

# IoT Platforms

Frank Walsh

# What are IoT platforms

- IoT applications combine sensors, devices, data, analytics and integrations in a seamless and unified way
  - e.g. your project!
- IoT Platforms provide software tools and components to:
  - connect sensors, devices, and data networks
  - Analyse and store data
  - Integrate with other apps
- So what? We know the tech for that now (I2C, SPI, BLE, MQTT, Python...)
- Main selling point of an IoT platform is software that it
  - accelerates the IoT development process
  - Focuses on IoT: brings in best of breed features
  - Provides initial scaffolding for IoT projects

# What are IoT Platform

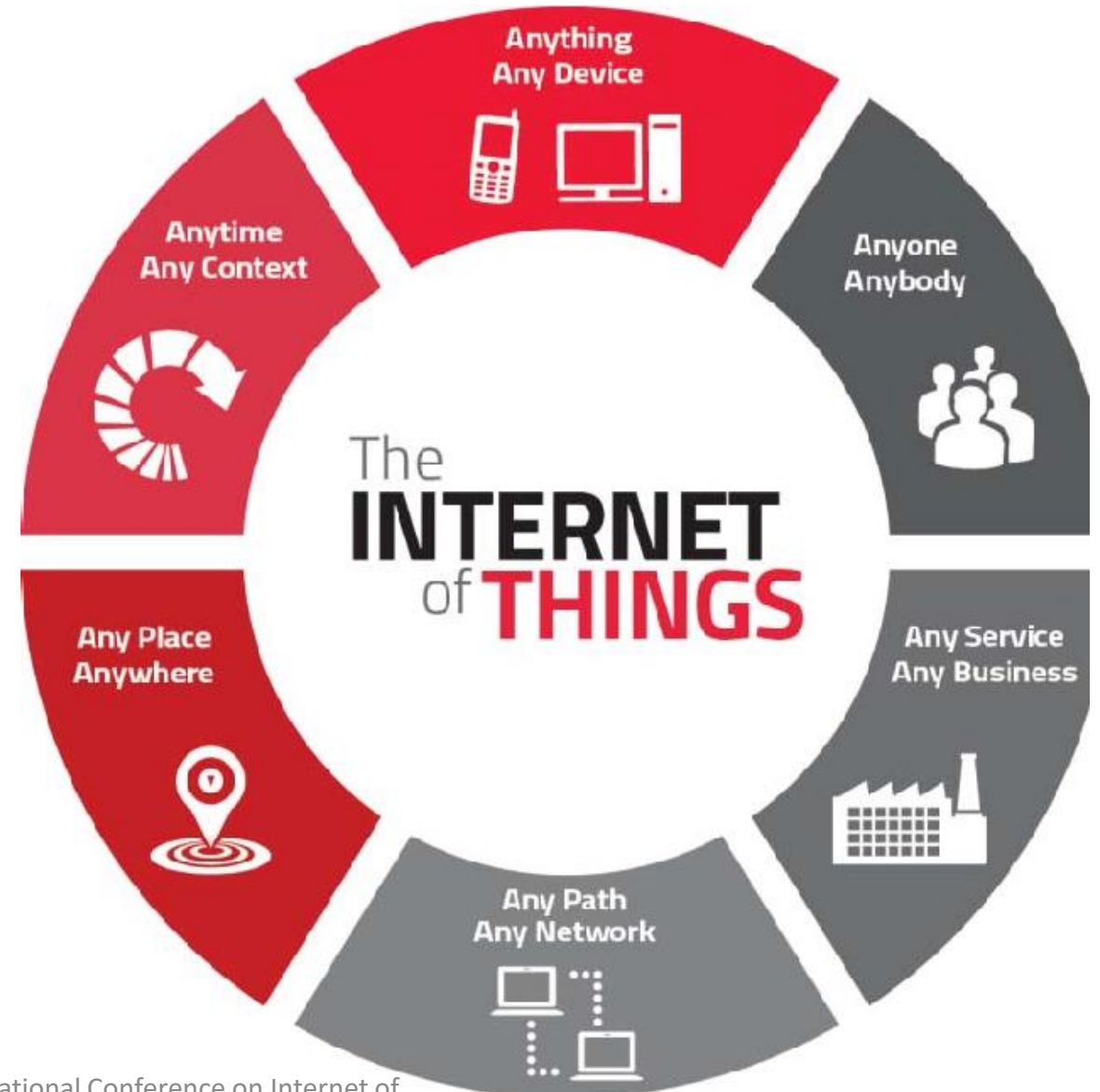
---



- Many(not all) are cloud-based platforms that require subscription
- Provide device/language agnostic set of Software Development kits
  - Arduino/RPi/beagleboard
- IoT development is generally iterative:
  - Starts with initial simple use case
  - Once operational, data/insights result in new usecases
- IoT platforms should promote scalable, iterative development
  - Allow for quick app development
  - Ability to adapt/optimize apps quickly

