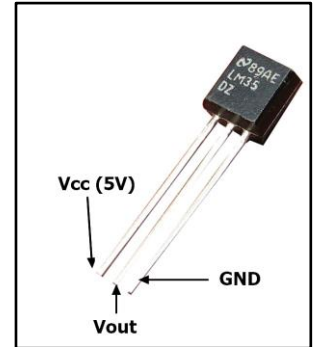
<https://youtu.be/ATuXLkguspo>

Temperature Sensing System using Bluetooth

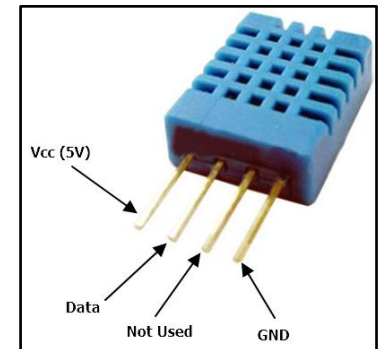
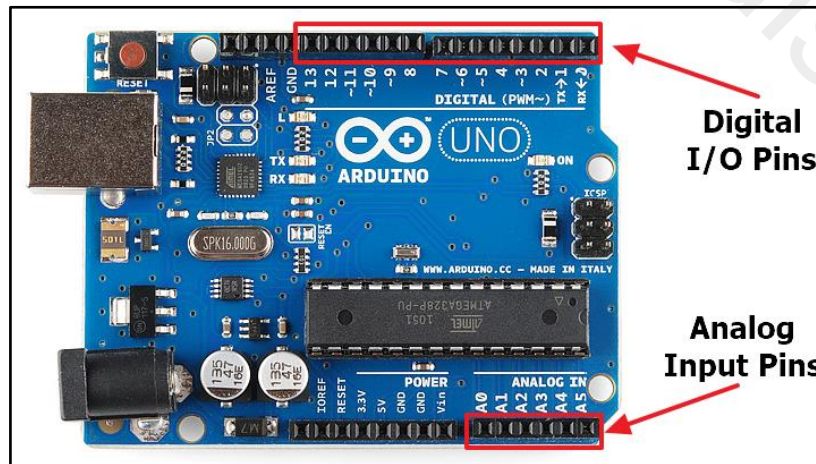
- Utilize your obtained skills on Arduino (Lab 1) and Raspberry Pi (Lab2)
- Connect an ambient temperature sensor to Arduino using Analog Input Pin
- Connect Arduino and Raspberry Pi wirelessly using Bluetooth
 - C programming on Arduino
 - Python programming on Raspberry Pi
- Connect Raspberry Pi to ThingSpeak
 - Display collected temperature sensor data from Arduino to the web

Hardware Set Up on Arduino

- Connect LM35 temperature sensor to Arduino
 - V_{cc} to 5V on Arduino
 - GND to GND on Arduino
 - V_{out} to one of Analog Input Pins #0 - #5
- Connect DHT11 temperature/humidity sensor to Arduino
 - V_{cc} to 5V on Arduino
 - GND to GND on Arduino
 - Data to one of Digital I/O Pins

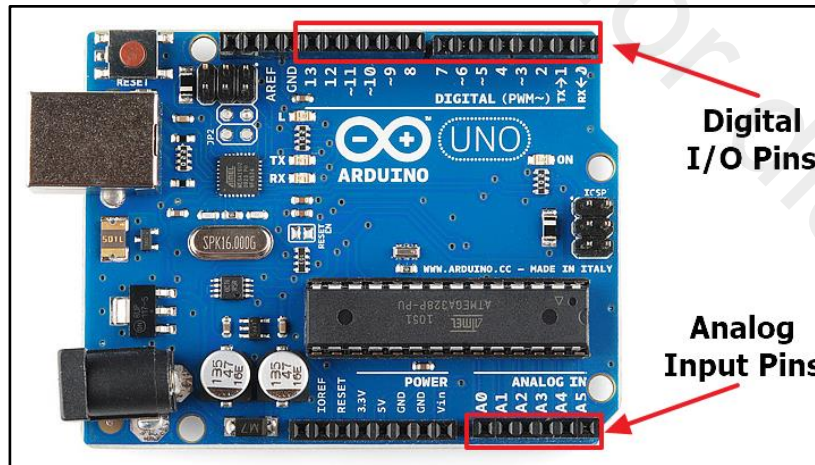


**** DO NOT CONNECT TO PIN #0 and #1**



Hardware Set Up on Arduino

- Connect HC-06 Bluetooth module to Arduino
 - V_{cc} to 5V on Arduino
 - GND to GND on Arduino
 - RxD to one of Digital I/O Pins
 - TxD to one of Digital I/O Pins
- ** DO NOT CONNECT TO PIN #0 and #1**



