

IoT laboratory 2.

In the IoT laboratory 1. sensors and actuator were connected to the Internet in order to display the data and control the devices. This laboratory now the IoT platform will be enhanced with data analytics and visualization.

The laboratory work is performed on the ThingSpeak platform. The ThingSpeak has the right tool for data storage, analytics and visualization and it also has API for sensor and application connections. The platform can be accessed via www.thingspeak.com

The measurements in the laboratory work are organized around different elements of the platform. The task of the student is to work with the measurements and report their results. The report should include the commands (program code) of the measurements, the configurations files and screenshots from the results. Basically the report should include everything to make the measurements and their results easily reproducible.

(This measurement work requires programming skills, however the skeletons of the code are usually provided, so the task should not be too hard.)

1. Introduction to ThingSpeak platform

The platform can be accessed via the www.thingspeak.com address. Students have to register to the service, which is free. There are some limitations on the free account, however the laboratory work does not require more than the free plan can offer. ThingSpeak organizes the incoming and outgoing data into channels. (In the previous IoT measurements we had MQTT channels also.) The channels can be further divided into fields. These fields provide the data for analytics, visualization and interactions. The data on the channels can be shared among users, there are private and public channels. Private channels can be shared as well, as there are separate read and write keys for the channel data. In order to access a channel, the user has to know the number of the channel and the right key for the channel (public channels do not require keys). Public channels can be searched as well. There are tags that help in the channel search.

Measurement task 1.

1. Register to the ThingSpeak platform.
2. Create a channel and publish some data using the MQTT method
3. Search for a channel, where fresh weather data can be accessed. The weather channel should contain more fields. It must contain a field for temperature, either in Celsius or Fahrenheit.

There are no further help for this task, as it should be simple and straightforward. The MQTT API has a documentation under the link:

<https://www.mathworks.com/help/thingspeak/mqtt-api.html> In order to search channels, the Channels/Public Channels menu item can be used. The search is using tags or username.

2. Analytics

In the platform we can perform analytics on the collected data. The analytics is provided by MathWorks using the MatLab language. The results of the analytics can be displayed on screen or they can feed an other channel.

Measurement task 2.

1. Change the previously found weather channel's temperature data from Fahrenheit to Celsius (or vice versa)
2. Create a channel which converts the other channel's temperature data from Fahrenheit to Celsius (or vica vera)
3. Collect data from a web page. Read data from a Hungarian weather info page and feed it to a channel.

(Sorry, the detailed measurement instructions are in Hungarian at the moment. Ask their help!)

3. Visualization

The platform supports two kinds of visualization. You can choose the Matlab visualization or there is a possibility to use any HTML based widget. A task is to use the first and later the second method for visualization.

Measurement task 3.

1. Based on the ThingSpeak example, display a temperature histogram on the data of the weather channel
2. Create a code that displays the weather channel data in a hourly average format for the last 24 hours
3. Create a code, which compares on a graph the last 7 days and the previous 7 days daily average temperature
4. Create an HTML/Javascript widget, using the example, for a Hungarian weather channel displaying the current temperature and humidity

(Sorry, the detailed measurement instructions are in Hungarian at the moment. Ask their help!)

4. Act, based on the data

The platform provide reaction triggers during the analytics. Reactions include Twitter messaging and sending an almost arbitrary constructed HTTP message, called ThingHTTP. Current task uses the latter one. The task is to analyze the data and in a certain condition, send a message to the student's mobile phone (or browser).

The PushBullet service connects mobile phones and browser applications simply in order to receive messages. The sent messages are reaching the user on all platforms. The service is free, but a google or a facebook account is required. The service can be accessed on the page: www.pushbullet.com

The PushBullet service is able to receive HTTP messages, and this task is using the ThingHTTP to send this messages.

Measurement task 4.

1. Register on the PushBullet service
2. Connect your own smartphone and/or browser. Install the necessary apps or browser plugins
3. On the ThingSpeak platform, create a ThingHTTP command, which analyzes the Hungarian weather channel info and periodically sends the temperature data to the smartphone using the PushBullet service
4. Analyze the Hungarian weather channel and create a trigger event when a new temperature value is appearing on the channel

(Sorry, the detailed measurement instructions are in Hungarian at the moment. Ask their help!)

5. The end of the measurements

Check whether the Hungarian weather channel works as it should be. As data is coming usually on a 5-10 minutes interval, the channel should already contain a fair amount of data.

Make the created channels public! Create tags on the channel with 'iot2' and 'meres' labels. Put the number of the channels into the report. Keep the channels till the end of the semester, as they could be checked later.