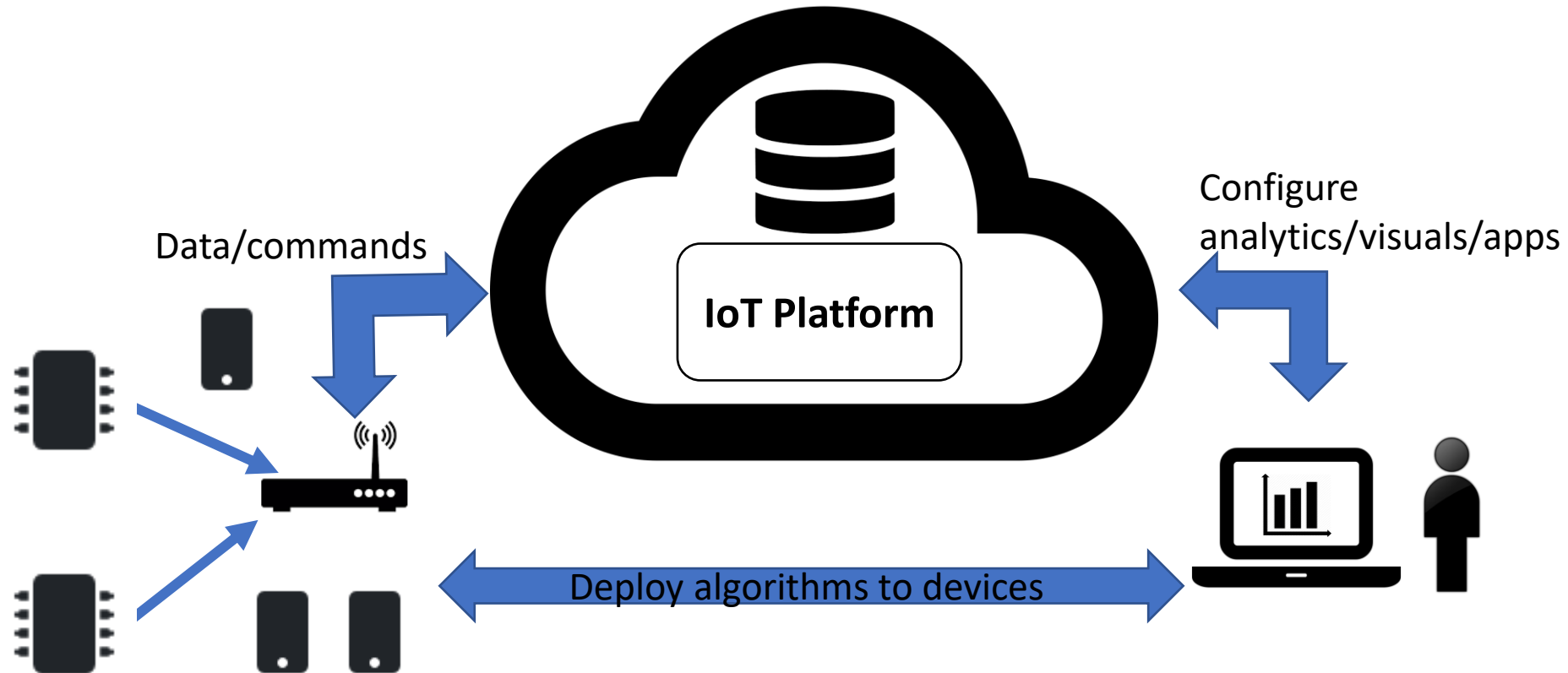
# IoT Platform Characteristics

- Manage many concurrent device connections
- Connectivity across several connection types
- "Off-the-peg" IoT protocol stack
- Manage/analyse/visualise data
- Integrations to other services/apps
- App Development



Published in 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

# IoT Platform – general architecture



# IoT Platform Advantages

- Use software component that has been pre-built and pre-tested. This increases the reliability of your application and reduces development effort.
- IoT frameworks constantly evolve, providing new features, integrations etc.
- Encourages better "design pattern" for your IoT app.
- Predefined APIs and docs
  - Great for collaboration
- "Baked-in" standards and features:
  - Security, authentication, scalability...

