POLITECNICO MILANO 1863

ANTLAB

IoT Challenge #3

Node-Red

+

LoRaWAN

Challenge 3: Node-Red and LoRaWAN

Part 1: Challenge

Node-Red

Part 2: Exercise

LoRaWAN

Also fill the form for delivery

PART 2 - Challenge

Part 1: Challenge – Node Red

- Download the **challenge3.csv** file here
  or from WeBeep

- Process the CSV file in Node-Red

```
"No.","Time","Source","Destination","Protocol","Length","Source Port","Destination Port","Info","Message"
"44","1.631805505","10.0.2.15","3.65.137.17","MQTT","79","34039","1883","Connect Command",""
"48","1.633054189","10.0.2.15","3.65.137.17","MQTT","85","47723","1883","Connect Command",""
"53","1.656703356","3.65.137.17","10.0.2.15","MQTT","62","1883","34039","Connect Ack",""
"57","1.666550621","10.0.2.15","91.121.93.94","MQTT","80","43133","1883","Connect Command",""
"59","1.673414543","3.65.137.17","10.0.2.15","MQTT","67","1883","47723","Connect Ack",""
"61","1.717782327","91.121.93.94","10.0.2.15","MQTT","62","1883","43133","Connect Ack",""
"64","2.634685221","10.0.2.15","3.65.137.17","MQTT","91","34039","1883","Subscribe Request (id=1) [metaverse/building3/section2]",
"66","2.635737036","10.0.2.15","3.65.137.17","MQTT","82","47723","1883","Subscribe Request (id=1) [university/+/area0]",""
"68","2.656128517","3.65.137.17","10.0.2.15","MQTT","62","1883","34039","Subscribe Ack (id=1)",""
"70","2.664970959","3.65.137.17","10.0.2.15","MQTT","62","1883","47723","Subscribe Ack (id=1)",""
```

Packet Number!

ANTLAB

Challenge: What to do? (1)

- Create a flow to periodically publish MQTT messages to the local mosquitto broker **(localhost, port 1884)**, to the topic ***challenge3/id_generator***
  (be sure to start the mosquitto broker locally with the correct port)

- Messages should be sent with a rate of **1 message every 5 seconds.**
- Each message should contain in the payload a string of JSON format with a **random** number (**id**) between **0 and 30000**, and the time in which the msg is generated (**UNIX timestamp**)

Message payload example: {"id": 7781, "timestamp":1710930219}

When sending the message, also save its field in a CSV (***id_log.csv***) with the form:

No.,ID,TIMESTAMP where No. is the row number (incremental)

Include this CSV in your delivery

What to do? (2)

In another branch of the flow (same flow):

- Subscribe to the topic ***challenge3/id_generator*** in the local broker **(localhost, port 1884)**

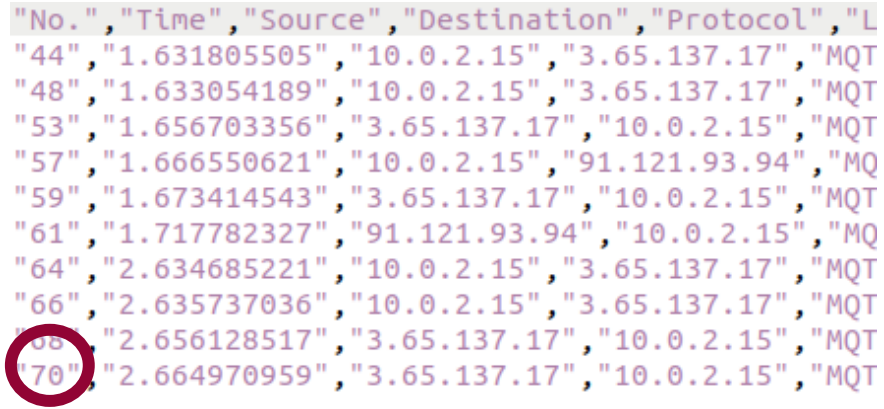
- After receiving a message from the subscription, take the ID and compute the **remainder of the division by 7711** to get **N**:

  N = ID modulo 7711

- At every message you receive, process the **challenge3.csv** file and take the message with frame number equal to the received identifier **N**

e.g. id=7781

N = 7781 modulo 7711 = **70** →



"No."	"Time"	"Source"	"Destination"	"Protocol"	"L
"44"	"1.631805505"	"10.0.2.15"	"3.65.137.17"	"MQT	
"48"	"1.633054189"	"10.0.2.15"	"3.65.137.17"	"MQT	
"53"	"1.656703356"	"3.65.137.17"	"10.0.2.15"	"MQT	
"57"	"1.666550621"	"10.0.2.15"	"91.121.93.94"	"MQ	
"59"	"1.673414543"	"3.65.137.17"	"10.0.2.15"	"MQT	
"61"	"1.717782327"	"91.121.93.94"	"10.0.2.15"	"MQ	
"64"	"2.634685221"	"10.0.2.15"	"3.65.137.17"	"MQT	
"66"	"2.635737036"	"10.0.2.15"	"3.65.137.17"	"MQT	
"68"	"2.656128517"	"3.65.137.17"	"10.0.2.15"	"MQT	
"70"	"2.664970959"	"3.65.137.17"	"10.0.2.15"	"MQT	