Bluetooth Set Up using Arduino

- Use example code from Arduino IDE called “SoftwareSerialExample”

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(TxD of BT module, RxD of BT module); // RxD of Arduino, TxD of Arduino

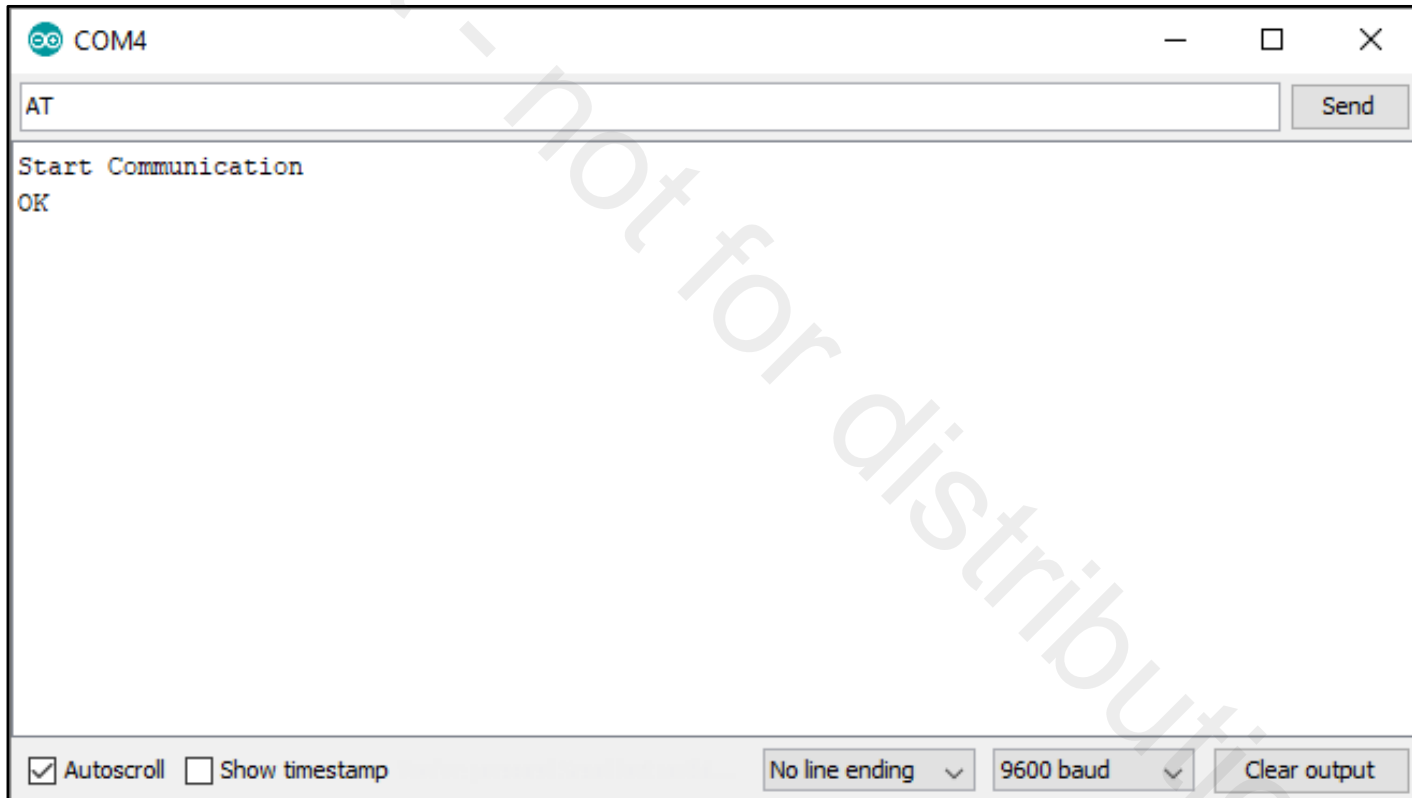
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  mySerial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  Serial.println("Start Communication");
}

