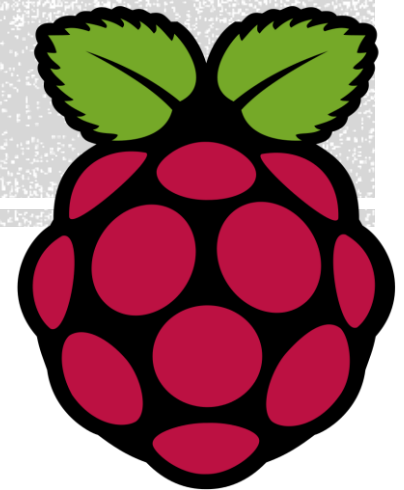# RASPBERRY PI WORKSHOP – DAY 2

ThinkLAB

www.thinklab.ph

secretariat@thinklab.ph

facebook.com/thinklab.ph

SparkLab Innovation Center

WHAT'S FOR TODAY?

- Function and Limitations of the GPIO

- Raspberry Pi to Arduino Communication with Temperature Sensor (LM35)

- IoT (Internet of Things)

- Cayenne (myDevices)

GPIO: Functions and Limitations

| Functions	Raspberry Pi	Arduino Uno
UART Universal Asynchronous Receiver Transmitter Also known as Serial	Yes	Yes
SPI Serial Peripheral Interface	Yes	Yes
I^{2}C Inter-Integrated Circuit	Yes	Yes
1-Wire	Yes	Yes
PWM Pulse Width Modulation	Yes	Yes
ADC Analog-to-Digital Converter	No	Yes
DAC Digital-to-Analog Converter	No	No

RASPBERRY PI – ARDUINO - LM35 THE LM35
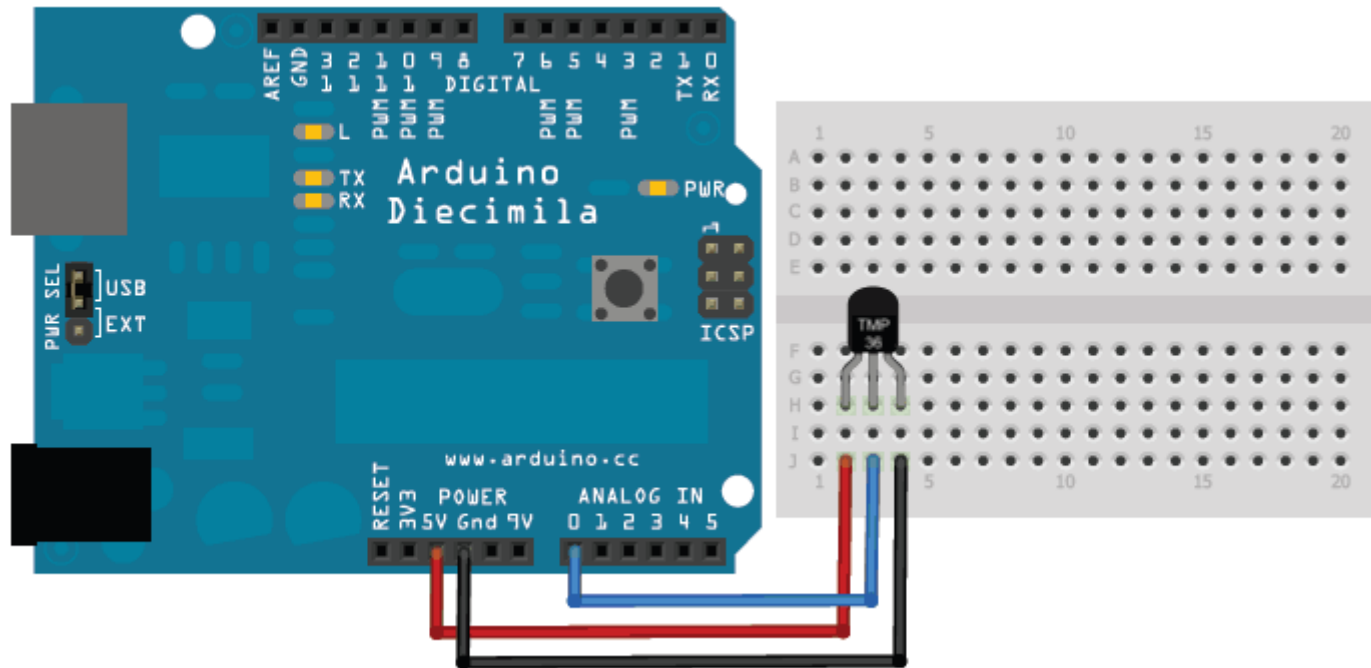
- Temperature Sensor