What to do? (3)

- If the message with **Frame No. = N** in the file contains an **MQTT Publish** then, send a publish message to the local broker to the same topic found in the **MQTT Publish**
  The message you publish should have as payload the following string:
  ‘{"timestamp":"CURRENT_TIMESTAMP","id":"SUB_ID","topic":"MQTT_PUBLISH_TOPIC", "payload":"MQTT_PUBLISH_PAYLOAD"}’
  Where:
  - CURRENT_TIMESTAMP **Current time** at the moment of the sending of the pub
  - SUB_ID Message **ID** received from the Subscription i.e. **7781**
  - MQTT_PUBLISH_TOPIC: Topic from the CSV of the Publish message with frame number **N**
  - MQTT_PUBLISH_PAYLOAD: Payload from the CSV of the Publish message with frame number **N**

- Limit the msg published in this step with a rate of **four messages per minute**
  (Use the rate limiter node)

What to do? (4)

- In addition, after publishing the publish message, if the Publish Message contains **in the payload a temperature in Fahrenheit** (check for the Type=Temperature and Unit=F attributes in the payload), take this message and plot its value in a Node-Red chart:

- For the Chart:
  - Take only publish messages having payload with temperature in **Fahrenheit**
  - Produce a **chart** in Node-Red plotting the temperature value**, taking the mean value in the "range" attribute in the paylaod as a number (min + max divided by two)**

At the same time, save the payload of these msgs (only those with Temp in Fahrenheit) in a CSV ***(filtered_pubs.csv)*** containing one msg Payload for each row:

filtered_publish.csv format:

No.,LONG, LAT, MEAN_VALUE, TYPE, UNIT, DESCRIPTION

where No. is row number (incremental)

Include this CSV in your delivery

Publish example (1)

No.,Time,Source,Destination,Protocol,Length,Source Port,Destination Port,Info,Payload

36,5.70180035,3.65.137.17,10.0.2.15,MQTT,322,1883,34039,"Publish Message [hospital/facility1], Publish Message [hospital/room1]","{""long"": 80, ""range"": [0, 59], ""lat"": 86, ""type"": ""temperature"", ""unit"": ""C"", ""description"": ""Room Temperature""},{""long"": 92, ""range"": [8, 37], ""lat"": 80, ""type"": ""temperature"", ""unit"": ""F"", ""description"": ""Room Temperature""}"

PUBLISH WARNING !!!

If packet contains multiple Publish, send them as separate publish messages (and plot in the chart separately if they match the filtering)

If some of the payload do not appear or is incomplete, consider it as empty payload

ANTLAB

Publish example (2)

No.,Time,Source,Destination,Protocol,Length,Source Port,Destination Port,Info,Payload

36,5.70180035,3.65.137.17,10.0.2.15,MQTT,322,1883,34039,"Publish Message [hospital/facility1], Publish Message [hospital/room1]","{""long"": 80, ""range"": [0, 59], ""lat"": 86, ""type"": ""temperature"", ""unit"": ""C"", ""description"": ""Room Temperature""},{""long"": 92, ""range"": [8, 37], ""lat"": 80, ""type"": ""temperature"", ""unit"": ""F"", ""description"": ""Room Temperature""}"

Parse the Payload in a JSON **when possible**.

We know, you have to deal with ""

Publish Message to topic: hospital/facility1


```
{
 "timestamp": "1712561821",
 "id": "7747",
 "payload":{"long":80,"range":[0,59],"lat":86,"type":"temperature","unit":"C","description":"Room Temperature"}
}
```


Publish Message to topic: hospital/room1


```
{
 "timestamp": "1712561821",
 "id": "7747",
 "payload": {"long": 92,"range": [8,37],"lat": 80,"type": "temperature","unit": "F","description": "Room Temperature"}
}
```


ANTLAB

What to do? (5)

- If the message with **Frame No. = N** instead contains an MQTT ACK message **(Publish Ack, Connect Ack, Sub/Unsub Ack)**, increment a global ACK counter, then save the message into a CSV file named "**ack_log.csv**" with the fields:

  No., TIMESTAMP, SUB_ID, MSG_TYPE **(No. is row number incremental)**

  Where:
  - TIMESTAMP is the current time when the msg is saved in the CSV
  - The SUB_ID is the Message **id** received from the Subscription i.e. **7781**
  - MSG_TYPE is the message type: **e.g Connect Ack**

  Include this CSV in your delivery

- After you find an ACK and you save it in the CSV: **SEND** the value of the global ACK counter to **your thingspeak channel**, passing in the **field1** of the channel the value of the global ACK counter. **SEND USING HTTP API**

  Include the channel link in the report and in the form, make it public!!

