Experiment No. 03:

TEMPERATURE SENSING SYSTEM USING BLUETOOTH

By: David Dalmeida

Lab Partner: Nicholas Velcich

Instructor: Dr, Won-Jae / Dr. Jafar Saniie

ECE 442

Lab Date: 05-28-2019

Due Date: 06-04-2019

Acknowledgment: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature: ______________________

I. Introduction

A. Purpose

The purpose of this experiment was to familiariaze us with the temperature sensing system using LM35 temperature sensor wired to the Arduino uno board, transmitting temperature sensor data to the Raspberry Pi via Bluetooth (HC-06 module) and at the same time uploading the received data to our ThingSpeak to visualize the sensor data online. This experiment was a way for us to see how the Arduino can communicate with the raspberry pi via Bluetooth and how the uploading process to the cloud works.

B. Background

The Raspberry Pi microcomputer is a single board computer capable of completing many tasks.

Specifications:

- CPU: Broadcom BCM2837 1.2 GHz 64-bit Quad-core (ARMv8 Cortex-A53)
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- Network: 10/100 Ethernet, 2.4 GHz 802.11n Wi-Fi, Bluetooth 4.1
- Storage: microSD card
- GPIO: 40-pin header, populated
- Ports: HDMI, 3.5mm audio jack, 4xUSB 2.0 ports, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Arduino is an open source computer hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. The board itself is a computer on a single integrated chip that has a CPU, Control Unit, Memory, Clock, I/O.

The analog sensors from the arduino generate a continuous output signal in different voltages. In order to process this captured analog data, it was required to translate the analog signal into digital signal through an Analog-to-Digital Conversion (ADC). In this lab experiment an analog ambient temperature sensor, LM35 will be connected to the Arduino board which will transmit data over Bluetooth to the Raspberry Pi that will display it on ThinkSpeak.

Thinkspeak is an open source “Internet of things” application and API to store and retrieve data from things using HTTP over the internet via a local Area Network. We can create with thinkspeak sensor logging applications, local tracking applications, and a social network of things with status updates.

II. Lab Procedure and Equipment List

A. Equipment

- Raspberry Pi Single Board Computer
- Arduino UNO board
- USB cable
- LM35 Ambient Temperature sensor
- Breadboard
- HC-06 Bluetooth Transceiver

B. Procedure

1. Connect the LM 35 temperature sensor and one Bluetooth HC-06 module to the Arduino Uno board using a breadboard.

2. Make sure everything is set up correctly using the AT commands and switch to baud 115200

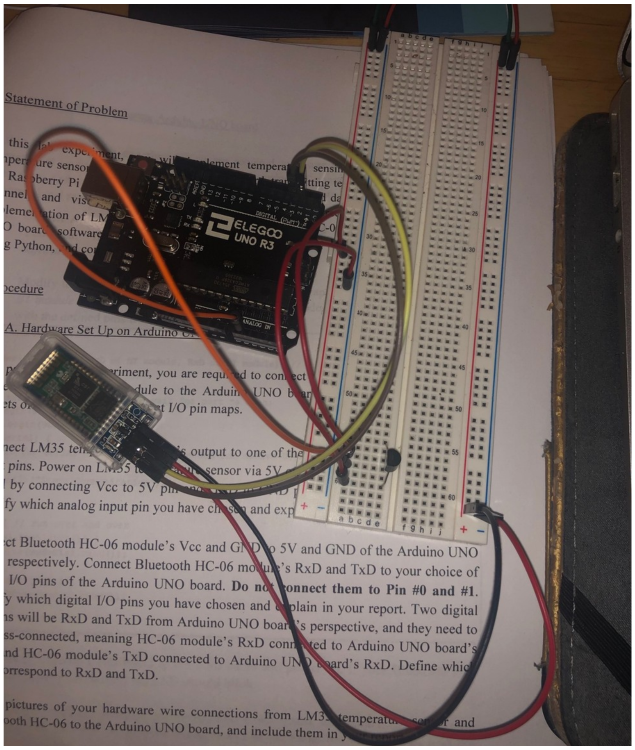
3. Retrieve the data and display on the serial monitor,