void loop() { // run over and over
  if (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }
}
```

- Modify this code to appropriate PINs for your TxD and RxD to HC-06

Bluetooth Set Up using Arduino

- Type “AT” and click on “SEND”. If your connection is set up correctly, you should have the same output like below:
 - make sure your Serial Monitor’s baud rate matches with you code



Bluetooth Set Up using Arduino

- Use AT commands to configure your HC-06 module
- Follow directions specified in Lab 3 Manual Procedure.
 - Change Bluetooth Module Name
 - Change Bluetooth Pairing PIN number
 - Change Bluetooth Communication Baud Rate

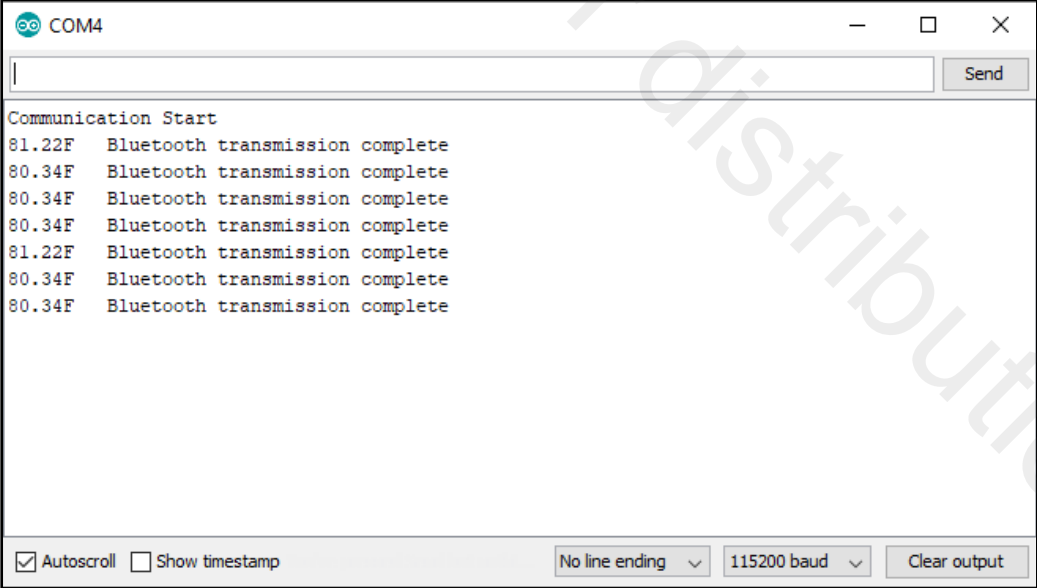
Command	Reply	Comment
AT	OK	Testing Communication Establishment
AT+VERSION	OKlinvorV1.8	Firmware version
AT+NAMElab3BTmodule	OKsetname	Sets the name of the Bluetooth module to "lab3BTmodule"
AT+PIN1234	OKsetPIN	Sets the PIN to 1234
AT+BAUD1	OK1200	Sets the baud rate to 1200
AT+BAUD2	OK2400	Sets the baud rate to 2400
AT+BAUD3	OK4800	Sets the baud rate to 4800
AT+BAUD4	OK9600	Sets the baud rate to 9600 (<i>default</i>)
AT+BAUD5	OK19200	Sets the baud rate to 19200
AT+BAUD6	OK38400	Sets the baud rate to 38400
AT+BAUD7	OK57600	Sets the baud rate to 57600
AT+BAUD8	OK115200	Sets the baud rate to 115200
AT+BAUD9	OK230400	Sets the baud rate to 230400
AT+BAUDA	OK460800	Sets the baud rate to 460800
AT+BAUDB	OK921600	Sets the baud rate to 921600
AT+BAUDC	OK1382400	Sets the baud rate to 1382400

Temperature Sensing using Arduino

- Write your own C program on Arduino IDE to retrieve value from LM35 sensor
- Convert retrieved value to Fahrenheit by using given equations

$$\text{temperature}_{\text{celsius}} = \frac{\left(\frac{\text{raw_sensor_value}}{1024}\right) \times 5000}{10} \quad \text{temperature}_{\text{fahrenheit}} = \frac{(\text{temperature}_{\text{celsius}} \times 9)}{5} + 32$$

- Use *write(val)* function to send data to HC-06
- Expected output is like below:



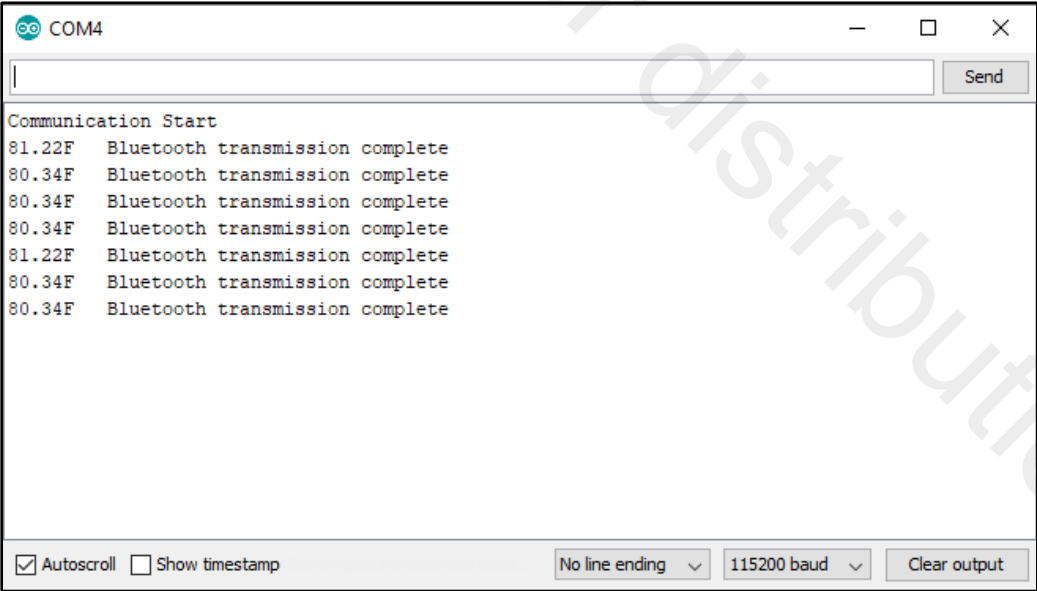
```
COM4
| Send
Communication Start
81.22F Bluetooth transmission complete
80.34F Bluetooth transmission complete
80.34F Bluetooth transmission complete
80.34F Bluetooth transmission complete
81.22F Bluetooth transmission complete
80.34F Bluetooth transmission complete
80.34F Bluetooth transmission complete
Autoscroll Show timestamp No line ending 115200 baud Clear output
```

Temperature Sensing using Arduino

- Write your own C program by utilizing DHT library on Arduino IDE to retrieve value from DHT11 sensor
- Convert retrieved value to Fahrenheit by using given equations

$$\text{temperature}_{\text{fahrenheit}} = \frac{(\text{temperature}_{\text{celsius}} \times 9)}{5} + 32$$

- Use *write(val)* function to send data to HC-06
- Expected output is like below:



