



POLITECNICO
MILANO 1863



Internet of Things Lab

Lab 3: ThingSpeak and Node-Red

Agenda

- ThingSpeak
 - HTTP
 - MQTT
 - Examples
- Node-Red
 - Basic flows
 - HTTP and MQTT
 - Dashboard
- **Challenge 2!**

What we will use

- Virtual Machine
- mosquitto_pub/sub
- Node-Red
- **Your attention 😊**



POLITECNICO
MILANO 1863



IoT Software Platforms...

... in the **CLOUD**

Why the Cloud?

- IoT is about **DATA** sensed and transmitted from **OBJECTS**
- How much data?
 - IPV6 may cover every atom on the earth
 - Recent estimates: 4.4 zettabytes (10^{21} B) of IoT data produced by 2020
 - That's 1 stack of 128 GB Ipad Air reaching the moon!
- How to manage it? **Go to the cloud with an IoT platform!**

IoT Platforms

- The term “platform” can refer to:
 - Hardware architectures (ARM, Arduino, ESP32, etc...)
 - **Software frameworks** to program smart things
 - **Cloud-based middleware** platforms to **manage IoT data** and devices

This lesson focuses on the last two!

IoT Platforms

- IoT platforms world is growing fast
 - **200+ platforms available online**
 - They are getting more and more mature and stable
 - Not only startups, **also big players** (Microsoft, IBM)
- Key value proposition
 - Reduce costs and time-to-market for IoT solution, reduce management complexity
- But: a lot of confusion for users!



Solutions

- Old Giants
 - IBM: Bluemix IoT Cloud, Node-RED
 - Microsoft: Azure IoT Suite + Windows 10 IoT Co
 - Amazon: AWS IoT
 - Google: Cloud IoT + Android Things
 - Intel: IoT platform
- Startups
 - **Thingspeak**, FreeBoard, ThingWorx, Dweet.io....

Desired Features

- **Device management**
 - Connect devices to the cloud, configure devices, update firmware, monitor devices...
- **Data management**
 - Store and retrieve data, manage events, visualize and share data
- **Data analysis / automation**
 - Statistical analysis, data mining, machine learning, etc...
- **Security**

Summary

- Many IoT platforms available
- Both startups and big IT players
- Different solutions / services for different users
- In general, it's quite a complex scenario