4. Set up the Bluetooth between the raspberry pi and the Bluetooth HC-06 MODULE.

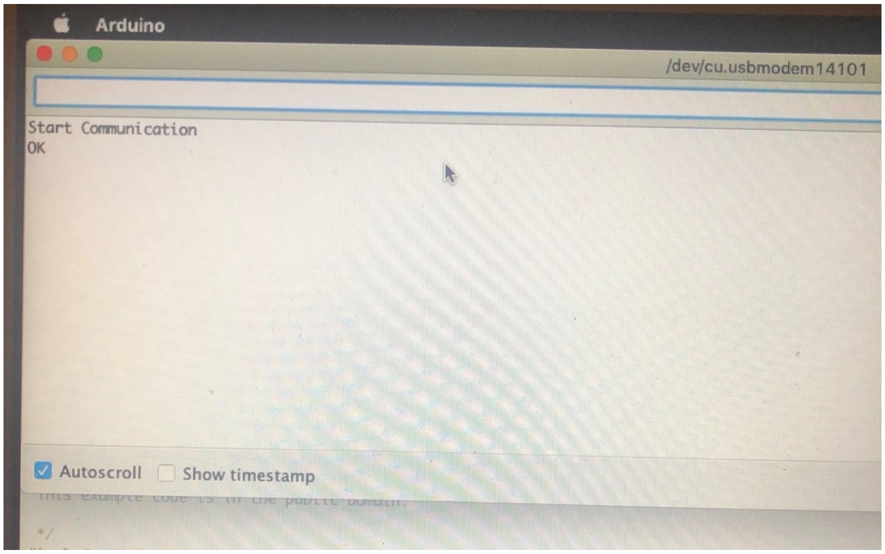
5. Perform the complete IoT temperature sensing system where the temperature data is collected on the Arduino UNO board transmitting to the Raspberry via Bluetooth.

III. Results

SETUP



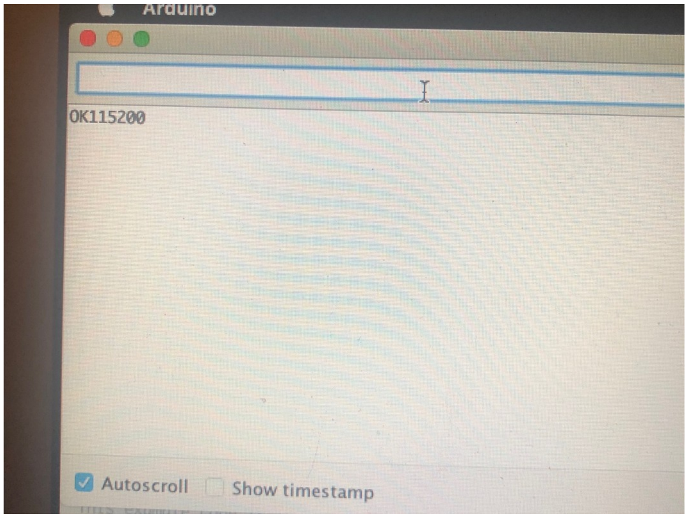
ON THE ARDUINO



```
Start Communication
OK
```

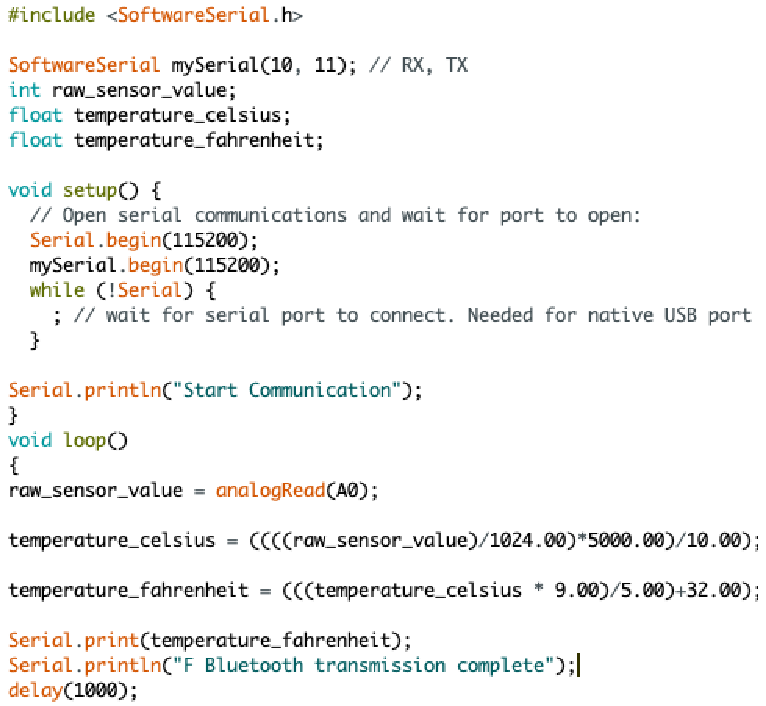
☑ Autoscroll □ Show timestamp

```
This example code is in the public domain.

*/
```