```
COM4
|
| Send
Communication Start
81.22F Bluetooth transmission complete
80.34F Bluetooth transmission complete
80.34F Bluetooth transmission complete
80.34F Bluetooth transmission complete
81.22F Bluetooth transmission complete
80.34F Bluetooth transmission complete
80.34F Bluetooth transmission complete
Autoscroll Show timestamp No line ending 115200 baud Clear output
```

Bluetooth and Python using RPi

- Need to identify HC-06's MAC address
 - You can find this using Raspberry Pi's Bluetooth device by scanning

```
pi@raspberrypi:~ $ bluetoothctl
[NEW] Controller B8:27:EB:24:53:79 raspberrypi [default]
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:24:53:79 Discovering: yes
[NEW] Device 12:E7:C6:D2:71:89 OfficeJet 5200 series
[NEW] Device 6D:CF:3D:11:0E:53 6D-CF-3D-11-0E-53
[NEW] Device 6B:4A:92:8B:B2:6C 6B-4A-92-8B-B2-6C
[NEW] Device 2C:41:A1:2C:8C:5C LE-Bose Micro SoundLink
[NEW] Device 20:15:07:27:50:83 20-15-07-27-50-83
[CHG] Device 20:15:07:27:50:83 LegacyPairing: no
[CHG] Device 20:15:07:27:50:83 Name: ECE442LAB3_YI
[CHG] Device 20:15:07:27:50:83 Alias: ECE442LAB3_YI
[CHG] Device 20:15:07:27:50:83 LegacyPairing: yes
[CHG] Device 6B:4A:92:8B:B2:6C RSSI: -91
[bluetooth]# pair 20:15:07:27:50:83
Attempting to pair with 20:15:07:27:50:83
[CHG] Device 20:15:07:27:50:83 Connected: yes
Request PIN code
[agent] Enter PIN code: 1234
[CHG] Device 20:15:07:27:50:83 UUIDs: 00001101-0000-1000-8000-00805f9b34fb
[CHG] Device 20:15:07:27:50:83 ServicesResolved: yes
[CHG] Device 20:15:07:27:50:83 Paired: yes
Pairing successful
[CHG] Device 20:15:07:27:50:83 ServicesResolved: no
[CHG] Device 20:15:07:27:50:83 Connected: no
[bluetooth]# exit
Agent unregistered
[DEL] Controller B8:27:EB:24:53:79 raspberrypi [default]
pi@raspberrypi:~ $
```

Bluetooth and Python using RPi

- Raspberry Pi out-of-the-box does not have full Bluetooth development support
 - You need to install packages separately
- **Bluetooth for Python**
 - `sudo apt update`
 - `sudo apt install bluetooth libbluetooth-dev`
 - `sudo python3 -m pip install pybluez`
- **ThingSpeak for Python**
 - `sudo python3 -m pip install thingspeak`

Arduino, RPi and ThingSpeak

- Create your own channel at ThingSpeak.com
- Obtain your channel's ID
- Obtain your channel's Write API Key under "API Keys" tab
- Modify given "lab3.py" code and run your Python code on RPi
 - HC-06 MAC address
 - ThingSpeak Channel ID, Write API Key