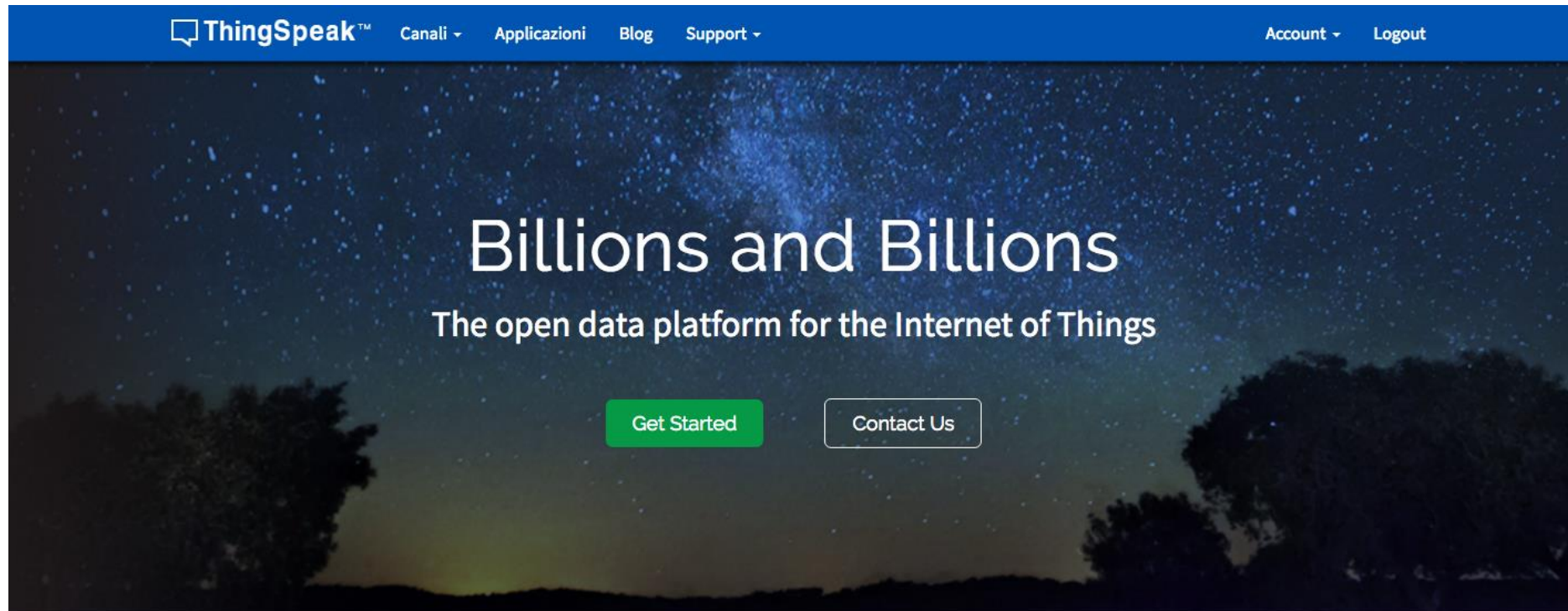
POLITECNICO
MILANO 1863



Hands-on on the cloud

Thingspeak

ThingSpeak



 **Collect**
Send sensor data to the cloud.

 **Analyze**
Analyze and visualize your data.

 **Act**
Trigger a reaction.

<https://thingspeak.com/>
(you need a MathWorks Account)

ThingSpeak

- An IoT cloud platform for:
 - Real-time data collection and storage
 - Data analytics and visualization (integrated with MATLAB)
 - Alerts and Scheduling
 - Device Communication, Open API

ThingSpeak featured projects

- Solar Home: <https://thingspeak.com/channels/34247>
Solar Power Monitoring, energy consumption in a house
- Weather station: <https://thingspeak.com/channels/895691>
Real-time weather station, based on WEMOS sensor

ThingSpeak Channels

- ThingSpeak uses channels to store data sent from devices or apps
- Data stored in channels:
 - Create / view / update / delete / import / export
- Channel settings
 - ID, name, description
 - Up to 8 fields (datastreams)
 - Public / private
 - Location, Metadata, etc...

ThingSpeak Channels

- Operations can be done from the web interface or through the REST API
- Create a channel:
 - Send an HTTP POST to:
<https://api.thingspeak.com/channels>
 - With parameters
 - user_api_key=XXXXXXXXXXXXXXXXXXXX*
 - name=My New Channel

*Find the user API-KEYS in your profile in ThingSpeak

Channel API Keys

- When performing an operation, your **API key** is required
- Keys can be regenerated if the user feels that they are compromised
- Keys are in:
 - channel-> “API Keys”
 - “My Profile” tab

Write API Key

Key

ABVIHE8AV0KXACK6

[Generate New Write API Key](#)

API Keys

User API Key

GKTNQMKLKYLS39PQ



MQTT API Key

LV93PN44QLFS5ZNT



Alerts API Key

<no API key>



Channel Fields

- Each channel supports up to 8 fields
- Each field represents an individual sensor stream and is associated to a chart
 - e.g., temperature, humidity, number of parked cars, etc...
- Field/Chart parameters:
 - Title, x axis, y axis, appearance, etc...

