



CAN to Middleware Connectivity Bridge

Batool M. Khader, Xinchun Hu, Judith Hernandez-Campillo, Haotian Wang

Advisor: Dr. John Kelly

This project is provided and sponsored by John Deere, Turf & Utility Care, Raleigh, NC

Electrical and Computer
Engineering Department

DESIGN PROBLEM

Problem Statement

For Deere machinery, CAN-bus has been doing a great job at controlling vehicle's communication, however, a middleware bridge is needed to provide high-level and high-speed service to enable functionality outside of the vehicle such as robotics, cloud computing, and autonomy.

Design Objective

A software system that can communicate between CAN-bus and MQTT protocols.

Design Criteria/Engineering Requirements

- The system shall be able to interact with any CAN-bus system with configuration of the system itself.
- The system shall be able to send specific CAN messages to specific MQTT topics, and according to certain thresholds and timings.

DESIGN ARCHITECTURE

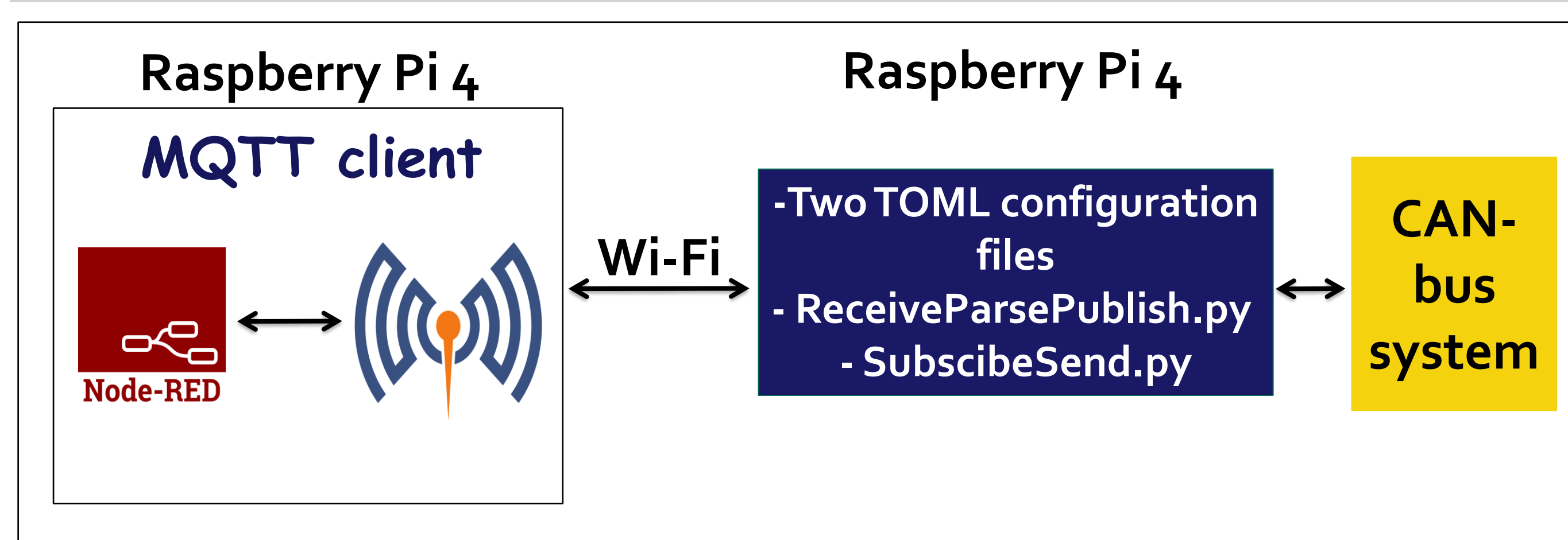


Figure 1. Software system architecture for CAN-MQTT connectivity bridge

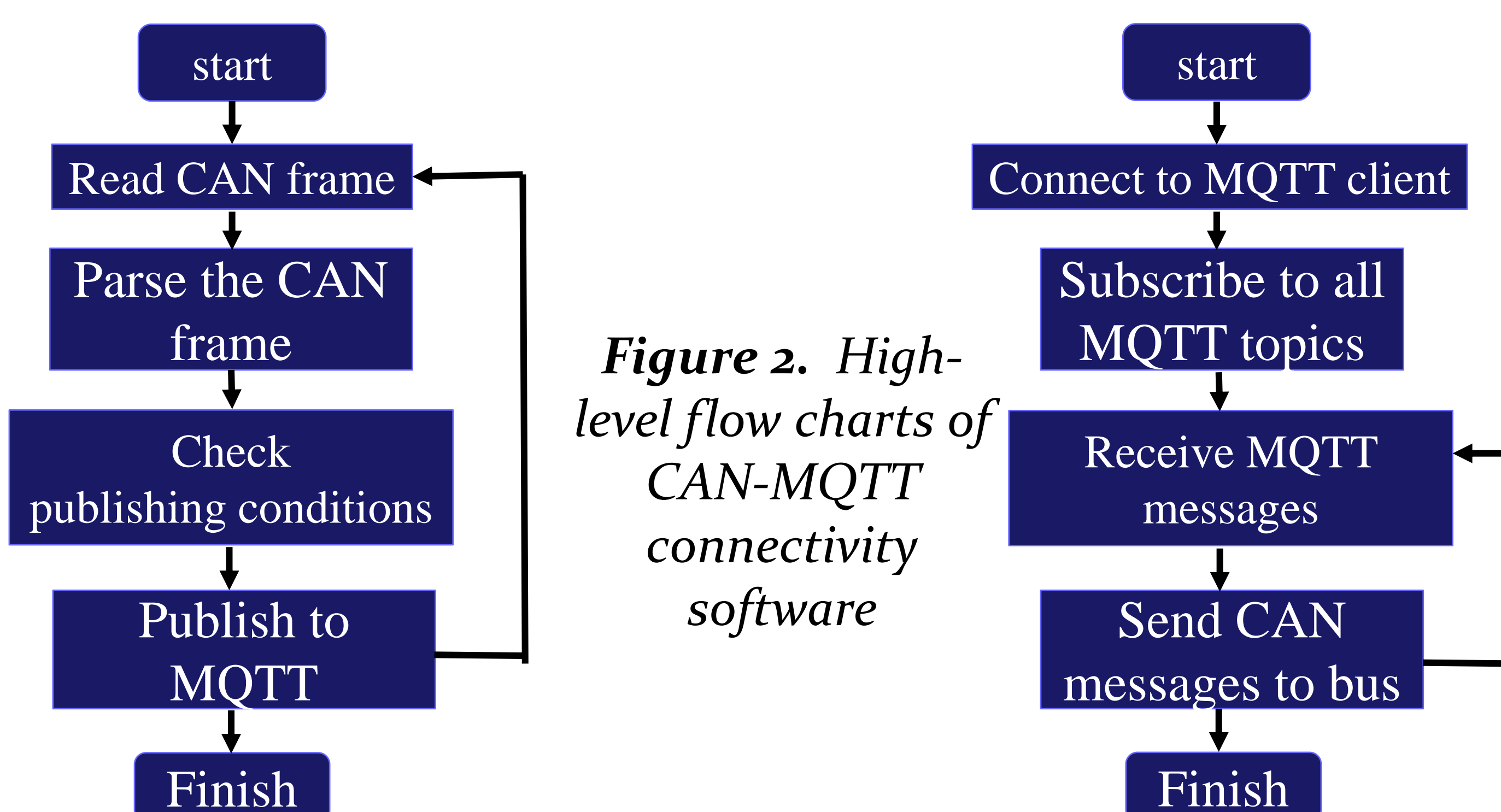


Figure 2. High-level flow charts of CAN-MQTT connectivity software

DESIGN IMPLEMENTATION

- Initially, the connectivity bridge was built using ESP8266 Wi-Fi module, however, it did not enable the system to meet the requirements.
- We built the software system on Raspberry Pi using python, and we implemented it first on virtual CAN data.
- To test and implement the system. we built a real CAN system.