Analog Voltage Output
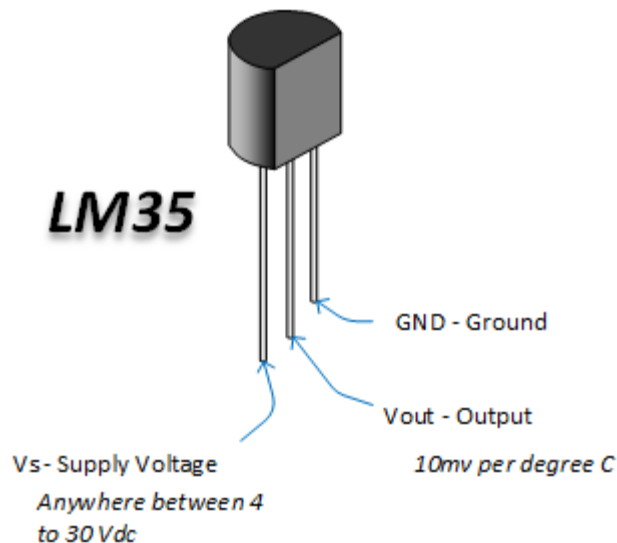
Linear + 10-mV/°C Scale Factor
0.5°C Ensured Accuracy (at 25°C)
Rated for Full −55°C to 150°C Range

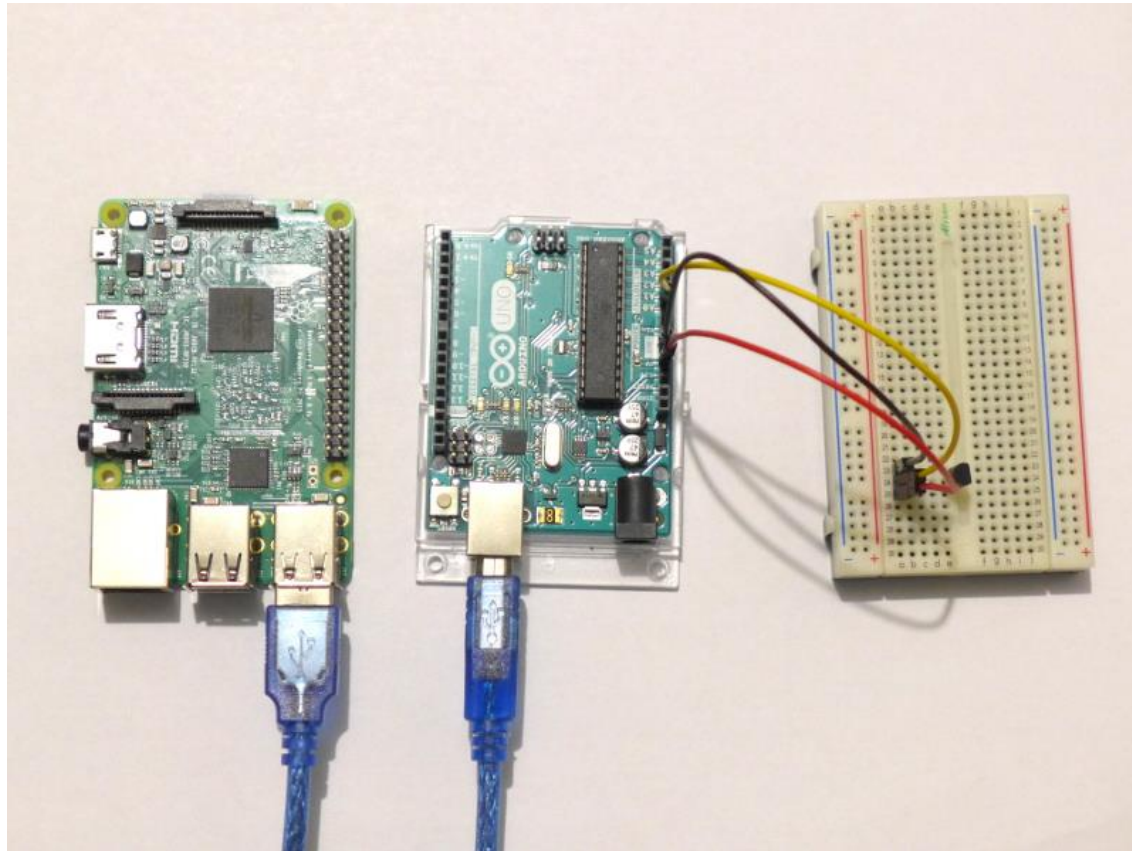Needs an Analog-to-Digital Converter (ADC).