Channel Example

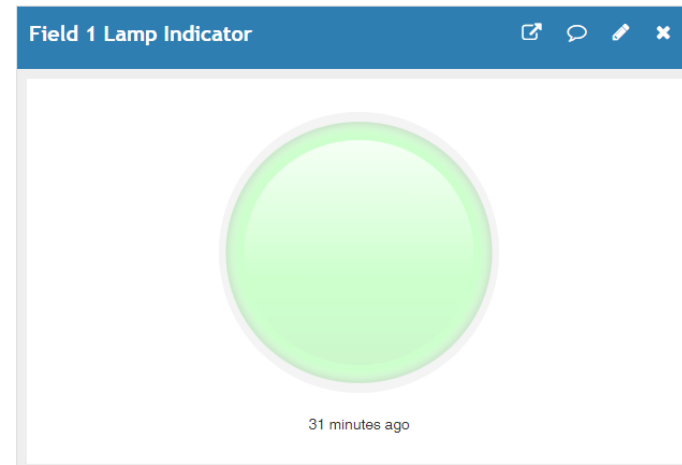
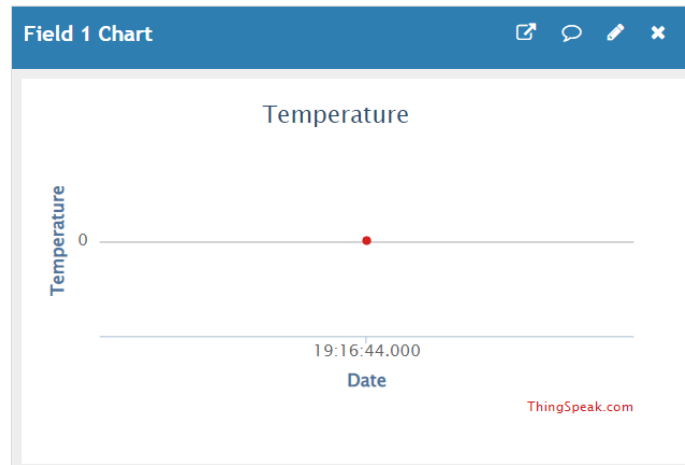
- Create an account on ThingSpeak
- Create a channel called “Weather Monitor” with two fields called “Temperature” and “Humidity”
- Create Widgets (lamps ...)

Channel Stats

Created: [32 minutes ago](#)

Last entry: [31 minutes ago](#)

Entries: 1



Channel Field update: HTTP

- Like channels, fields can be updated via HTTP requests:
 - POST version
 - POST `https://api.thingspeak.com/update.json \`
`write_api_key=XXXXXXXXXXXX \`
`field1=YY`

CHANNEL API KEY
 - GET version
 - GET `https://api.thingspeak.com/update?api_key=XXXXXXXXXXXX&field1=YY`

CHANNEL
API KEY

Value

Please note: ThingSpeak limits incoming messages to 1 message every 20/30 seconds

Channel Field update: HTTP

- **cURL**: CLI tool for transferring data using HTTP
- Remember to **escape** “special” characters

```
curl https://api.thingspeak.com/update\?write_api_key\=YOUR_API_KEY\&field1\=XXXX
```

MQTT with ThingSpeak

- First let's create an **MQTT Device in ThingSpeak**:
 - **Devices** ->**MQTT** and add a **new Device** (add your channel in the new device list)
 - **Save the MQTT credentials** (**id,user,psw**) to be used later
- Then to change values of the fields we use the MQTT Publish:

```
mosquitto_pub -h "mqtt3.thingspeak.com" -p 1883 -u <MQTT_USER>  
-P <MQTT_PSW> -i <MQTT_ID> -t "channels/<CHANNEL_ID>/publish"  
-m "field1=XX&field2=YY&status=MQTTPUBLISH"
```

MQTT with ThingSpeak

Check all the ThingSpeak MQTT API here

https://it.mathworks.com/help/thingspeak/mqtt-api.html?s_tid=CRUX_lftnav

You find how to:

- Publish channel feed (more field at once)
- Publish channel single field
- Subscribe channel feed (all info)
- Subscribe to channel fields (single or wildcard)

Other IoT Platforms...

- <https://thingsboard.io>
- <https://thinger.io>
- <https://www.kaaproject.org>
- <https://www.ptc.com/en/products/iiot/thingworx-platform>
- <https://freeboard.io>



POLITECNICO
MILANO 1863



Node-RED

Node-RED

- Visual tool for wiring the Internet of Things
- Can be run on a variety of platforms:
 - Edge devices: Raspberry PI, BeagleBone black
 - Cloud: IBM Bluemix, Amazon AWS, Azure
- The user creates **flows** by wiring together different nodes:
 - **I/O** (serial port, tcp sockets, http, mqtt, files)
 - **Functions** (built in or custom JavaScript functions)
 - **Advanced** (execute programs, post a tweet)

