Pi Sensors Cloud Data with ThingSpeak

Contents

ThingSpeak Sensor Data Example

http://www.billthecomputerguy.com/gopigo

http://www.billthecomputerguy.com/iot

https://sites.google.com/view/williamaloringwncc

Create ThingSpeak Account

ThingSpeak.com is a free cloud service that can be used to collect and display data from the GoPiGo3. You can create a maximum of 4 channels with 8 data fields per channel.

1. Go to www.thingspeak.com Create a free account.
2. Go to **My Profile**. Edit and change your time zone to your local time zone.

Setup ThingSpeak Channel

1. Logon to your **ThinkSpeak** account.
2. Click **New Channel** to create a new channel. Give it a name.
3. Field 1: **Sensor** Click **Save Channel**.

4. Click the **API Keys** tab. Copy the **Write API Key**. We will use this key to upload data to this channel.

ThingSpeak API Key

We are going to upload sensor data to our ThingSpeak channel.

Create the following file to hold your **Write API Key** for the channel you are using. You can keep multiple API keys in this file, just be sure to give each one a different name.

```
# thingspeak_api_key.py
# ThingSpeak channel write api key

# GoPiGo Sensor Channel
THINGSPEAK_API_KEY = 'Your Thinkspeak Key'

THINGSPEAK_URL = 'http://api.thingspeak.com/update'
```

Tutorial 1: BME280 ThingSpeak

Make a copy of your **bme280_read_3.py to bme280_ts.py** if you have it.

```python
#!/usr/bin/env python3
"""
    Name: bme280_ts.py
    Purpose: Use Pimoroni library to read
    temperature, pressure, and humidity from Bosch bme280 sensor
    !Connect to I2C bus
    Press Ctrl+C to exit
"""
import api_key_ts
# sudo pip3 install requests
import requests
from time import sleep
try:
    from smbus2 import SMBus
except ImportError:
    from smbus import SMBus

# sudo pip3 install pimoroni-bme280
from bme280 import BME280

# Initialize the BME280 sensor
bus = SMBus(1)
sensor = BME280(i2c_dev=bus)

# Substitute your api key in this file for updating your ThingSpeak channel
TS_KEY = api_key_ts.THINGSPEAK_API_KEY

# Create ThingSpeak data dictionary
ts_data = {}

print("BME280 send TPH to ThingSpeak")
print("Ctrl+C to exit!")
```

```python
def main():
    try:
        while True:
            # Temperature in celsius
            temp_c = sensor.get_temperature()
            # Convert celsius to fahrenheit
            temp_f = ((temp_c * 9.0) / 5.0) + 32

            # Relative humidity in %
            humidity = sensor.get_humidity()

            # Barometric pressure in hPa (hectopascal)
            pressure_hpa = sensor.get_pressure()
            # Convert hPa hectopascals to inHg Inches of Mercury
            pressure_inhg = pressure_hpa / 33.863886666667
            # Compensate for 3960' altitude 4.04
            # Scottsbluff, NE, Heilig Field, 4.04
            pressure_inhg = pressure_inhg + 4.04

            print(f" {temp_f:.1f} °F | {humidity:.1f}% | {pressure_inhg:.2f} inHg")

            # Send sensor data to ThingSpeak
            thingspeak_send(temp_f, humidity, pressure_inhg)

            sleep(20)

    except KeyboardInterrupt:
        print("Bye!")
        exit(0)
```

```python
def thingspeak_send(temp, hum, bp):
    """Update the ThingSpeak channel using the requests library"""
    print("Update Thingspeak Channel")

    # Each field number corresponds to a field in ThingSpeak
    params = {
        "api_key": TS_KEY,
        "field1": temp,
        "field2": hum,
        "field3": bp
    }

    # Update data on Thingspeak
    ts_update = requests.get(
        "https://api.thingspeak.com/update",
        params=params
    )

    # Was the update successful?
    if ts_update.status_code == requests.codes.ok:
        print("Data Received!")
    else:
        print("Error Code: " + str(ts_update.status_code))

    # Print ThingSpeak response to console
    # ts_update.text is the thingspeak data entry number in the channel
    print(f"ThingSpeak Channel Entry: {ts_update.text}")


# If a standalone program, call the main function
# Else, use as a module
if __name__ == '__main__':
    main()
```

Tutorial 1: BME680 ThingSpeak

Make a copy of your **bme680_read_3.py to bme680_ts.py** if you have it.

```python
#!/usr/bin/env python3
"""
    Filename: bme680_ts.py
    Description: Display temperature, pressure, and humidity
    from Bosch bme680 sensor
    !Connect to I2C bus
    Press Ctrl+C to exit
"""
import api_key_ts
# sudo pip3 install requests
import requests
from time import sleep

# sudo pip3 install bme680
import bme680

# Initialize sensor object, make connection to sensor over I2C
sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)

# Substitute your api key in this file for updating your ThingSpeak channel
TS_KEY = api_key_ts.THINGSPEAK_API_KEY

# Create ThingSpeak data dictionary
ts_data = {}

print("BME680 send TPH to ThingSpeak")
print("Ctrl+C to exit!")
```

```
def main():
    try:
        while True:
            # Can the sensor data can be retrieved successfully?
            if sensor.get_sensor_data():
                # If sensor data retrieval is successful,
                # retrieve and display the data

                # Sensor output in celsius
                temp_c = sensor.data.temperature
                # Convert celsius to fahrenheit
                temp_f = ((temp_c * 9.0) / 5.0) + 32

                # Relative humidity in %
                humidity = sensor.data.humidity

                # Sensor output in hectoPascals (hPa), also called millibars
                pressure_pascals = sensor.data.pressure
                # Convert hPa hectopascals to inHg Inches of Mercury
                pressure_inhg = pressure_pascals / 33.863886666667
                # Compensate for 3960' altitude 4.04
                # Scottsbluff, NE, Heilig Field, 4.04
                pressure_inhg = pressure_inhg + 4.04

                print(
                    f" {temp_f:.1f} °F | {humidity:.1f}% | {pressure_inhg:.2f} inHg")

                # Send sensor data to ThingSpeak
                thingspeak_send(temp_f, humidity, pressure_inhg)

                # 20 seconds is the minimum amount of time between uploads
                # Sleep is set to 20 seconds for testing purposes
                sleep(20)

    except KeyboardInterrupt:
        print("Bye!")
        exit(0)
```

```python
def thingspeak_send(temp, hum, bp):
    """Update the ThingSpeak channel using the requests library"""
    print("Update Thingspeak Channel")

    # Each field number corresponds to a field in ThingSpeak
    params = {
        "api_key": TS_KEY,
        "field1": temp,
        "field2": hum,
        "field3": bp
    }

    # Update data on Thingspeak
    ts_update = requests.get(
        "https://api.thingspeak.com/update",
        params=params
    )

    # Was the update successful?
    if ts_update.status_code == requests.codes.ok:
        print("Data Received!")
    else:
        print("Error Code: " + str(ts_update.status_code))

    # Print ThingSpeak response to console
    # ts_update.text is the thingspeak data entry number in the channel
    print(f"ThingSpeak Channel Entry: {ts_update.text}")


# If a standalone program, call the main function
# Else, use as a module
if __name__ == '__main__':
    main()
```

Upload the Sensor Data

1. Open a terminal
2. **python3 bme680_ts.py**

Go to your ThingSpeak channel. Your data should show up almost immediately.

What's Next?

Work with other sensors to read and upload the data.