LM35

GND - Ground

Vout - Output

Vs - Supply Voltage
Anywhere between 4 to 30 Vdc

10mv per degree C

RASPBERRY PI – ARDUINO - LM35
CONNECTING ARDUINO TO LM35

RASPBERRY PI – ARDUINO - LM35
RASPBERRY PI TO ARUINO

2 TYPES OF WIRED COMMUNICATION
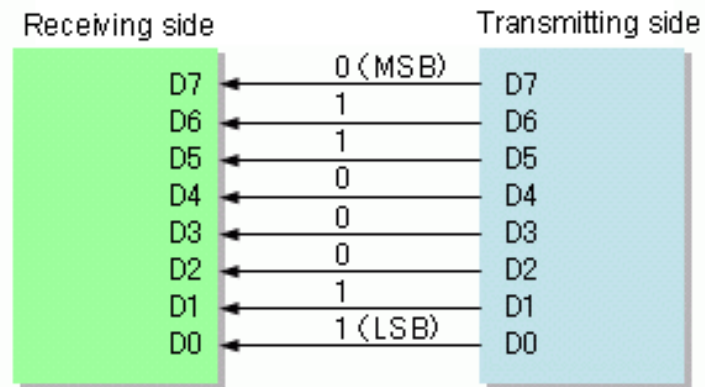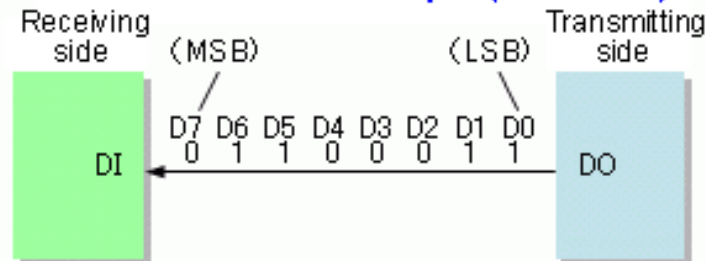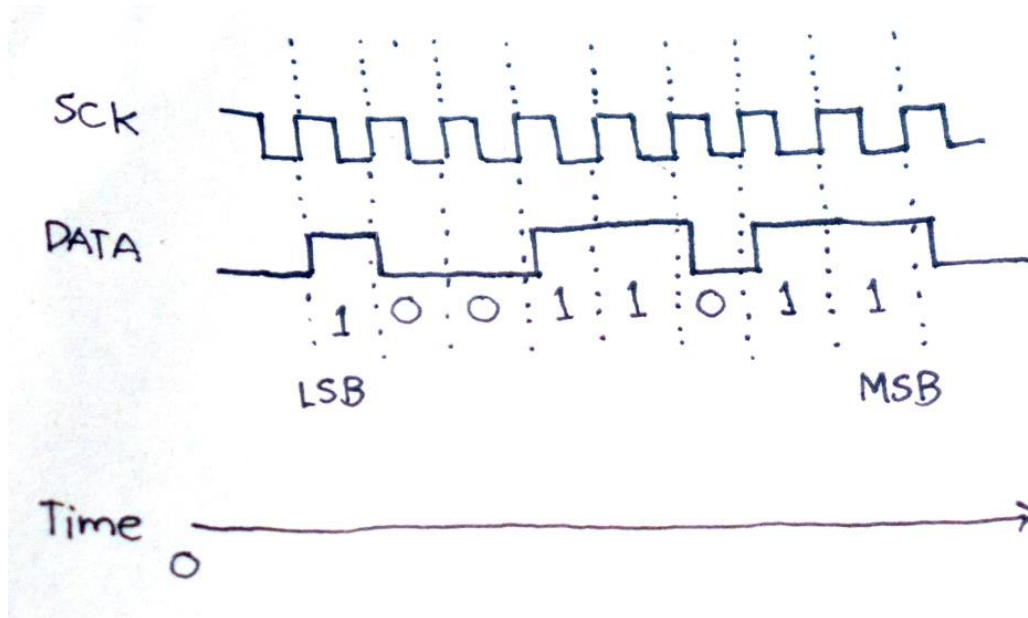
2 TYPES OF SERIAL COMMUNICATION