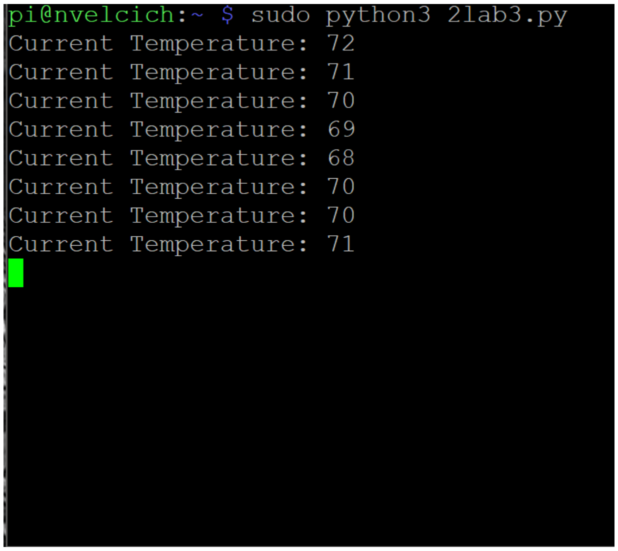
```
OK115200
```

CODE

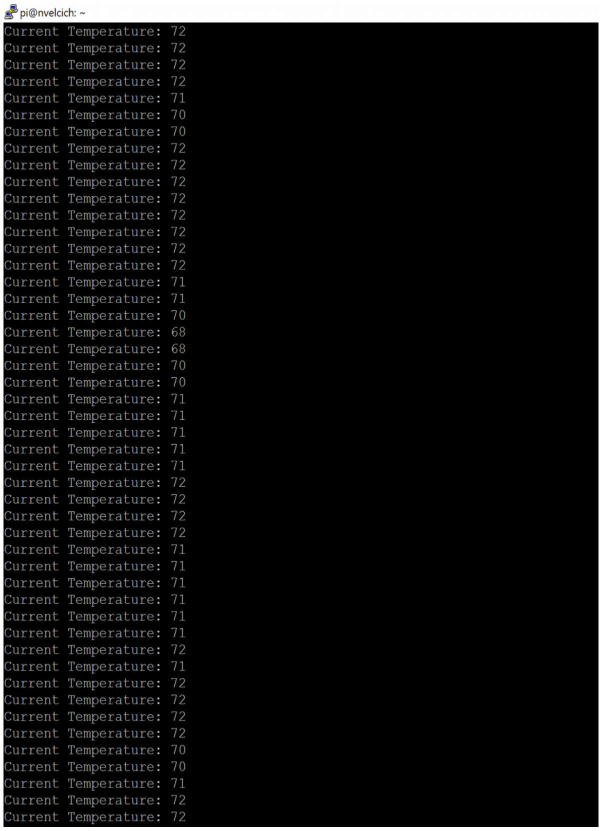


```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX
int raw_sensor_value;
float temperature_celsius;
float temperature_fahrenheit;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  mySerial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port
  }

Serial.println("Start Communication");
}
void loop()
{
raw_sensor_value = analogRead(A0);

temperature_celsius = ((((raw_sensor_value)/1024.00)*5000.00)/10.00);

temperature_fahrenheit = (((temperature_celsius * 9.00)/5.00)+32.00);

Serial.print(temperature_fahrenheit);
Serial.println("F Bluetooth transmission complete");
delay(1000);
```

