Aurora V Avionics Documentation

FreeRTOS Task Architecture

RMIT University

Version 1.0
Last Updated 2024–07–14
Major Contributors . . . . Matthew Ricci

Date	Changes Made	Made By
2024–07–14	Create initial document	Matthew Ricci

Contents

1 Architecture Overview

The Aurora V avionics firmware package implements FreeRTOS for task scheduling. The firmware is designed to provide logical implementation of hardware interfaces for communications and telemetry, state calculations, data logging, and more, while adhering to stringent timing requirements.

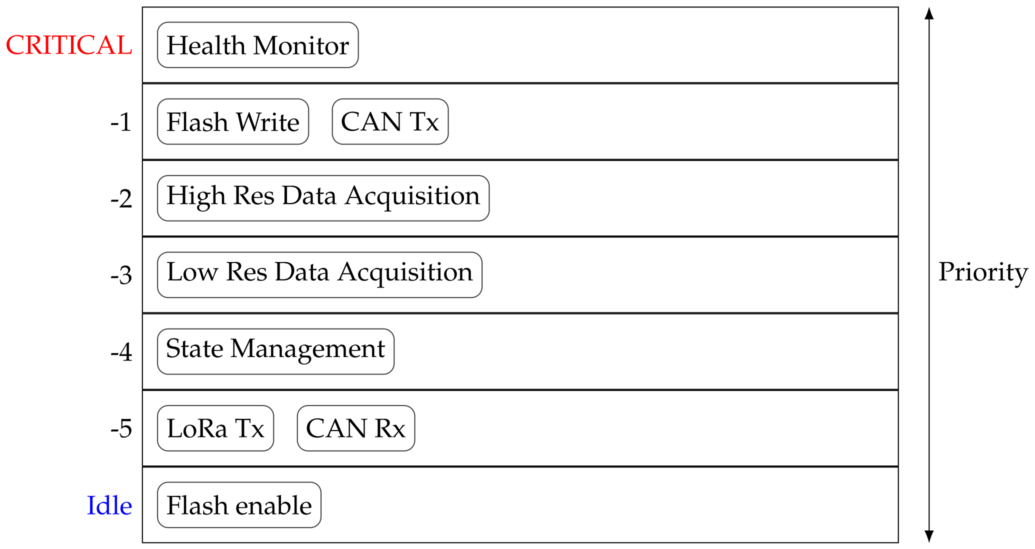

Figure 1: Avionics RTOS task hierarchy

Figure 1 illustrates a high-level overview of the task architecture. Tasks with higher priority are designed to execute promptly whenever they are not blocked, ensuring critical operations are performed without delay.

Lower priority tasks are scheduled with more flexible timing but may enter critical sections to prevent preemption during essential operations.

2 Task Description

2.1 Health Monitor

<To be designed>

2.2 Flash

Flashing of data primarily occurs during idle time on the CPU. This is managed through a task group flag that enables the operation, unblocking the task that handles writing of flash.

2.2.1 Enable

The avionics firmware implements a FreeRTOS idle hook that runs when no tasks are occupying time on the CPU.

If a launch event has occurred, the memory buffer is polled to determine if data is ready to be written. When this is the case an event group flag is set to enable the flash write task.

2.2.2 Write

2.3 Communications

2.3.1 LoRa

2.3.2 CAN

2.4 Data Acquisition

2.4.1 High Resolution

2.4.2 Low Resolution

2.5 State Management