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

Summary of the research and conclusion, Problem, method, results, conclusion....

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, a dedicated low-power backup is required. This paper therefore 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 and 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 describes the firmware developed for the STM32U5 microcontroller [1], which forms the core of our low-energy backup communication system. It handles real-time LoRa communication, sensor data acquisition, and voice output via speech synthesis or WAV playback. To guarantee deterministic timing, we bui9ld on FreeRTOS [2], as illustrated in Figure 1. Task isolation and priority levels make the codebase modular and maintainable.

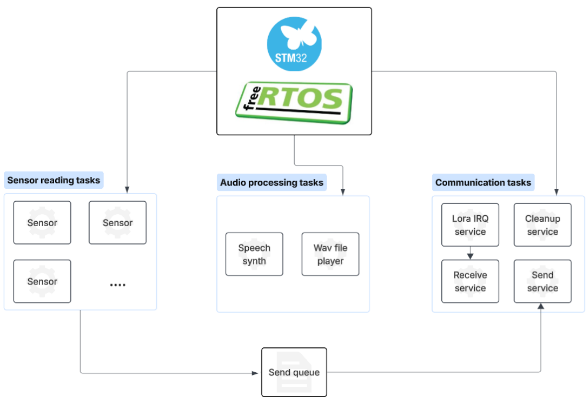

Fig. 1. Overview of the FreeRTOS-based firmware architecture

A. Overall Architecture

Figure 1 shows three primary functional domains, each implemented as one or more FreeRTOS tasks communicating via inter-task communication mechanisms provided by FreeRTOS.

B. LoRa Communication

- **IRQ Handler** Waits on the SX1276 [3] interrupt line to detect packet RX/TX completion, then gives a binary semaphore.
- **Receive Task** Blocks on that semaphore, retrieves incoming packets, decrypts them with hardware-accelerated AES-128 in CTR mode, and forwards them for processing.
- **Transmit Task** Pulls outgoing messages from a FreeRTOS queue, encrypts and formats them, then issues the LoRa send command.
- **Cleanup Task** Periodically scans stored packet buffers for timeouts and frees associated heap memory.

C. Sensor Management

Each sensor (e.g. temperature, pressure, speed) runs its own task at a low priority. Tasks periodically sample the hardware interface, package reacdings, and enqueues them for transmission.

D. Audio Processing

- **Speech Synthesis Task** A port of `espeak-ng` [4] with all file I/O replaced bu in-memory C arrays. Wich dequeues strings from a FreeRTOS queue, synthesises, streams audio to I2S hardware.
- **WAV playback Task** Streams hard-coded WAV data (e.g. racing flags, standard phrases) to the I2S hardware.

E. Power and Memory Management

All tasks are assigned carefully chosen priorities (Table I) so that time-critical communication tasks preempt lower-priority work. We enable FreeRTOS tickless idle to allow the STM32U5 to enter deep sleep whenever the system is idle.

TABLE I. Task priorities and stack usage

Task	Priority	Stack (bytes)
LoRa IRQ Handler	8	128
Speech Synthesis	7	48 000
WAV Playback	7	256
LoRa Transmit	5	128
Cleanup	5	128
Sensor (each)	3	256

III. Backend and graphical user interface Design and Implementation

IV. Hardware Design and Implementation

V. Testing

A. Test Setup

The system was evaluated primarily in a laboratory environment. Key aspects such as power consumption, LoRa communication integrity, and audio output functionality were tested. Most tests were conducted manually, involving the transmission of periodic test messages and monitoring system responses via serial output and audio feedback. Power draw was observed using an inline current measurement tool, and LoRa transmissions were validated using logic analyzers and debug logs.

B. Communication Testing

To verify LoRa communication, encrypted packets were periodically transmitted and received between two system nodes. Tests confirmed successful decryption, CRC validation, and end-to-end data integrity.

C. Audio Output Testing

WAV file playback was validated by connecting the system's analog output to external speakers and assessing the clarity of preloaded audio samples. Speech synthesis was tested by transmitting predefined sentences to the device and evaluating intelligibility through blind listening tests. Testers rated intelligibility qualitatively. While overall clarity was sufficient for basic commands, future improvements may include tuning output filters and optimizing speech synthesis parameters for better naturalness.

VI. Conclusion

Acknowledgment

The authors would like to thank the supervising professors and lab instructors for their guidance throughout the development of this project. Special thanks go to the hydrogen race team for providing valuable context and support related to the vehicle's communication needs.

Additionally, the authors used generative AI tools to assist with language refinement and grammar correction during the preparation of this manuscript.

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

[4] eSpeak-NG *eSpeak NG – Next Generation Text to Speech*, 2025. Available: https://github.com/espeak-ng/espeak-ng