# Which one?

- Connectivity
  - Does the platform provide suitable capability and integrations (WiFi/Cellular/LPWan-Sigfox)
- Maturity
  - In business for long? Critical mass in developer community?
- Free
  - Is there a free tier (handy for evaluation)?
- Service type
  - Platforms try to distinguish themselves – what specialisms/USP does it have?
- Security
  - What security model do they use? Is there security issues reported in past?
- Geographic area
  - Does it operate well at your location (can you select edges/data centres)





Examples:

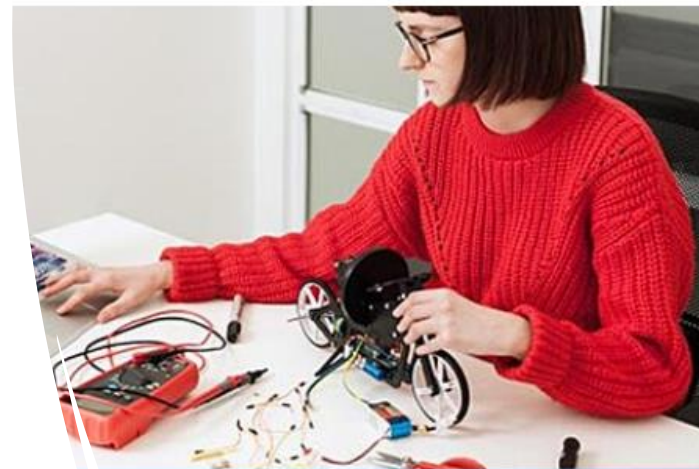




Thingspeak

# Thingspeak

- **ThingSpeak** is a cloud-based IoT platform to store and retrieve data from devices.
  - Uses HTTP protocol/Restful APIs
- "Collect and analyse data quickly and easily"



