Elias Borge-Ilseth, 11075526
Eskil Myklatun Østbø, 11074713

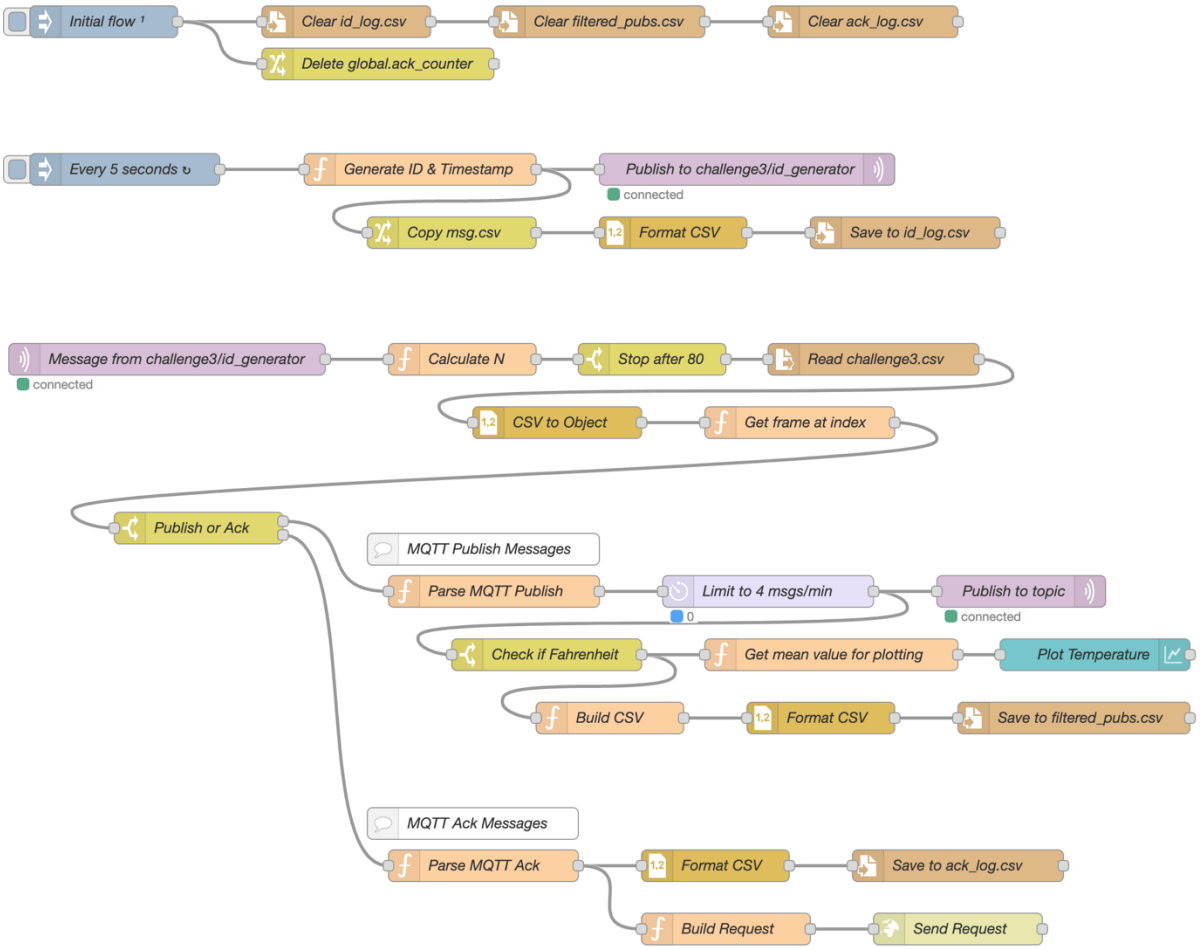
2727.04.2025

Internet of Things Challenge 3

Part 1 – Node Red

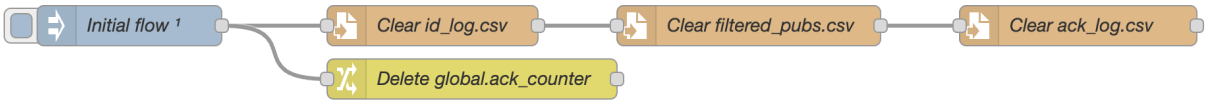
ThingSpeak channel ID: 2924784
URL: https://thingspeak.mathworks.com/channels/2924784/

This document describes each section and node of the flow requested in the challenge. The flow automates the generation, processing, and logging of messages within a deployment.
The full flow is seen below with snippets of each section included in their discussion:



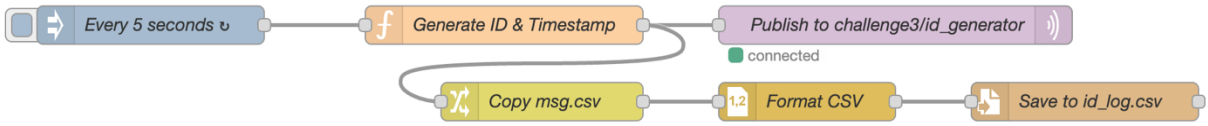
Elias Borge-Ilseth, 11075526
Eskil Myklatun Østbø, 11074713 2727.04.2025

Initialization



The Initial flow is triggered immediately after a new deployment starts. It clears the contents of three CSV files and deletes the global acknowledgment counter, ensuring a clean state for the new deployment.