RASPBERRY PI

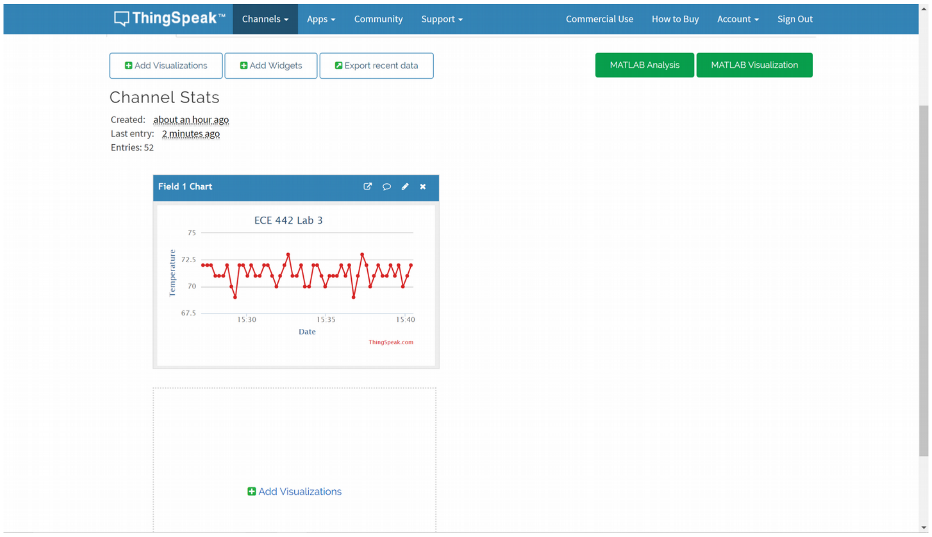


```
pi@nvelcich:~ $ sudo python3 2lab3.py
Current Temperature: 72
Current Temperature: 71
Current Temperature: 70
Current Temperature: 69
Current Temperature: 68
Current Temperature: 70
Current Temperature: 70
Current Temperature: 71
```



```
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 71
Current Temperature: 70
Current Temperature: 70
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 71
Current Temperature: 71
Current Temperature: 70
Current Temperature: 68
Current Temperature: 68
Current Temperature: 70
Current Temperature: 70
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 71
Current Temperature: 72
Current Temperature: 71
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 72
Current Temperature: 70
Current Temperature: 70
Current Temperature: 71
Current Temperature: 72
Current Temperature: 72
```

THINKSPEAK



Channel Stats

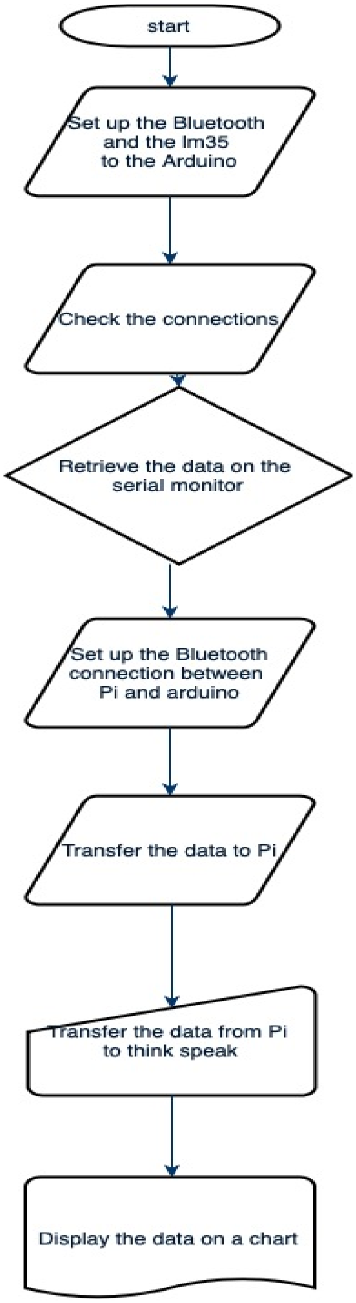
Created: about an hour ago

Last entry: 2 minutes ago

Entries: 52

IV. DISCUSSON

1. See Appendix at the end
2.


start
Set up the Bluetooth and the lm35 to the Arduino
Check the connections
Retrieve the data on the serial monitor
Set up the Bluetooth connection between Pi and arduino
Transfer the data to Pi
Transfer the data from Pi to think speak
Display the data on a chart

