Engineering Experiences 3, Electronics-ICT, May 2025, Bruges, Belgium

Low-energy backup communication system for hydrogen racecar

Jarno Mechele Joey De Smet Robijn Ameye

Faculty of Engineering Technology, KU Leuven - Bruges Campus
Spoorwegstraat 12, 8200 Bruges, Belgium
{jarno.mechele, joey.desmet, robijn.ameye}@student.kuleuven.be

Abstract

Geef een beknopte samenvatting van het uitgevoerde onderzoek. Vermeld de conclusies die zijn getrokken en de mogelijke implicaties daarvan.
geeft probleemstelling,Aanpak,methode,Belangrijkste resultaten,Conclusies en implicaties

Keywords—Low power, Long-range wireless, Embedded systems

I. Introduction

In hydrogen-powerd endurance racing, uninterupted telmetry and communication between the pitwall and the car are essential for both competitive performance and for driver safety. As system could fail due to multiple reasons, and cause a loss of communication costing laps or even endager lives, ad dedicated low-power backup is required. This paper therefor proposes a sub-miliwatt, long-range wireless solution capable of transmitting and receiving both critiacal sensor data and messages to the driver over distances up to 2km (the approximate diameter of the Le Mans circuit). By combining a encrypted LoRa-based RF link with embedded speech synthesis en WAV playback on an ultra-low-power STM32U5 microcontroller, our design ensures that even in the event of primary-system failure, the pit-crew retains awareness of the car's most critical data and issue instruction to the driver.

II. Firmware Design and Implementation

This section presents the firmware development for the STM32U5-microcontroller [1], which forms the core of the low-energy backup communication system. The firmware is responsible for real-time handling of LoRa communication, sensor data processing, and voice output via speech synthesis or WAV file playback. To achieve real-time performance, the firmware is built on FreeRTOS [2], as illustrated in Figure-1. This enables clear separation of tasks with different priorities, making development more structured and maintainable.

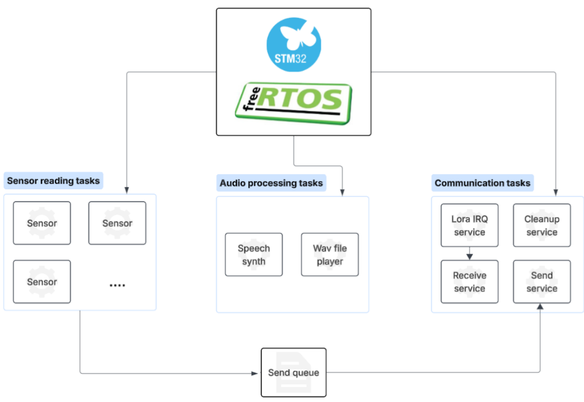

Fig. 1. Firmware system architecture overview

A. Firmware Architecture

As seen on Figure-1, the architecture is organized into three different modules which use inter-task communication mechanisms provided by FreeRTOS.

a) LoRa Communtion Group

This group consists of four different tasks: *Receive*, *Send*, *IRQ Handling* and *Cleanup*. The Recieve task blocks on a semaphore that will be set when the IRQ handling task get interupted by the sx1276-module [3], to notfiy (rx/tx)-completion.

b) Sensor Management Group

c) Audio Processing Group

d) Power and Memory Management

III. Backend and graphical user interface Design and Implementation

IV. Hardware Design and Implementation

V. Testing

VI. Conclusion

References

[1] STMicroelectronics, *STM32U5-series overview*, 2023. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32u5-series.html
[2] Real Time Engineers Ltd., *FreeRTOS Kernel Reference Manual*, 2024. Available: https://www.freertos.org/Documentation/RTOS_book.html
[3] Semtech Corporation, *SX1276 LoRa Transceiver IC*, 2025. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1276