1. Synchronous
2. Asynchronous

RASPBERRY PI – ARDUINO - LM35 CHECK FOR SERIAL PORT

- dmesg | grep tty

- python -m serial.tools.list_ports

Serial Port	Description
/dev/ttyS0	GPIO Serial
/dev/ttyACM0	Arduino Serial
/dev/ttyUSB0	Gizduino Serial

```python
import serial
import time

try:
        serialData = serial.Serial('/dev/ttyACM0', 38400)

        time.sleep(3)

        while True:

                userInput = input("Input a letter then press enter")

                serialData.write(userInput.encode())

                data = serialData.readline()

                dataDecoded = data.decode('utf-8')

                dataDecoded = dataDecoded[:-2]

                temperature = float(dataDecoded)

                print(temperature)

except KeyboardInterrupt as e:
        serialData.close()
        print(str(e))

except Exception as e:
        serialData.close()
        print(str(e))
```

RASPBERRY PI SERIAL NOTES

- Upgrade Raspbian

- From initialization of Serial wait before reading or writing to Serial (4 seconds)

EXERCISE

- Remove the user input

- Remove the checkers

- Temperature sampling is 1 Hz (every 1 second)

IoT (Internet of Things)

- The Internet of Things (IoT) is the **internetworking** of
  - physical devices,
  - vehicles,
  - buildings, and
  - other items

- embedded with
  - electronics,
  - software,
  - sensors,
  - actuators, and
  - network connectivity

- that enable these objects to collect and exchange data.

  -- Wikipedia

CAYENNE (MYDEVICES.COM)

- MyDevices – IoT solutions company

- Cayenne – product of MyDevices
    - Drag & Drop IoT Project Builder

INSTALLATION OF CAYENNE

- Go to mydevices.com
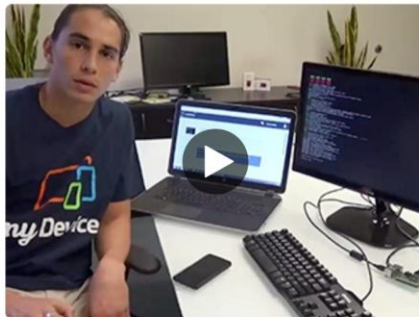- Sign Up Free

INSTALLATION OF CAYENNE
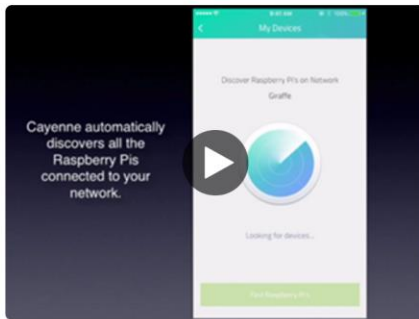
Cayenne

OPTION 1:

myDevices Cayenne Smartphone App

Our smartphone app can be used to automatically locate and install myDevices Cayenne on your Pi.