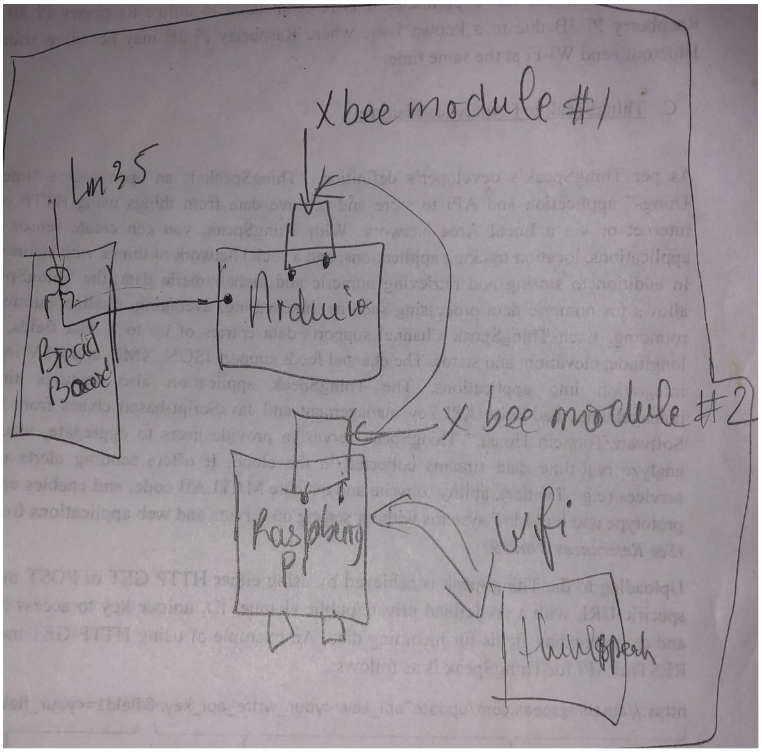
3. The datasheet of the Atmega states that its ADC has a definition of 1024 values (i.e. 10bits). So we need a formula to convert that retrieved integer from Analog to Digital.
   So we calculate the temperature in Celsius first by using a formula
   temperature_celsius = ((((raw_sensor_value)/1024.00)*5000.00)/10.00);

Then we convert it to Fahrenheit temperature_fahrenheit = (((temperature_celsius * 9.00)/5.00)+32.00);

And we output the result on the serial monitor.

4. So we first pair the raspberry pi to the Bluetooth of the Arduino.
   We connect the pi to the internet then we use the command bluetoothctl to get the commands to pair the 2 devices. We scan first then find the Bluetooth module and then pair them using The mac address and a pin number choosen.
   Then we set up Python on the raspberry pi using
   Sudo apt update
   Sudo apt install Bluetooth libbluetooth-dev
   …
   Then we connect the two devices to thinkSpeak
   We create an account and channel to get the API key.
   We use the code given to us that we modify the Mac address, the channel ID and API key.
   We then execute the code and get the results on thinkSpeak.

5. And 6. ZigBee is a communication device used for the data transfer between the controllers, computers, systems, really anything with a serial port. As it works with low power consumption, the transmission distances is limited to 10–100 meters line-of-sight.
   You connect the zigbee to the Arduino and another to the raspberry pi, set up the proper code for them to communicate.



