Space Systems Lab Firmware and Safety Verification

Hersch Nathan

November 2024

1 General Requirements

This software is to support the KRUPS missions. The current mission is KRUPS Aboard Norwegian GhostSat (KANGS). The hardware paradim is based around an ESP32-S3.

This work is porting Matt Ruffner's firmware for Adafruit Feather M0/M4 to the ESP32-S3. This including switching from the Arduino IDE to ESP-IDF.

The goal of this project is to make the firmware more modular and configurable for each mission and hardware.

2 Background Information

2.1 freeRTOS

freeRTOS is a real time operating system. A real time operating system works by having tasks that run at different priorities. The tasks are scheduled by the operating system and those tasks occur through a determanistic prosess. The tasks can communicate with each other using message queues. The tasks can also synchronize with each other using semaphores.

2.2 ESP-IDF

ESP-IDF is the Espressif IoT Development Framework. It is the official development framework for the ESP32 and ESP32-S3. It is a set of libraries and tools that are used to develop software for the ESP32 and ESP32-S3.

3 Architecture Design

The firmware's framework is designed to be modular and configerable thmission and each hardware. The backbone of the firmware is using freeRTOS tasks to handle the different modules. Each taks is its own module that encapsulates the relevant hardware. The tasks communicate with each other using a publisher/subscriber model.

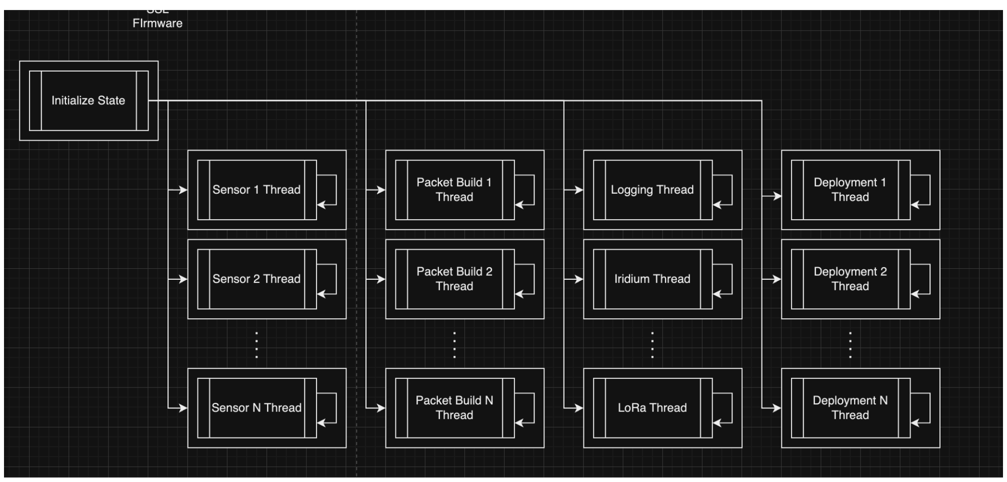

Figure 1: Firmware Architecture

In the figure 1, the main tasks there are four types of tasks.

3.1 Sensor Tasks

Each sensor task manages one sensor or group of sensors, see 2. At startup it configures the sensors with the necessary parameters, then it calls a looping task. That task, takes the semaphore for the used serial communication interface (I2C, SPI, UART, etc). It sends the command to the sensor to get the data, reads the response, and then releases the semaphore. The task then sends the data to the packet build tasks queues.

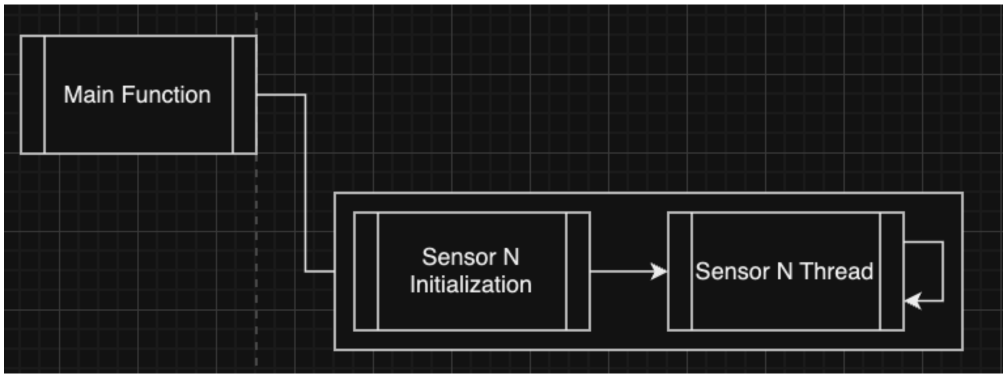

Figure 2: Sensor Task Architecture

3.2 Packet Build Tasks

3.3 Wireless Communication Tasks

3.4 Deployment Tasks

4 Kentucky Flight Computer (KFC) Requirements

5 FemptoSats Requirements

The FemptoSats submission is to test the viability of using Wifi/LoRa for intercapsuole communication.

5.1 Sensors Requirements

The FemptoSats will have the following sensors:

- BME280 Temperature/pressure/humidity sensor
- BNO086 9 - Axis IMU

5.1.1 BME280 Temperature/pressure/humidity sensor

The BME280 will be run with the following configuration settings:

5.1.2 BNO086 9 - Axis IMU

The BNO086 will be run with the following configuration settings:

5.2 Wireless Communcation Requirements

the FemptoSat will use the following Wireless Communcation modules:

- Integrated Wifi
- LoRa

The Wifi will fail over to the LoRa when the Wifi gets out of range

6 Rocketstation Transmitter (RST) Requirements

uart to raspi to shut off need

7 Rocketstation Requirements

uart to raspi to shut off need

8 Groundstation Requirements