Node-RED Flow

The screenshot displays the Node-RED web interface. On the left, a 'Nodes' palette is visible, categorized into 'input' and 'output' sections. The 'input' section includes nodes like 'inject', 'catch', 'status', 'link', 'mqtt', 'http', 'websocket', 'tcp', 'udp', and 'coap'. The 'output' section is partially visible. The main workspace, labeled 'Flow Area', contains two tabs: 'Flow 1' and 'Flow 2'. A red callout box labeled 'Flow Tabs' points to these tabs. In the top right corner, a red 'Deploy' button is highlighted with a red callout box labeled 'Deploy button'. A dark sidebar menu is open on the right, showing options like 'View', 'Import', 'Export', 'Search flows', 'Configuration nodes', 'Flows', 'Subflows', 'Manage palette', 'Settings', 'Keyboard shortcuts', 'Node-RED website', and 'v0.20.7'. A red callout box labeled 'Nodes' points to the 'link' node in the 'input' section of the nodes palette.

Node-RED

Flow 1 Flow 2

Flow Tabs

Nodes

Flow Area

Deploy button

Deploy

View

Import

Export

Search flows

Configuration nodes

Flows

Subflows

Manage palette

Settings

Keyboard shortcuts

Node-RED website

v0.20.7

Home

The Hello-World in Node-RED

- Open a terminal and type: **node-red**
- Open the browser and go to <http://localhost:1880>

Now you are in Node-RED!

- Insert an **inject** node, a **function** node and a **debug** node. Wire them together.



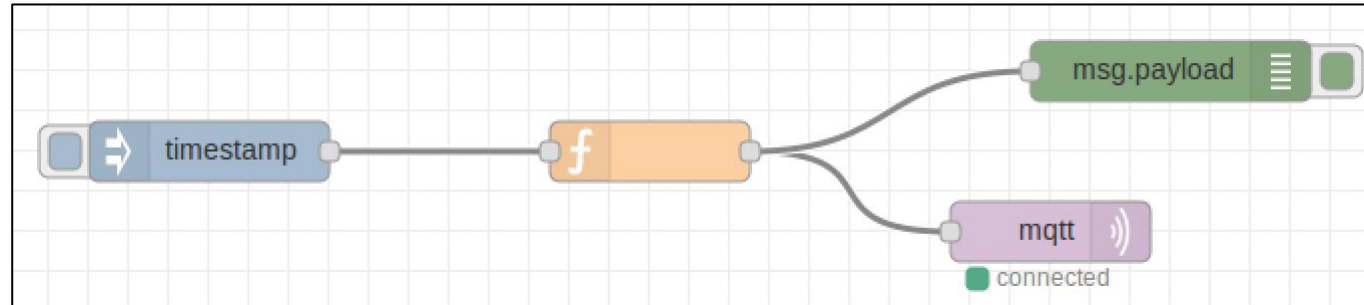
- Modify the **function** block as follows:
`msg.payload = "Hello World!";`
`return msg;`
- Click on **Deploy** and then on the inject button



A Working Example

- Let's create a flow in Node-RED that:
 - Generates random data periodically
 - Upload the random data on the ThingSpeak channel created before
- We need:
 - Inject node
 - Function node
 - MQTT publish node (remember to set user authentication)
 - Debug node

Random Data Generation



- Modify the function block as it follows

```
var CHANNEL_ID = "XXXXXX";
var field1=Math.random()*10;
var field2=Math.random()*10;
msg.topic = 'channels/'+CHANNEL_ID+'/publish/';
msg.payload='field1='+field1+'&field2='+field2+'&status=MQTTPUBLISH;
return msg;
```
- Modify the inject block to run periodically every 30 seconds
- What happens on your ThingSpeak page?

Variable types

- **Local variables:** Only exist in one execution of the single block
Input and Output are local
- **Context variables:** Exist in all the executions of the same block
Accessible with the ***context*** variable in that block
- **Flow variables:** Exist in all the flow, they are global in the flow
Accessible with the ***flow*** variable
- **Global variables:** Exist in all nodes from all flows!
Accessible with the **global** variable

E.g. `global.set("name","Fabio")`

You can access to the variable with `global.get("name")` anywhere

Memory Usage Example - 1

- Create another flow in Node-Red
- Import the code from examples (node-red-exec-thingspeak)
- Create a new channel on Thingspeak called *Memory Usage*, with one field
- Modify the block “*Thingspeak Message Configuration*” with your API_KEY.
- ADJUST THE MQTT auth (user,password,id)!!