Schematic

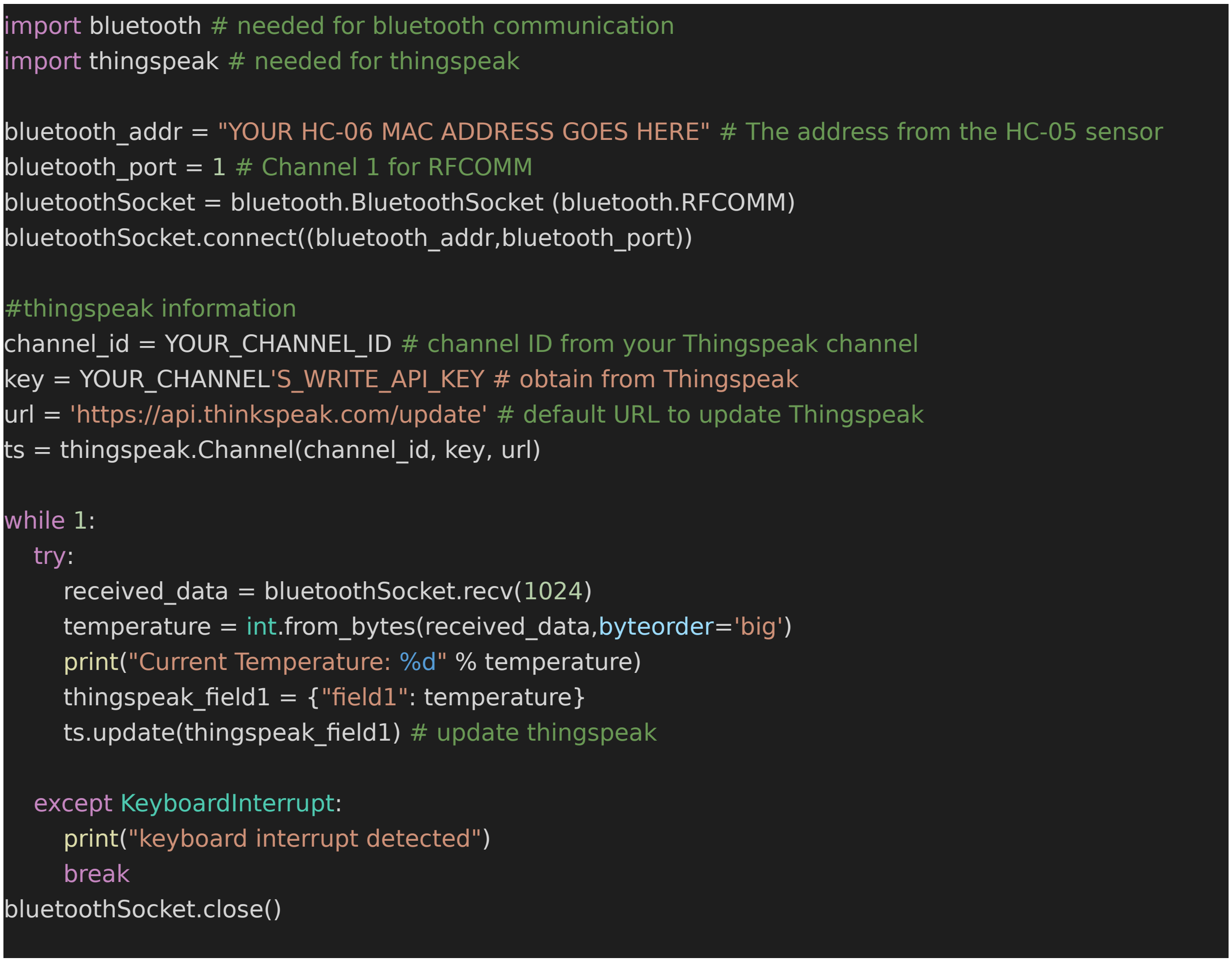
Conclusion

This lab was fantastic and showed us you can get reading from the real world and displaying the result on the internet. It is all a synchronized dance, with the temperature sensing system using LM35 temperature sensor wired to the Arduino uno board, transmitting temperature sensor data to the Raspberry Pi via Bluetooth (HC-06 module) and at the same time uploading the received data to our ThingSpeak to visualize the sensor data online.

Code on Arduino

```
#include <SoftwareSerial.h>

    SoftwareSerial mySerial(10, 11); // RX, TX
    int raw_sensor_value;
    float temperature_celsius;
    float temperature_fahrenheit;

    void setup() {
      // Open serial communications and wait for port to open:
      Serial.begin(115200);
      mySerial.begin(115200);
      while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
      }

    Serial.println("Start Communication");
    }
    void loop()
    {
    raw_sensor_value = analogRead(A0);

    temperature_celsius = ((((raw_sensor_value)/1024.00)*5000.00)/10.00);
    temperature_fahrenheit = (((temperature_celsius * 9.00)/5.00)+32.00);


    Serial.print(temperature_fahrenheit);
    Serial.println("F Bluetooth transmission complete");
    delay(1000);


    }
```

Code Raspberry Pi



```
import bluetooth # needed for bluetooth communication
import thingspeak # needed for thingspeak

bluetooth_addr = "YOUR HC-06 MAC ADDRESS GOES HERE" # The address from the HC-05 sensor
bluetooth_port = 1 # Channel 1 for RFCOMM
bluetoothSocket = bluetooth.BluetoothSocket (bluetooth.RFCOMM)
bluetoothSocket.connect((bluetooth_addr,bluetooth_port))

#thingspeak information
channel_id = YOUR_CHANNEL_ID # channel ID from your Thingspeak channel
key = YOUR_CHANNEL'S_WRITE_API_KEY # obtain from Thingspeak
url = 'https://api.thinkspeak.com/update' # default URL to update Thingspeak
ts = thingspeak.Channel(channel_id, key, url)

while 1:
    try:
        received_data = bluetoothSocket.recv(1024)
        temperature = int.from_bytes(received_data,byteorder='big')
        print("Current Temperature: %d" % temperature)
        thingspeak_field1 = {"field1": temperature}
        ts.update(thingspeak_field1) # update thingspeak

    except KeyboardInterrupt:
        print("keyboard interrupt detected")
        break
bluetoothSocket.close()
```

References

LAB MANUAL

THINSPEAK

LAB PARTNER

LM 35 DATASHEET

HC 06 DATASHEET