## Students and Educators

Projects quickly with built-in MATLAB data analysis tools and real-

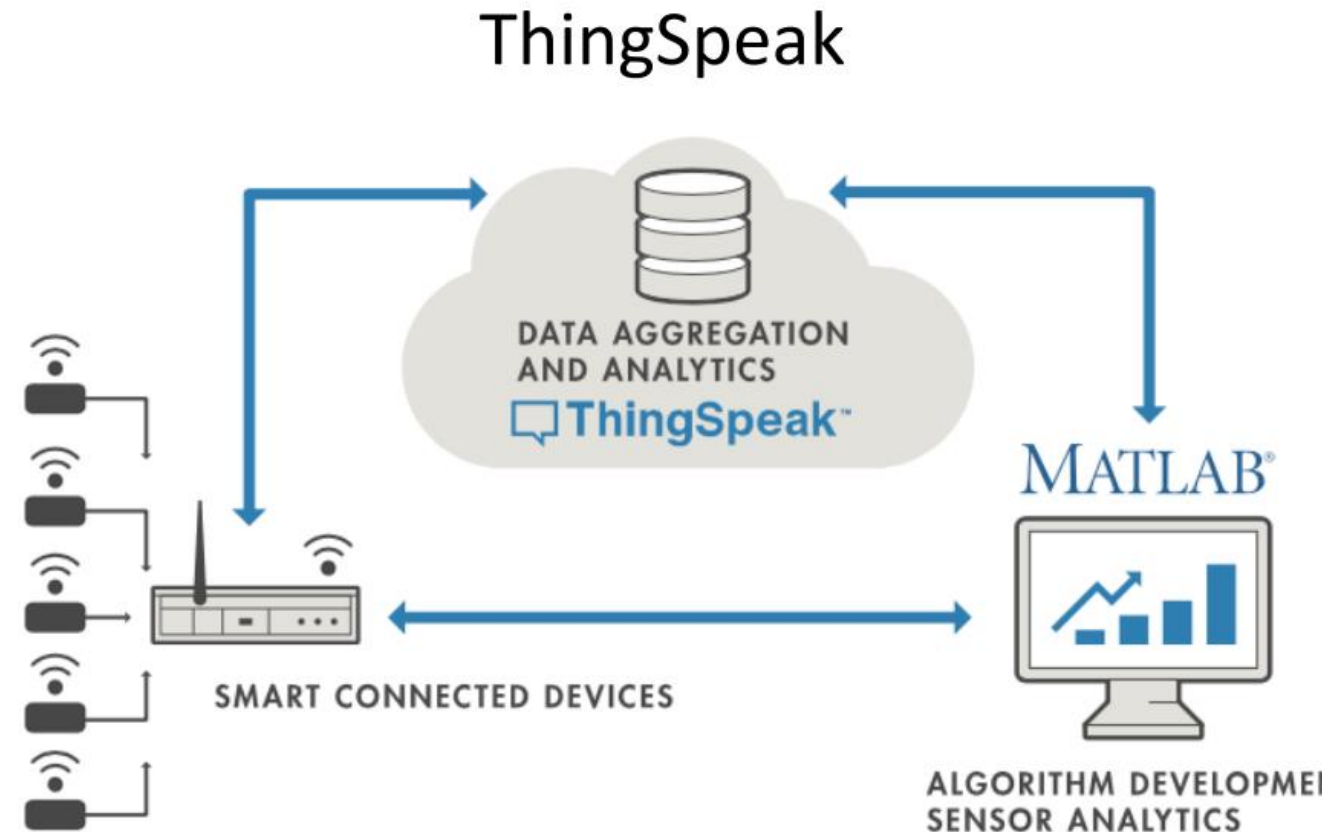


## ThingSpeak for Environmental Monitoring

Build IoT services for remote monitoring of air quality sensors, and create MA to predict pollution levels

# Thingspeak Overview

- Account-based
  - Can create free account online
- Brought to you by the people who made Matlab
  - Uses Matlab features/toolboxes
- SDKs/librariys for popular languages/devices
- Restful API means should work with any device



# Thingspeak – basic use

## Create a new channel

- Channels collect data

## Collect data in the channel

- Devices write data to channels

## Analyse the data

- Run analytical algorithms/visualise your data

## Act on the data

- Test for certain conditions and perform actions

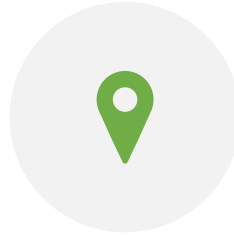
# ThingSpeak – Create new channel



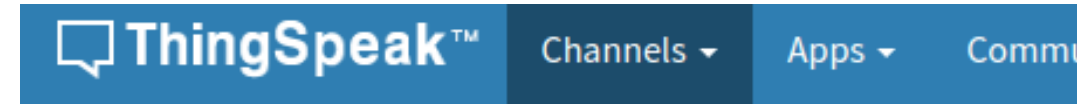
CREATE A NEW  
CHANNEL TO COLLECT  
DATA FROM DEVICES



DEFINE DATA FIELDS  
FOR THE  
CHANNEL(MAX 8)



