UFSC 2022 - Embedded Systems Programming in C++

Telemetry Central Unit

Gabriel Goulart Müller[1]

[1]UFSC

December 18, 2022

Abstract

This project proposes a system to communicate data from battery management system of Vento Sul boat, from sea to devices anywhere. To reach this objective, the system uses Can protocol to communicate with battery management system and MQTT protocol to publish messages on internet. The whole system is designed in C++, using includes like paho mqtt from Eclipse Foundation and CAN librarys from Arduino.

Keywords
IoT, Cpp, MQTT, Arduino, ESP8266

1 Introduction

The team Vento Sul from UFSC is one of the most competitive teams in Brazil on Desafio Solar Brasil (DSB) and the one five-time champion on the challenge. This boat solar challenge is like many others around the world, we have monohulls and catamarans and within a week we perform several tests, as sprint, slalom and sprint-slalom. There is only one test by a day and it lasts between a half and 6 hours. The shorter test is a sprint, like a speed test and the harder is known as "long test", when boats performs like F1 cars, doing as many laps as possible in track, but on the ocean. In the last competitions, we have suffered with management of lithium battery, suffering faults on many tests. The battery management system(BMS) is complicated to understand and manage. By the way, the BMS is essential for the project to use lithium batteries, to keep them on safe operating area (SOA), on the other hand, we could use lead batteries, heavier and ineffective. In this dilemma, we choose the tougher way. Our BMS was provided by Atlas Power, a brazillian startup who made solutions for solar and engineer projects. This device manages the lithium battery and monitors it's temperatures, voltages, for each cell, and currents, flowing through and out the pack, while the core of calculates with this parameters power, pack voltage, state of charge and mean temperature. If the parameters are out the safe operating area, the BMS has an actuator to cut off the battery from the eletric system, preventing a disaster. The BMS communicate its data on Can network, where any device could access the parameters of battery anytime. Here is the point, if we could connect a Telemetry Central Unit (TCU), we could communicate the lithium data from boat with the world. Thus, the project of TCU consists in integrate the BMS with smartphones and computers through the internet.

2 Materials & Methods

First, we need an platform to embed on boat and who speaks CAN. Therefore, we'll using MCP2515 to connect on CAN and we could use any Texas Instruments(TI) DSP, Arduino or ESP.

The next step was choose our DSP, despite the robustness of TI, we decided for ESP8266 because it has many readable documentation and other similar projects to facilitate the research and make the project ready to DSB on time.

Using ESP8266 we could have our entire project on C++, even the embedded software and smartphone application. Futhermore, ESP8266 has Wi-fi hardware and software ready to put our device online. Besides, if we don't use Wi-fi or bluetooth, using LTE/HPSA/GSM the project ends up having extra monthly costs with network services.

For finish the project, we need an HTTP protocol or websockets or MQTT and a server. Between the options, the MQTT is the most indicated for IoT Projects. The choice for server was HiveMQ, because of its suport for Arduino IDE and its Free clusters. On the other hand, for

embedded software the Paho MQTT C++ from Eclipse was choose because of its robustness.

3 Proposed design

The Telemetry system has to publish BMS data on topic at HiveMQ Broker, that has to be online for smartphone and computer clients, that will subscribe on these topics and receive the messages with data from batteries analysis.

On pilot version, just messages with the pack voltage, state of charge and currents will be published on broker. Only to save payloads and to simplify the code. Otherwise, the classes for cell voltages and temperatures are written already. The explanation is cleared on next topic.

The code has one main class, CanMsg, responsible by structure the messages with address, arrays of data and methods for read and store data. Generic class for pack voltage, state of charge and currents have the same address on Can, that's why they are at the same class. Remains three classes for cell voltages, temperatures and BMS information.

3.1 Host computer software

In order to provide data for computer, the software was designed with the same classes in the TCU. The difference between two softwares is that the computer has the client version of MQTT.

The main objective of computer software is to plot charts containing BMS data in real time or in series selected by the user, like specific days or hours. I'm planning to use plplot, but this functionality is under construction.

3.2 Smart phone software

The smartphone software has to be the same that the computer software.

3.3 Integration and test

The plan test is to connect everything: BMS, TCU and computer at the same place and monitor the charging process of battery. The first test was unsuccessfully, because we couldn't read Can messages from BMS on TCU.

4 Results

In the beggining, we tested BMS and data from Can on Arduino with MCP2515 during charging of batteries. We have problems with temperature and state of charge data, what is probably because calibration of BMS sensors. After that, we put TCU in action, making the device publish random messages on broker and reading that on linux terminal. The delivery was successfully. Otherwise, when we tried to connect on Can it failled. But they were just pilot tests to make the project reliable and stable for competition.

5 Discussion

In our efforts to make this dream real, we shared so many experiences and knowledge. Learning C++ is one of the most expressive profissional experience on the Eletronical Engineering course, making us crazy about segmentation fault errors. And this bad memory usage occurs because inexperience on the language tools, like STL Containers. The greatest obstacle here is the queue library, when using this with other classes abstractions. When I implemented the classes of this project, I was thinking on be able to search the events on report, i.e. making a report of logs with date, time and the event that happened. The report was working fine before I tried to implement the search tool, that's make me think about the methods that I implemented on classes to search the events, by using the Date to create a way to encapsulate events that share same day or also hour. By the way, I tried many forms to implement MQTT communication with Paho on C++, but it wasn't successfully. When I tested the MQTT with Mosquitto on Linux it was easily successfully to communicate with my ESP12E.

Conclusions

The project wasn't successfully on the way I initially proposed. On the other hand, I found interesting results using ESP and Linux combination, that make my project able to reach the objective initially proposed. Now, I can implement the embedded system on ESP12E, that communicate with BMS by CAN and with some Linux by MQTT.