Internet of Things

Challenge 02.

***Student**: Marco Bendinelli* ***ID**: 10673478*

***Required Software**: Node-RED, ThingSpeak*

Node-RED implementation

Node *file:*

it reads the CSV input file, and it outputs an object containing as payload a single utf8 string. Such property contains the values of each row separated by commas.

Node *csv:*

it converts the string contained in the payload into an array of objects:
"payload":[{"code":0, "field1":24, …, "field7":2}, {"code":1, "field1":24.1, …, "field7":2}, …,])

Function *filterAndConstruct:*

by using the filter method provided by Javascript, it creates a new array with the feeds that satisfy the condition "feed.code >= 3478 && feed.code < 3478 + 100".
For each element of the new array, the function *constructObjects* creates two objects (publishObj, field5Obj), and it adds them in two arrays (msgsToPublish, field5ToShow).

- The *publishObj* object contains the correct topic and payload to permit the publication of the values of the field 1, 2 and 5 of the current feed, to the ThingSpeak's MQTT device.
- The *field5Obj* object has as a topic the string "RSSI" and as a payload the value of the field 5.

The function has two outputs: the first array is sent on the first output, the second array is sent on the second one. In this way the function returns 100 objects on each output. In particular, the first output is directed towards the *MQTT* node, the second one towards the *chart* node.

Node *MQTT* (+ *delay* node) and node *chart* (+ *delay* node):

The two nodes take as input the messages coming from the function. At first, the messages are delayed to be properly processed. The *MQTT* node publishes to the ThingSpeak's MQTT device and the *chart* node adds the input values into the chart.

Thingspeak's channel:

Link: https://thingspeak.com/channels/1711441 Channel ID: 1711441