CAN ALSO INPUT  
LOCATION(LAT/LONG)  
OF CHANNEL SOURCE



## New Channel

Name	<input type="text" value="SensePi"/>	
Description	<input type="text" value="Environment data from &lt;u&gt;senspi&lt;/u&gt;"/>	
Field 1	<input type="text" value="temperature"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="pressure"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text" value="humidity"/>	<input checked="" type="checkbox"/>

# Thingspeak - New channel

- Once saved you can access channel page:

## SensePi

Channel ID: **625505**

Author: [fxwalsh](#)

Access: Private

Environment data from senspi

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

[+ Add Visualizations](#)

[+ Add Widgets](#)

[Export recent data](#)

[MATLAB Analysis](#)

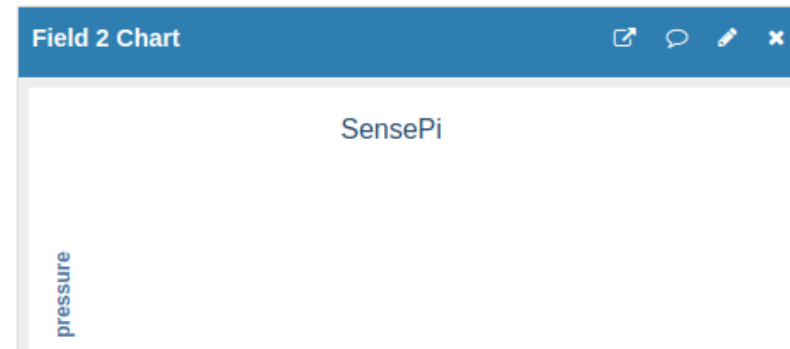
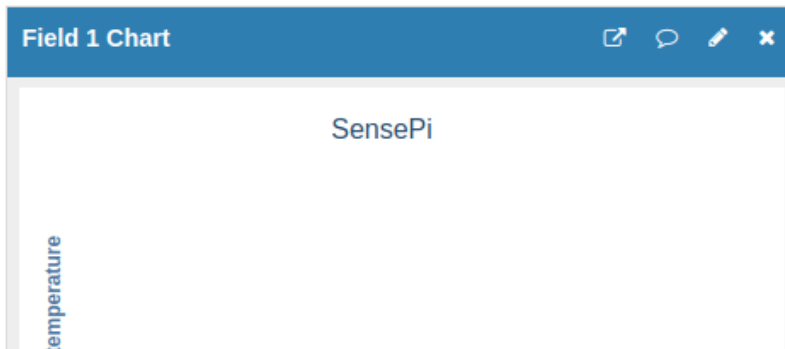
[MATLAB Visualization](#)

## Channel Stats

Created: 4 minutes ago

Updated: 4 minutes ago

Entries: 0



# Thingspeak - Add data to channel

- Programmatically, many ways!
  - Construct HTTP GET request and include field
  - MQTT (see in Lab)

## API Reference

Use the REST and MQTT APIs to update ThingSpeak™ channels and to chart numeric data streams.

ThingSpeak is an IoT platform that uses channels to store data sent from apps or devices. With ThingSpeak, you can create a channel and use the REST API calls such as GET, POST, PUT, and DELETE, you can create a channel and receive messages when a channel updates. Learn more about when to [Choose Between REST and MQTT](#).

MATLAB® analysis and visualization apps enable you to explore and view your channel data.

### REST API

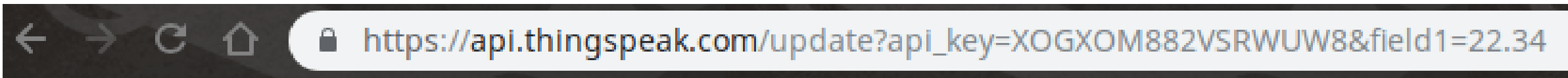
Use REST API calls to create and update ThingSpeak channels and charts

### MQTT API

Use the MQTT API to update ThingSpeak channels

GET [https://api.thingspeak.com/update?api\\_key=<WRITE-KEY>&field1=12](https://api.thingspeak.com/update?api_key=<WRITE-KEY>&field1=12)

