Home Challenge #4 Internet of Things 2019/20

Armenante Valerio 10691549

Di Salvo Dario 10687968

GitHub link: *https://github.com/valearm/IoT*

ThingSpeak link: *https://thingspeak.com/channels/1065976*

First of all, we have used an “Inject node” in order to start the computation of our Node-RED program. Then we put a “File” block called “Read CSV File” in order to import our data from “traffic.csv” file located in the path “/home/user/Desktop/Submission4/traffic.csv”. After that block we put a “Debug” to check the correctness of our output from the previous block. The “Publish message filter” block have been used to filter only the “Publish” type message, then we put a double condition on the two topic we was interested in “factory/department1/section1/plc” and “factory/department3/section3/plc” to update the field1, “factory/department1/section1/hydraulic_valve” and “factory/department3/section3/hydraulic_valve” to update the field2. In this point of our scheme there are the same blocks in parallel to handle the different “values” for each field. We notice that there are some messages from the MQTT broker to the publishers related to the topics, but we decided to consider them too. The “csv formatting object” block takes in input a Javascript object and using the space as separator, we obtain an output formatted in CSV in order to split data, using “split” block. After that we put a “Switch” block called “Regex condition” to filter message payload. We used the “Function” block “Hex to Dec” to convert our hexadecimal payload into normal string payload. The “Json” block is used to format the output string of the previous “function” block to a json file, and the function block “Accessing value field1” and “Accessing value field2” to access the two different “values” type of message payload by means of assignments “msg.payload = msg.payload.value; msg.topic= 'field#' ” and as usual we put a debug to verify it (where # is specified for each condition, and it is 1 or 2).In order to update at the same time the two fields on our ThingSpeak, the final MQTT block must receive as input “field1=#&field2=#” (where at usual # are the found values), so for this reason we have used two “template” blocks and a “join” block to reach our goal. We used two “delay” blocks in rate limit mode to obtain the wait one minutes between two consecutives entries. At the end we set the “MQTT” block called “ThingsMQTT” inserting as topic “channels/1065976/publish/MLNYHZWWBZNAY83A” where we specified our Channel ID and our API key, also we specified the broker server as “mqtt.thingspeak.com:1883”. Below there are screenshot of our entries on ThingSpeak.