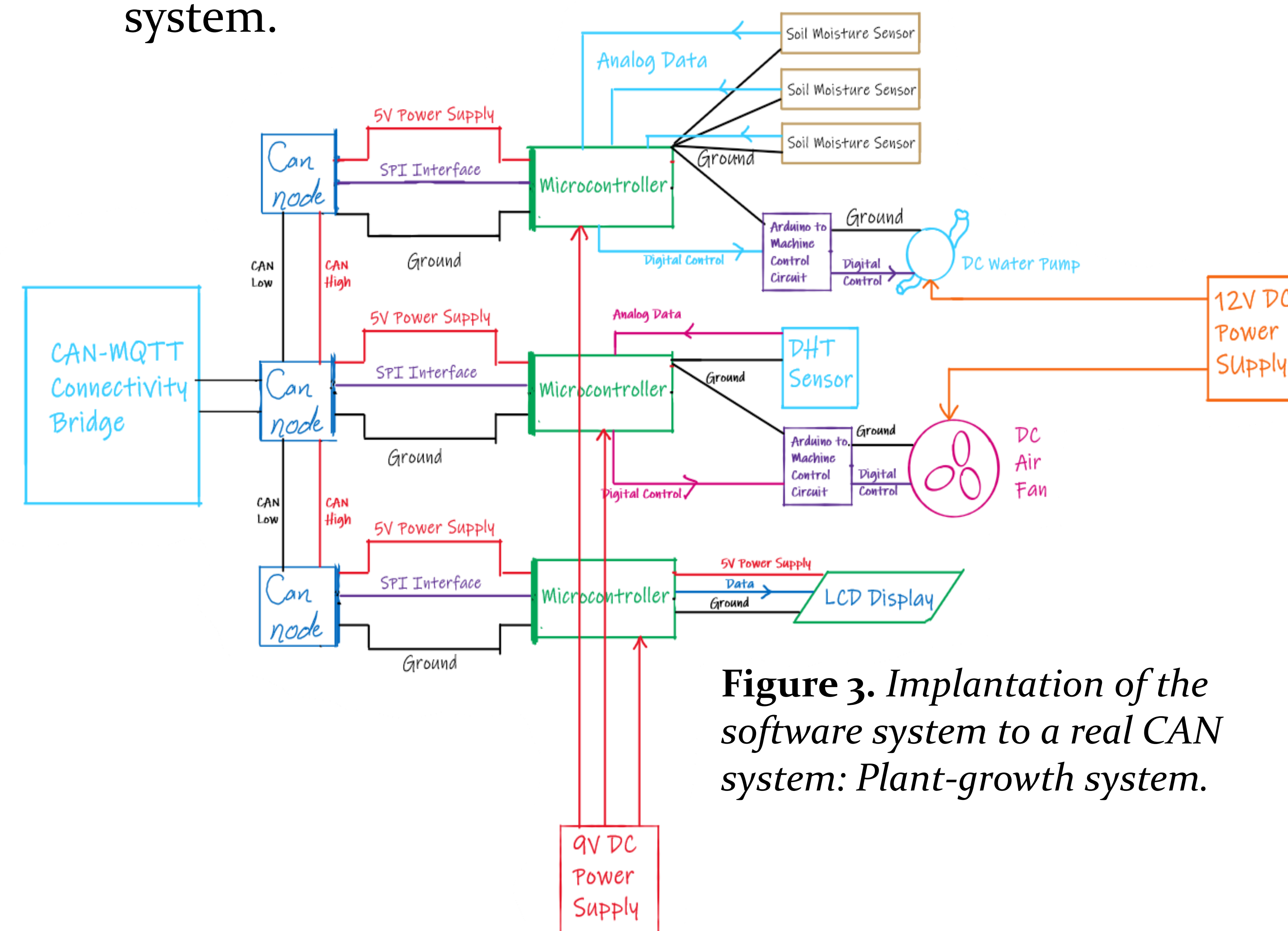


Figure 3. Implantation of the software system to a real CAN system: Plant-growth system.