- Using HTTP GET request, can test from a browser:



# Thingspeak – Add data with python

- Make HTTP request from Python:

```
def writeData(temp,press,hum):  
    # Sending the data to thingspeak in the query string  
    conn = urllib2.urlopen(baseURL + '&field1=%s&field2=%s&field3=%s' % (temp, hum,press))  
    print(conn.read())  
    # Closing the connection  
    conn.close()  
  
while True:  
    temp=round(sense.get_temperature(),2)  
    press=round(sense.get_pressure(),2)  
    hum=round(sense.get_humidity(),2)  
    writeData(temp,press,hum)  
    time.sleep(60)
```

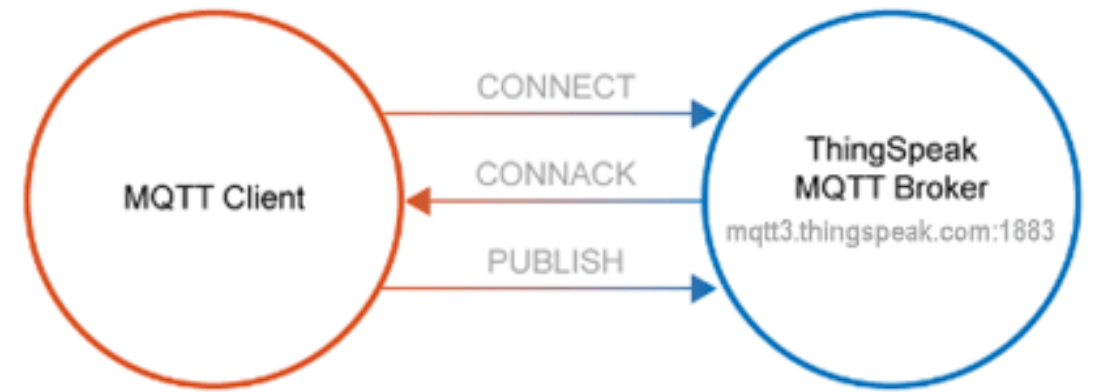


# Thingspeak – Add data with python

Create a MQTT client:

```
# connect to Broker
mqttc.username_pw_set(config["username"], config["password"])
mqttc.connect("mqtt3.thingspeak.com", 1883)
mqttc.loop_start()
topic = "channels/1558158/publish"

# Publish a message to temp every 15 seconds
while True:
    try:
        temp=round(sense.get_temperature(),2)
        payload="field1="+str(temp)
        mqttc.publish(topic, payload)
        time.sleep(15)
    except:
        print('Interrupted')
        try:
            sys.exit(0)
```



## 1. Publish to a channel feed

channels/<channelID>/publish.

## 2. Publish to a channel field

channels/<channelID>/publish/fields/field<fieldnumber>

# Which to use: HTTP or MQTT API

- HTTP (REST) API
  - You need to retrieve any historical data, such as data within a defined time range.
  - You want to Create Channel, Read Settings, Clear Channel, Delete Channel, or Create Chart.
  - You need a response!
- MQTT
  - Your device is power-constrained, and you want lower battery consumption to send data to ThingSpeak. Also, an MQTT PUBLISH operation is typically faster in this scenario.
  - Your device connectivity is intermittent, and you have limited bandwidth usage.
  - You want immediate updates of data posted to a channel.
  - You want messages pushed to you instead of needing to poll the server for new messages.

