GoPiGo Cloud Data with ThingSpeak

Contents

ThingSpeak Sensor Data Example

http://www.billthecomputerguy.com/gopigo

Create ThingSpeak Account

ThingSpeak.com is a free cloud service that can be used to collect and display data from the GoPiGo. 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 timezone.

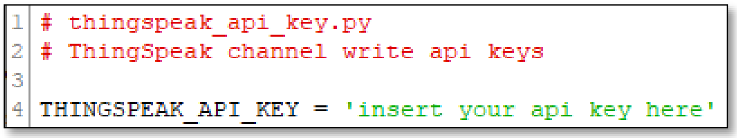
Setup ThingSpeak Channel

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

Create ThingSpeak Python Program

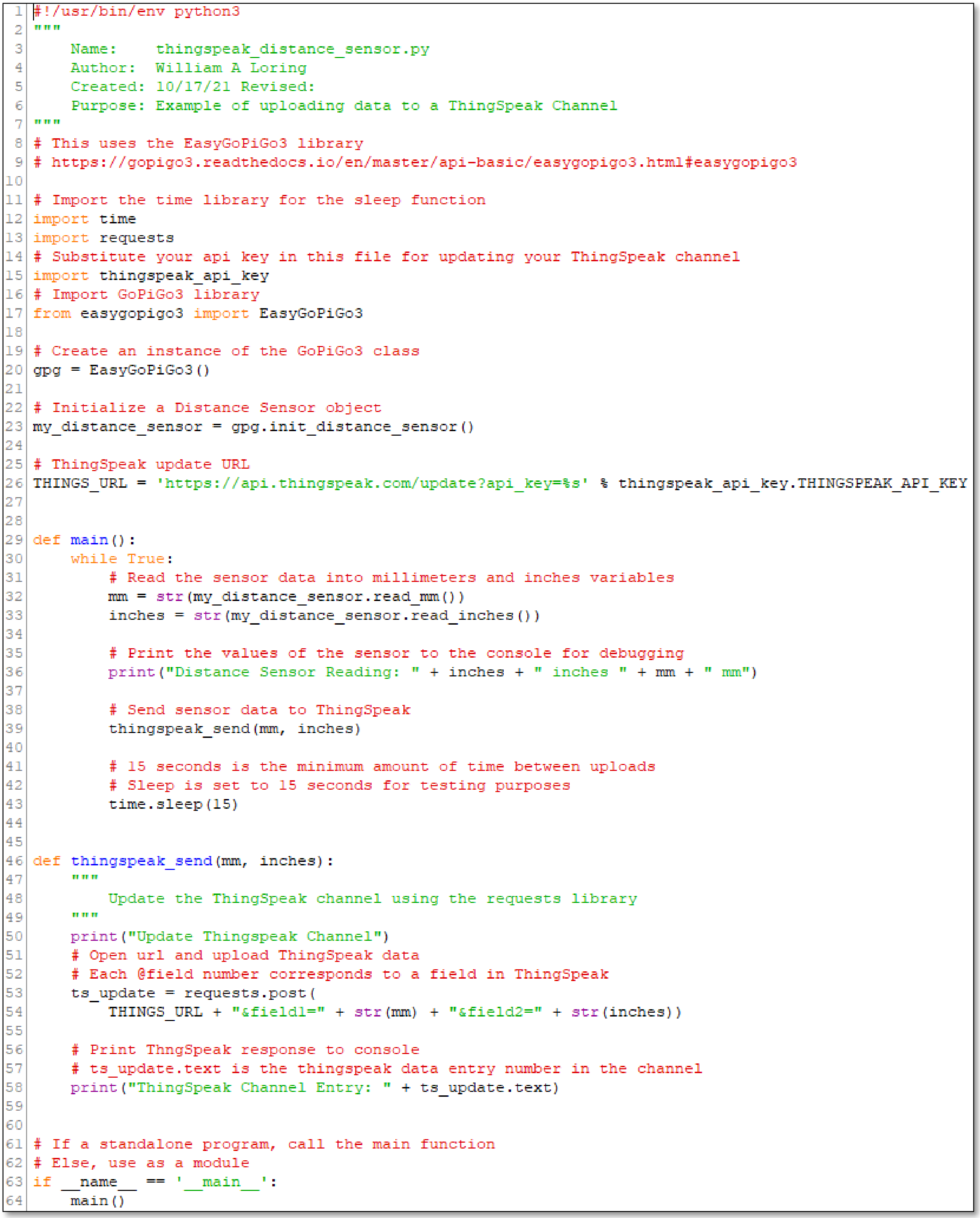
We are going to upload Distance Sensor Data to our ThingSpeak channel.

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 keys

THINGSPEAK_API_KEY = 'insert your api key here'
```



```python
#!/usr/bin/env python3
"""
    Name:    thingspeak_distance_sensor.py
    Author:  William A Loring
    Created: 10/17/21 Revised:
    Purpose: Example of uploading data to a ThingSpeak Channel
"""
# This uses the EasyGoPiGo3 library
# https://gopigo3.readthedocs.io/en/master/api-basic/easygopigo3.html#easygopigo3

# Import the time library for the sleep function
import time
import requests
# Substitute your api key in this file for updating your ThingSpeak channel
import thingspeak_api_key
# Import GoPiGo3 library
from easygopigo3 import EasyGoPiGo3

# Create an instance of the GoPiGo3 class
gpg = EasyGoPiGo3()

# Initialize a Distance Sensor object
my_distance_sensor = gpg.init_distance_sensor()

# ThingSpeak update URL
THINGS_URL = 'https://api.thingspeak.com/update?api_key=%s' % thingspeak_api_key.THINGSPEAK_API_KEY


def main():
    while True:
        # Read the sensor data into millimeters and inches variables
        mm = str(my_distance_sensor.read_mm())
        inches = str(my_distance_sensor.read_inches())

        # Print the values of the sensor to the console for debugging
        print("Distance Sensor Reading: " + inches + " inches " + mm + " mm")

        # Send sensor data to ThingSpeak
        thingspeak_send(mm, inches)

        # 15 seconds is the minimum amount of time between uploads
        # Sleep is set to 15 seconds for testing purposes
        time.sleep(15)


def thingspeak_send(mm, inches):
    """
        Update the ThingSpeak channel using the requests library
    """
    print("Update Thingspeak Channel")
    # Open url and upload ThingSpeak data
    # Each @field number corresponds to a field in ThingSpeak
    ts_update = requests.post(
        THINGS_URL + "&field1=" + str(mm) + "&field2=" + str(inches))

    # Print ThngSpeak response to console
    # ts_update.text is the thingspeak data entry number in the channel
    print("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

Run the program. Move the GoPiGo around by hand or by a remote control program. Go to your ThingSpeak channel. Your data should show up almost immediately.