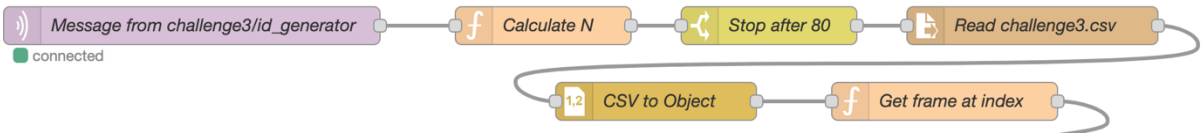
Message Generation



Every five seconds, a new ID and corresponding timestamp are generated. The function node creates a random ID, stores it along with the timestamp in the message payload, and saves these values – together with the current count – in a separate CSV property within the message.

The payload is then published to the broker channel. Simultaneously, the CSV object is copied to the payload, formatted as a CSV string, and stored in the output file.

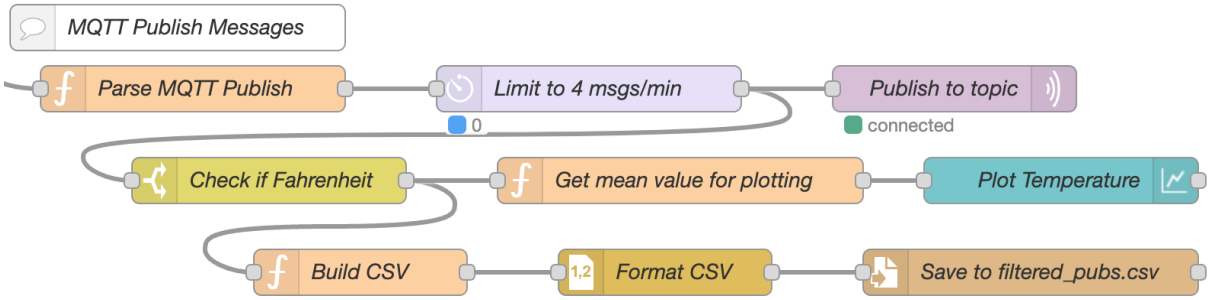
Message Reception and Processing



The client listens for messages on the channel used to send generated IDs. When a message arrives, it calculates the value N – as described in the task –and increments a counter. The counter tracks how many messages have been received, and the program continues until 80 messages have been processed. Afterward, the challenge data is read from the file, converted into an object, and the correct frame at index N is retrieved.

From this point, the flow takes one of two paths depending on if the frame is a Publish message or an Ack message.

Handle Publish messages



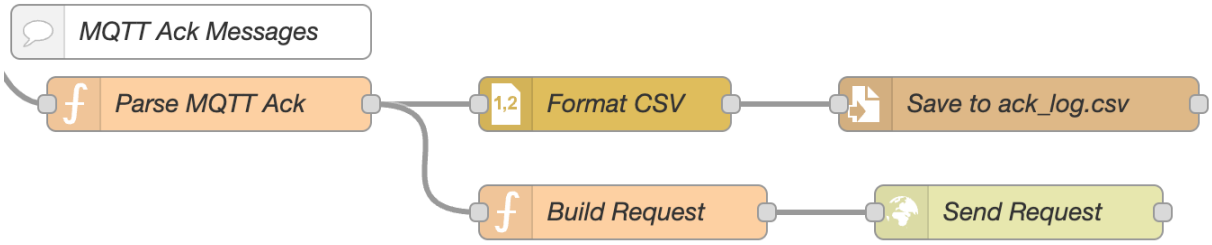
This section begins by parsing the frames into JavaScript objects. Since these are Publish messages, they include payloads and topics to which they are sent. The process involves two key steps:

1. Extracting topics using regex to identify target destinations.
2. Parsing the payload from JSON into structured data.

With the parsed topics and payloads, the system combines items at corresponding indices to build a list of messages. These messages enter a rate-limiting queue, where they await release. Once released, they are published to their designated topics and checked to determine if they contain temperature values in Fahrenheit.

For messages with Fahrenheit values, the system calculates the average of the measured values, plots this average for visualization, and prepares the data as CSV and store it in the correct file.

Handle Ack messages



The acknowledgment messages are parsed in a manner like the publish messages. The process begins by incrementing a counter and extracting the correct acknowledgment type, with any extra data in brackets removed. Next, the payload is constructed, converted into CSV format, and stored. After updating the counter, the system sends an update to ThingSpeak with the current total number of acknowledgments received.

Results

After one deployment of the flow a total of 80 generated ID messages has been received and analysed. The list of generated IDs can be found in the file named ‘id_log.csv’. Similarly, the recorded temperatures and acknowledgment messages can be found in the files ‘filtered_pubs.csv’ and ‘ack_logs.csv’, respectively.

Below we can also see the plot of the recorded temperatures.