# Think Speak – Analyse data

- Thingspeak will visualise each field by default in channel view

## SensePi

Channel ID: 625505

Environment data from senspi

Author: [fxwalsh](#)

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

Export recent data

MATLAB Analysis

MATLAB Visualiz

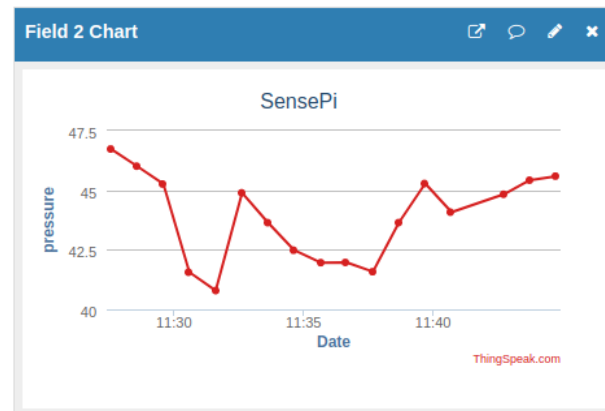
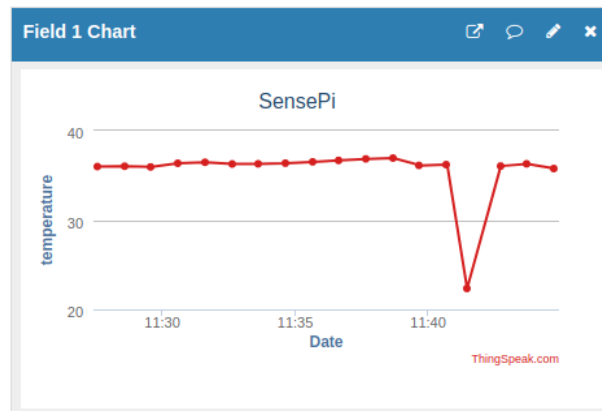
## Channel Stats

Created: [about an hour ago](#)

Updated: [about an hour ago](#)

Last entry: [about a minute ago](#)

Entries: 17



# Thingspeak - Apps

- The Apps tab provides various mechanism to transform, analyse, visualise and act on data.

## Analytics



### MATLAB Analysis

Explore and transform data.



### MATLAB Visualizations

Visualize data in MATLAB plots.



### Plugins

Display data in gauges, charts, or custom plugins.

## Actions



### ThingTweet

Connect a device to Twitter® and send alerts.



### TimeControl

Automatically perform actions at predetermined times with ThingSpeak apps.



### React

React when channel data meets certain conditions.

# ThingSpeak Example: ThingTweet

- Link Twitter account to Thingspeak
- Create a **React** to tweet when a certain condition is met.
- Also tweet from device using HTTP POST:

```
POST https://api.thingspeak.com/apps/thingtweet/1/statuses/update
api_key=WLKX9PPPWCRZ6H1
status=I just posted this from my thing!
```

Apps / React / Fermenting Beer Too Cold

Edit React

Name:	Fermenting Beer Too Cold
Condition Type:	Numeric
Test Frequency:	Every 30 minutes
Last Ran:	
Channel:	SensePi
Condition:	Field 1 (temperature) is less than 17
Message:	frankwalsh59: BEER TO COLD!!!
Options:	Only the first time the condition is met

# ThingSpeak Example: Analysis

- Can write Matlab Code to analyse and transform data
- Possible uses:
  - Clean data (remove outliers)
  - Statistical analysis
  - Transformations
  - Data Fusion
- Generally write results to second channel for further analysis/visualisation.

# Thinkspeak: Convert Celcius to Fahrenheit

Convert temperature units

## MATLAB Code

```
1  
2 readChannelID = 12397;  
3 % Temperature Field ID  
4 temperatureFieldID = 4;  
5  
6 readAPIKey = '';  
7  
8 tempC = thingSpeakRead(readChannelID, 'Fields', temperatureFieldID, 'ReadKey', readAPIKey);  
9  
10 % Convert to Fahrenheit  
11 tempF = tempC*1.8+32;  
12 display(tempC, 'Temperature in Fahrenheit');  
13  
14 % Replace the [] with channel ID to write data to:  
15 writeChannelID = 1234;  
16 % Enter the Write API Key between the '' below:  
17 writeAPIKey = 'abcd';  
18  
19 thingSpeakWrite(writeChannelID, [tempF, tempC], 'Writekey', writeAPIKey);
```

Save and Run

Save

# Other Platforms

- Ubidots
- Amazon Web Services
- Microsoft Azure
- Evothings