Memory Usage Example - 2

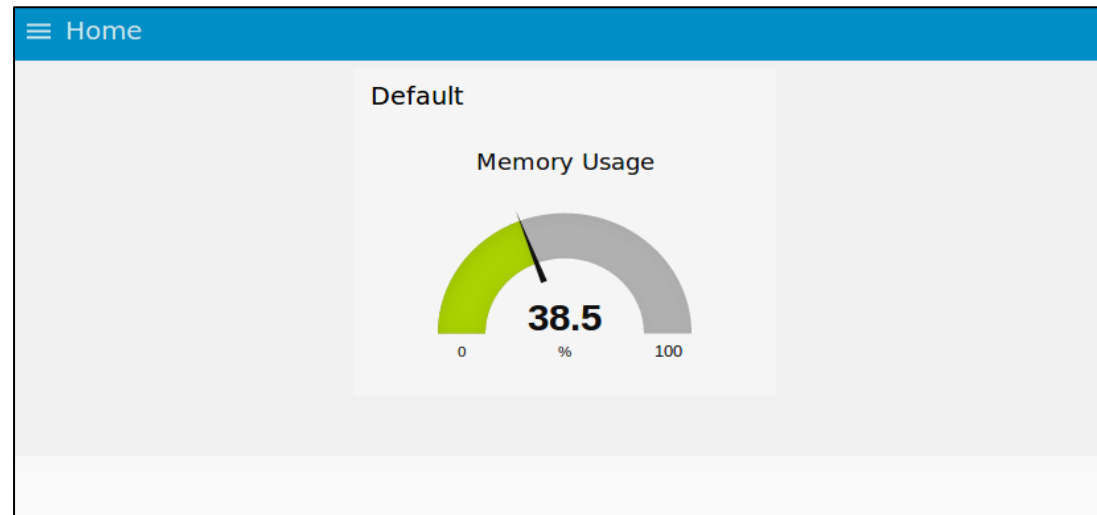
- This program will simply calculate the used memory percentage of the machine is running node-red and push the data to Thingspeak.
- It will rate-limit the message, only 2 message/minute will be sent to thingspeak

Memory Usage Example - 3

- Select the gauge node:
- Select the chart node:
- Connect them to the flow (to upper function block)



It will display a gauge and a chart that represent the used memory (available at <http://localhost:1880/ui>)

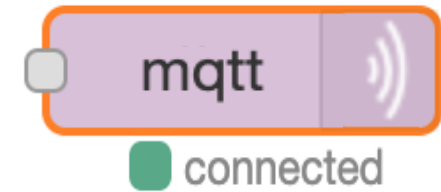


Processing Data (exercise)

- Create another flow in Node-Red
- Import the **node-red-alert_template** flow
- Modify the **CreateHttpRequest** block to read the last 10 values from your ThingSpeak feed.
- Modify the **ComputeAverage** block to compute the average of the 10 samples. Send an alert email if the average is greater than a fixed threshold.

Node-Red to ThingSpeak with MQTT (PUB)

- Add the MQTT publisher node and the inject
- **Server:** **mqtt3.thingspeak.com**
- **Client-id:** <MQTT_ID>
- **Topic:** channels/<channelID>/publish
- **Payload:** field1=XX&field2=YY&status=MQTTPUBLISH
- **Username:** <MQTT_USER>
- **Password:** <MQTT_PSW>



MQTT API here:

https://it.mathworks.com/help/thingspeak/mqtt-api.html?s_tid=CRUX_lftnav

Node-Red from ThingSpeak with MQTT (SUB)

- Add the MQTT subscriber node and the debug
- **Server:** **mqtt3.thingspeak.com**
- **Client-id:** <MQTT_ID>
- **Topic:** channels/<channelID>/subscribe/fields/+
- **Username:** <MQTT_USER>
- **Password:** <MQTT_PSW>



MQTT API here:

https://it.mathworks.com/help/thingspeak/mqtt-api.html?s_tid=CRUX_lftnav

Node-Red Summary

- Hello world + Functions
- Debugger
- Inject number + random
- thingspeak HTTP/MQTT
- memory usage (exec + delay)
- Dashboard

Challenge 2

40

Challenge 1

Challenge 1



People who
don't know



People
who know