Download on the App Store　　GET IT ON Google Play

OR

OPTION 2:

Terminal / SSH

To download and install myDevices Cayenne on your Pi, use the Terminal on your Pi or SSH. Run the following commands:

```
wget https://cayenne.mydevices.com/dl/rpi_null.sh
sudo bash rpi_.sh -v
```

Back

INSTALLING CAYENNE

OPTION 2:

Terminal / SSH

To download and install myDevices Cayenne on your Pi, use the Terminal on your Pi or SSH. Run the following commands:

```
wget https://cayenne.mydevices.com/dl/rpi_7ebfed2rxj.sh
sudo bash rpi_7ebfed2rxj.sh -v
```

Note:
- For every account this link is unique.
- Installation may take a while.
- It will automatically reboot

CAYENNE INTERFACE (DESKTOP)

CAYENNE INTERFACE (GPIO)

Raspberry Pi
OS: raspbian 8 (debian) | Hardware: Pi 3 Model B

Pin	Mode	Device	Name	Value	Name	Device	Mode	Pin
●			V33		V50			●
I2C			GPIO 2 SDA		V50			●
I2C			GPIO 3 SCL		GND			●
1-W			GPIO 4 ONEWIRE		GPIO 14 TX			UART
●			GND		GPIO 15 RX			UART
●	IN		GPIO 17 P17	LOW	LOW GPIO 18 P18		IN	●
●	IN		GPIO 27 P27	LOW	GND			●
●	IN		GPIO 22 P22	LOW	LOW GPIO 23 P23		IN	●
●			V33		LOW GPIO 24 P24		IN	●
SPI			GPIO 10 MOSI		GND			●
SPI			GPIO 9 MISO		GPIO 25 P25		IN	●
SPI			GPIO 11 SCLK		GPIO 8 CE0			SPI
●			GND		GPIO 7 CE1			SPI