RESULTS



Figure 4. Results for publishing CAN messages to MQTT client on Node-Red for both virtual and real CAN systems

Final Project Prototype

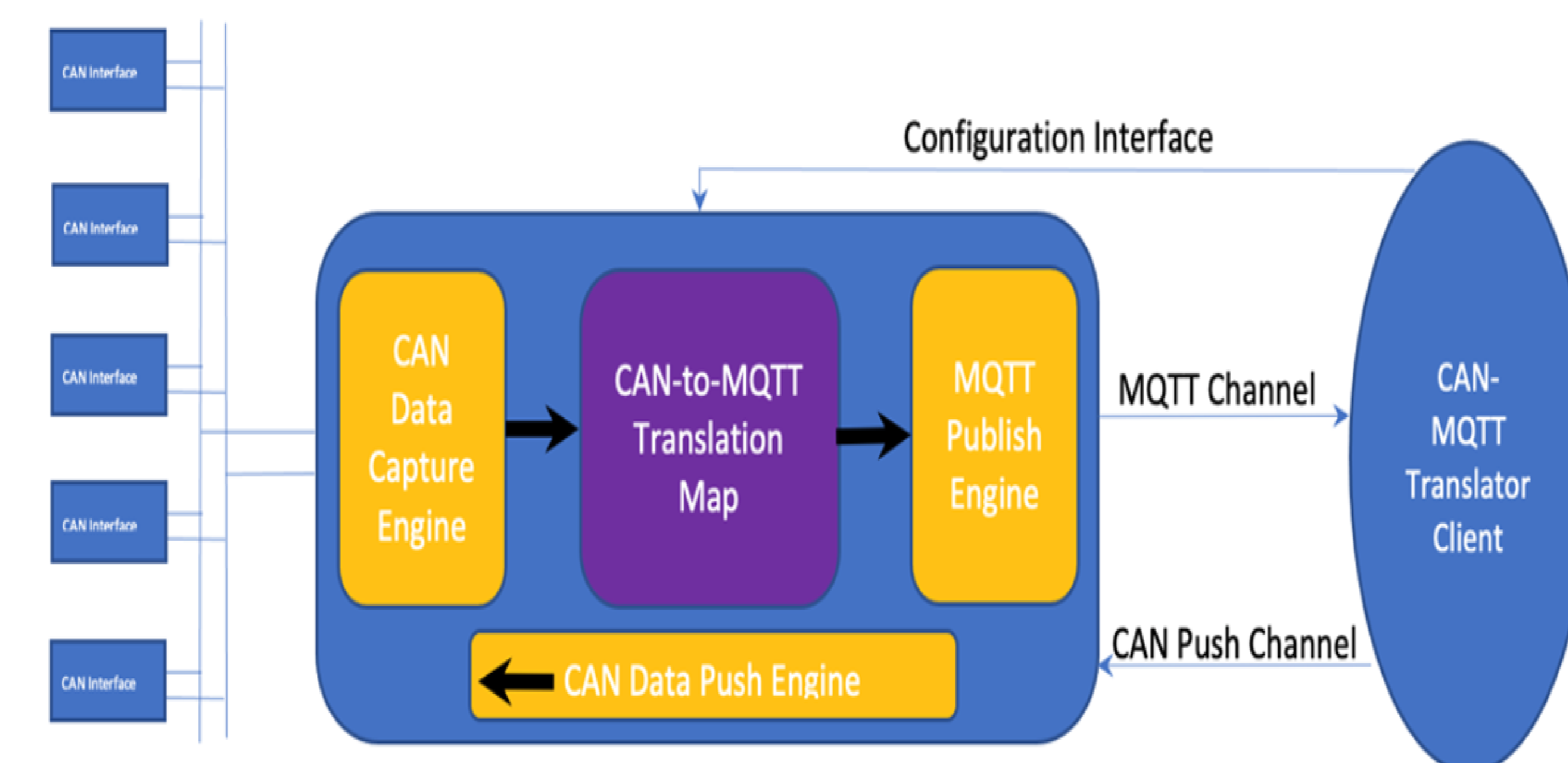


Figure 5. Final CAN-MQTT bridge prototype

CONCLUSION

- These results suggest that it is possible to link the two protocols together and meet Deere requirements successfully.
- The user of this system does not need to hand-code the process for every CAN bus system that this software would be applied to.
- This system can provide feedback to the user about their CAN bus system in a convenient way that saves time. Also, it would allow functionality outside of the CAN bus system such as data analytics, robotics, and cloud-computing.

FUTURE WORK

- Future development should focus on creating robust data visualization and analysis along on the MQTT side of the system as an alternative to Node-RED.
- Use CAN data to monitor and predict the health of the CAN-bus system.