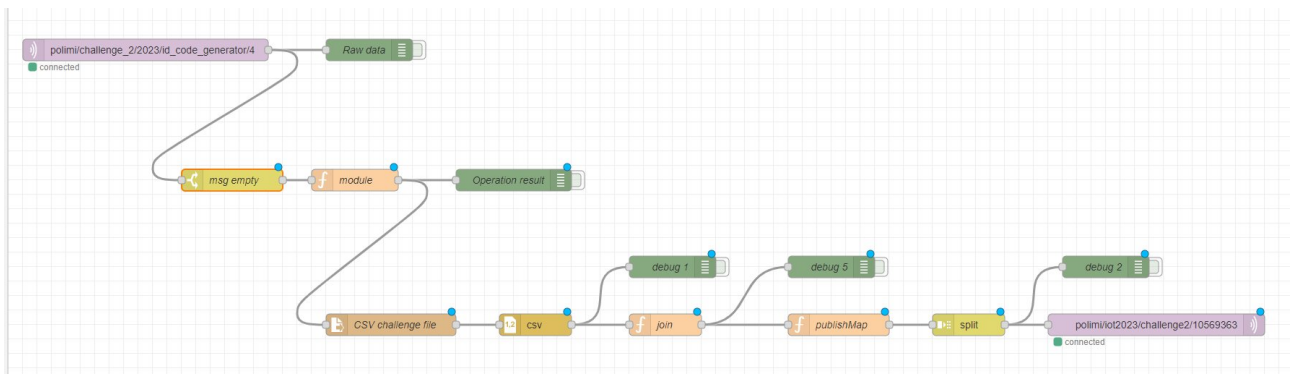


IoT 2023 CHALLENGE 2

- (1) Name: Alessandrini Luca Person Code: 10569363
- (2) Name: Renzi Flavio Person Code: 10659808

Re-Publish flow

Image of the flow (screenshot from Node-RED):



Node explanation (except for the debug nodes, which were needed only in debug phase and are not necessary for the workflow):

- `polimi/challenge_2/2023/id_code_generator/4`: this node subscribes to the mqtt topic. We used the `/4` because the someone were spamming messages on the others.
- `msg empty`: switch node to check if the incoming payload is not empty. If it is not, the flow continues.
- `module`: it first checks that the id is well formatted and then performs the function written on the delivery
- `CSV challenge file`: it opens the given CSV file. N.B.: to replicate our results, please change the path of the file.
- `CSV`: it parses the CSV file.
- `join`: it extracts the correct packet from the CSV using `myid` (the value computed in the `module` function).
- `publishMap`: it splits the field `info` of the messages using `" , "`. For each element, if they contain a publish message, then we search that publish message in the `Message` part, and we check that it is well formatted and it creates the message to be sent with timestamps, ids, and payload as string. If the packet contains multiple messages, we add them to an array as separate elements, in order to be sent separately later in the flow.

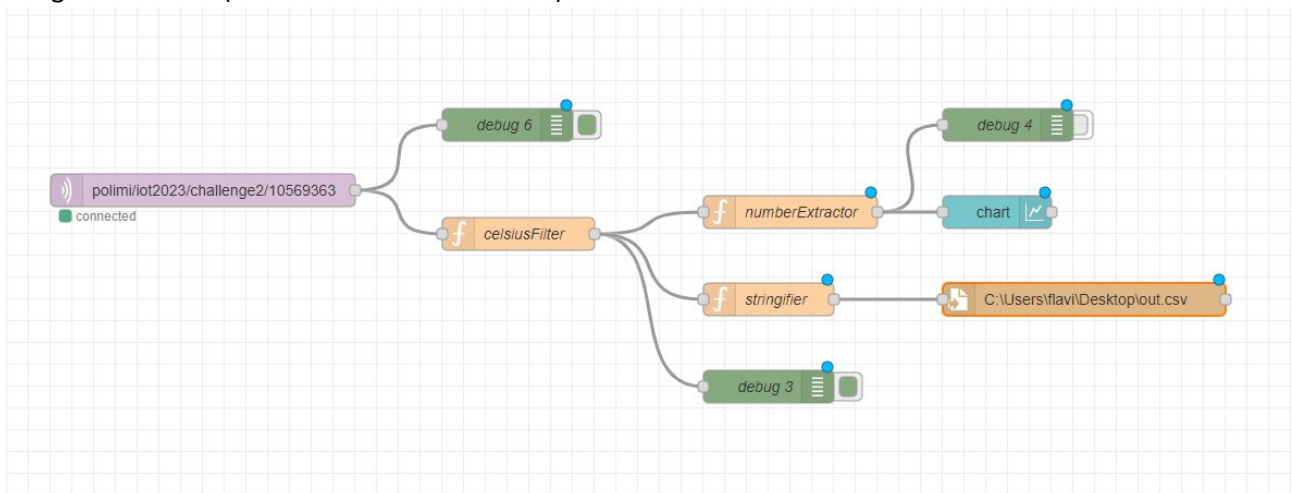
It may happens that some packets are not well formatted in the provided CSV, such as no. 2950. In this case, the firsts publish messages will be added to an array correctly, the broken one will be added as it is (broken), and the subsequent ones with an empty string as payload. If instead the publish message does not have a field `Message`, and empty string is set as payload.

- `split`: it takes each element of the array provided by the previous node, and it forwards it as a single message.

- `polimi/iot2023/challenge2/10569363`: it publishes the received message on the topic related to the person code.

Filter Flow

Image of the flow (screenshot from Node-RED):



Node explanation (except for the debug nodes, which were needed only in debug phase and are not necessary for the workflow):

- `polimi/iot2023/challenge2/10569363`: this node is the responsible for the subscription to the topic on which the last node of the Re-Publish flow publishes.
- `celsiusFilter`: it forwards the message if contains a payload well formatted with "C" as unit.
- `numberExtractor` + `chart`: it extracts the number of the range to be plotted in the `chart` node.
- `stringifier`: it converts the json into a string in order to be added to the output CSV
- `C:\Users\flavi\Desktop\out.csv`: writes to file. Be sure to change the path if you want to replicate our results. It is formatted as follows: one string per each row representing the payload of the message received.

Plot