CAYENNE INTERFACE (GPIO)
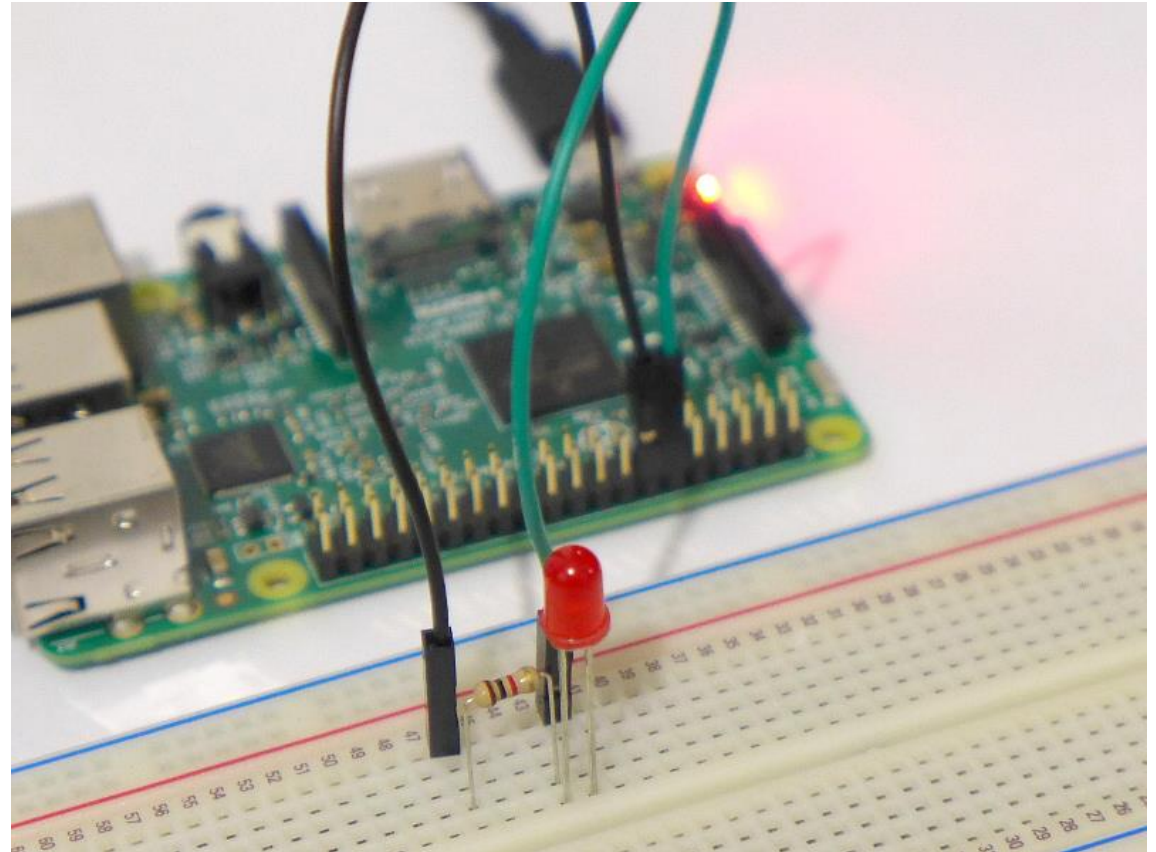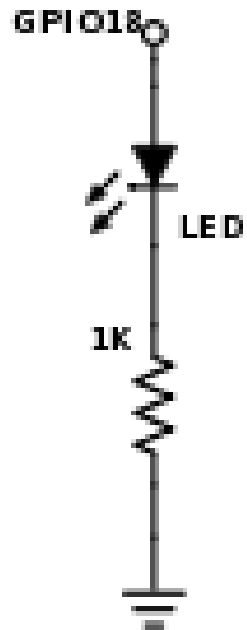
Cayenne on Mobile

- Install Cayenne on your Android or iPhone.

- Connect your phone to your Raspberry Pi.