What to do? (6)

- In all the other cases (frame No = **N** not containing an ACK or a publish) → Ignore the message!

Program your flow to stop working after **receiving exactly 80 id** messages from the subscription:

do not process more than 80 ID messages

Discarded msgs (i.e., no pub or no ack) should be counted for the 80 messages limit

Recap on CSV File Formatting

The first row should include the header.
Report the various fields separated by a comma.

- **id_log.csv** format:

  No.,ID,TIMESTAMP

e.g. *1,7781,152952003*

- **filtered_publish.csv** format:

  No.,LONG, LAT, MEAN_VALUE, TYPE, UNIT, DESCRIPTION

e.g *1, 92, 80, 23, temperature, F, Room Temperature*

- **ack_log.csv** format:

  No., TIMESTAMP, SUB_ID, MSG_TYPE

e.g. *1, 1569239210,7781,Connect Ack*

PART 2 - Exercise

Exercise Questions (EQ1-EQ2)

EQ1) A LoRaWAN network in Europe (carrier frequency 868 MHz, bandwidth 125 kHz) is composed by one gateway and 50 sensor nodes. The sensor nodes transmit packet with **payload size of L byte** according to a Poisson process with intensity lambda = 1 packet / minute. **Find the biggest LoRa SF** for having a success rate of at least 70%. Hint: use https://www.thethingsnetwork.org/airtime-calculator to compute the airtime of a packet.

Report the result in the form!!

For the payload size L of your packet, take it as follows:

Take **XY** = Last two digit of your person code (leader code)

L = 3 + XY bytes

e.g. personcode = 10692**11** -> **XY** = 11 -> **L = 3 +11 = 14**

EQ2) You have purchased an Arduino MKR WAN 1310 and wish to create a system that reads temperature and humidity data from a DHT22 sensor and sends this data wirelessly to ThingSpeak over LoRaWAN. Design a complete system block diagram (sketch in Node-Red) and describe, in detail, the steps you would need to take to get the system fully operational.

Exercise Questions (EQ3)

EQ3)

Using the paper *"Do LoRa Low-Power Wide-Area Networks Scale?"* by M. Bor et al. and the LoRa simulator available at LoRaSim, your task is to reproduce Figure 5 and Figure 7 from the paper

Instructions:

1.Read the Paper

Carefully study the relevant sections of the paper to understand the experimental setup, parameters, and key findings, especially those associated with Figures 5 and 7.

2.Explore LoRaSim

Familiarize yourself with how the LoRaSim simulator works. Understand its configuration options and how to run experiments that model LoRa network behavior.

3.Reproduce the Figures

1. Use LoRaSim to replicate the simulations that produced **Figure 5** and **Figure 7**.
2. Ensure your simulation parameters (e.g., number of nodes, spreading factors, traffic load, transmission power, etc.) match those used in the original experiments as closely as possible.
3. Present your results in the same format as the original figures for easy comparison."

You can use as reference the Python Notebook you find in WeBeep (LoRasim.ipynb)

Challenge deliverables

PART 1 – Challenge:

- A **PDF** report **"Challenge.pdf"** containing the explanation of the Node-Red nodes. Include an image of **the Node-Red flow**, explain the **meaning of each node**. Report a picture of the obtained **node-red Chart!**
- **Node-Red** flow export as **JSON:** ***nodered.txt***
- **CSV files** produced: **id_log.csv**, **filtered_pubs.csv**, and **ack_log.csv**
- **Thingspeak channel ID** (make it public)

PART 2 – Exercise:

- A PDF **"Exercise.pdf"** containing the answers to EQ1-EQ2 and comments/figures for EQ3
- The python code used to replicate the two figures for the EQ3

FILL THE FORM!!!

Include YOUR NAMEs and PERSON CODEs in the two PDF Files

The files should be included in a ZIP (personcode1_personcode2.zip) and uploaded in WeBeep

Challenge delivery: HOW?

How to deliver?

- Upload the files in a zip archive as .zip file on the **folder Challenge #3** on WeBeep “Assignments” folder

- **Fill this form** with the csv values produced from Node-Red filtered messages

For 2-people teams:

- Choose your team leader and name the file as:
  <leader_personcode>_<other_personcode>.zip

- **Only the teamleader** should upload the challenge in WeBeep
  Do note upload the same challenge more times

- *Can I take the challenges with the other class students*?
  YES, **but** only the team leader should upload the challenge in WeBeep

Delivery Deadline

- **STRICT** Deadline:

  April 27, 2025 h 23.59

!!! BAD DELIVERY -> penalty points !!!

Example: bad names, no form, wrong file extensions

- Max 2 people

GOOD LUCK!

ANTLAB