Adding a Button

- Add New >> Device/Widget

- Actuators >> Light

- Channel 18
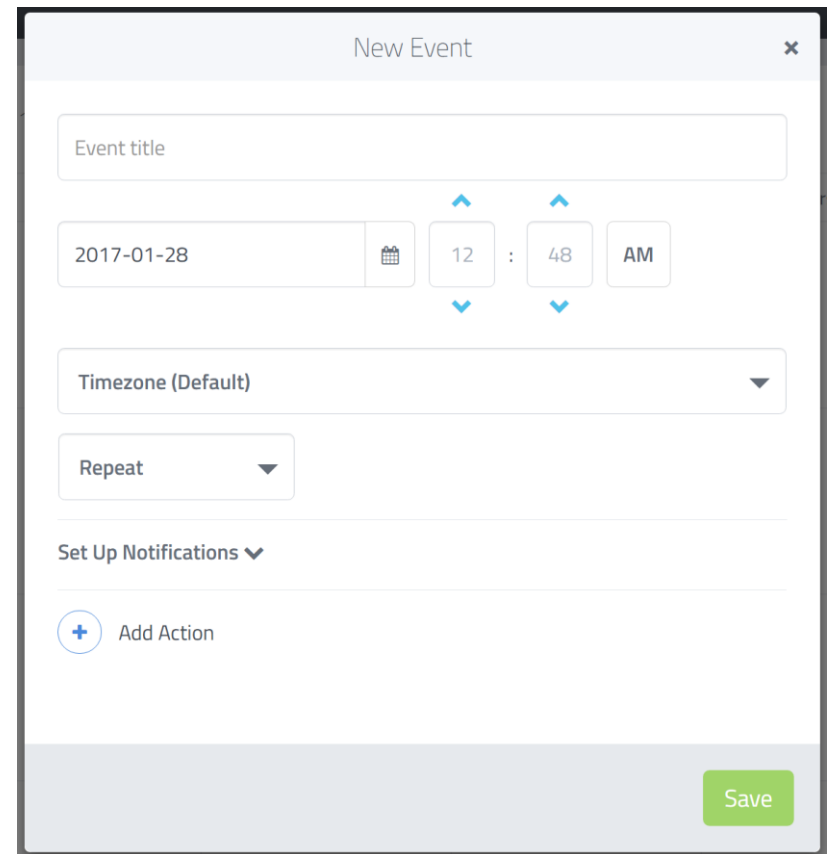
- Choose Icon >> Light

- Add Actuator

CAYENNE (EVENT)

Add New >> Event

Create an Event that would turn the LED On after 2 minutes.

Create an Event that would turn the LED Off after a minute it was turned On.

New Event

Event title

2017-01-28   12 : 48   AM

Timezone (Default)

Repeat

Set Up Notifications ⌄

＋ Add Action

Save

CAYENNE (TRIGGER)

ADDING A DEVICE THAT IS NOT IN CAYENNE'S LIST

- Cayenne MQTT Python Library

Install (3 ways):

  - sudo pip install cayenne-mqtt **(Python 2)**

  - git clone https://github.com/myDevicesIoT/Cayenne-MQTT-Python
    cd Cayenne-MQTT-Python
    sudo python setup.py install

- MQTT Cayenne Client Documentation
        pydoc cayenne.client

ADDING A DEVICE THAT IS NOT IN CAYENNE'S LIST

- Add New >> Device/Widget

- Bring Your Own Thing

CAYENNE API

Bring Your Own Thing
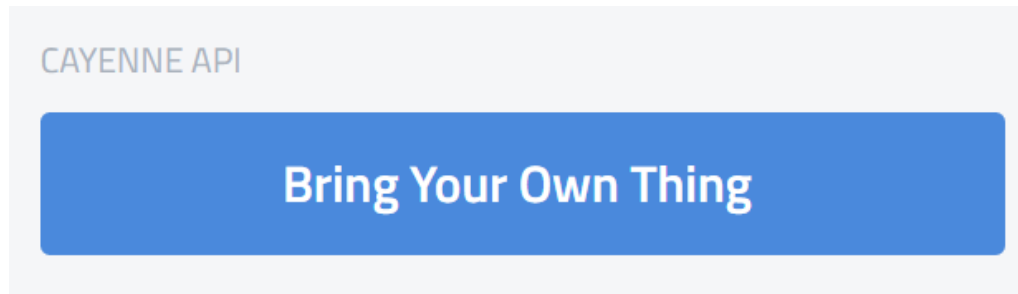
- Note the following
  - MQTT Username
  - MQTT Password
  - Client ID

ADDING A DEVICE THAT IS NOT IN CAYENNE'S LIST

```python
1  import cayenne.client
2  import time
3
4  MQTT_USERNAME  = "MQTT_USERNAME"
5  MQTT_PASSWORD  = "MQTT_PASSWORD"
6  MQTT_CLIENT_ID = "MQTT_CLIENT_ID"
7
8  client = cayenne.client.CayenneMQTTClient()
9
10 client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID)
11
12 data=0
13 timestamp = 0
14
15 while True:
16     client.loop()
17
18     if (time.time() > timestamp + 10):
19         client.celsiusWrite(1, data)
20         timestamp = time.time()
21         data = data + 1
```

EXERCISE

- Integrate our Arduino – LM35 Temperature Sensor to send data to Cayenne and show it as a graph.

- Sampling Rate: 0.5 Hz (a sample every 2 seconds)