THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Real time SDR based ADSB transponder tracking in a selective sky area with location sensing and Bluetooth low energy

Samuel Kwort & Flynn Kelly, Supervised by Dr. Matt D'Souza

Introduction

This project aims to create a system that can track and display ADS-B transponders of nearby aircraft within a designated portion of the sky. By harnessing the capabilities of Software Defined Radio (SDR) and embedded systems, the goal is to provide real-time transponder data including aircraft IDs, latitude and longitude based on a selected portion on sky. The selected portion of sky is localised through a magnetometer, GPS sensor which are fused in a Madgwick filter. The communcation protocols utilised include UART, iBeacon BLE and NUS BLE. The outcome of the project can be quantified using the below Key Performance Indicators:

1. System Performance The system should be able to track and visualise transponder data of aircraft in selected directional window in real time.
2. User Interface and Experience The visual display of transponder information should refresh at once per second. The user interface options (ie iterate button) should be actioned in at least 50ms.
3. Sensor 'Sky-Selection' AccuracyThe use of magnetometer and GPS sensing modules will enable accurate detection of transponder enabled aircraft in a given sector defined by sensors. This will be accurate where possible (ie outdoor area)
4. Connectivity and Communication The system should reliably connect and communicate in the vicinity of a small area, regardless of internet connectivity - using BLE and UART protocol.
5. Deployability and Scaleability The system is easily deployable via clear user instruction documentation along with intuitive protocol between relevant hardware. Potential for system to be networked as extension.

System Overview

The implementation of this project can roughly be broken down into the below block diagram, which utilizes an array of micro-controllers, sensors and iterfaces.

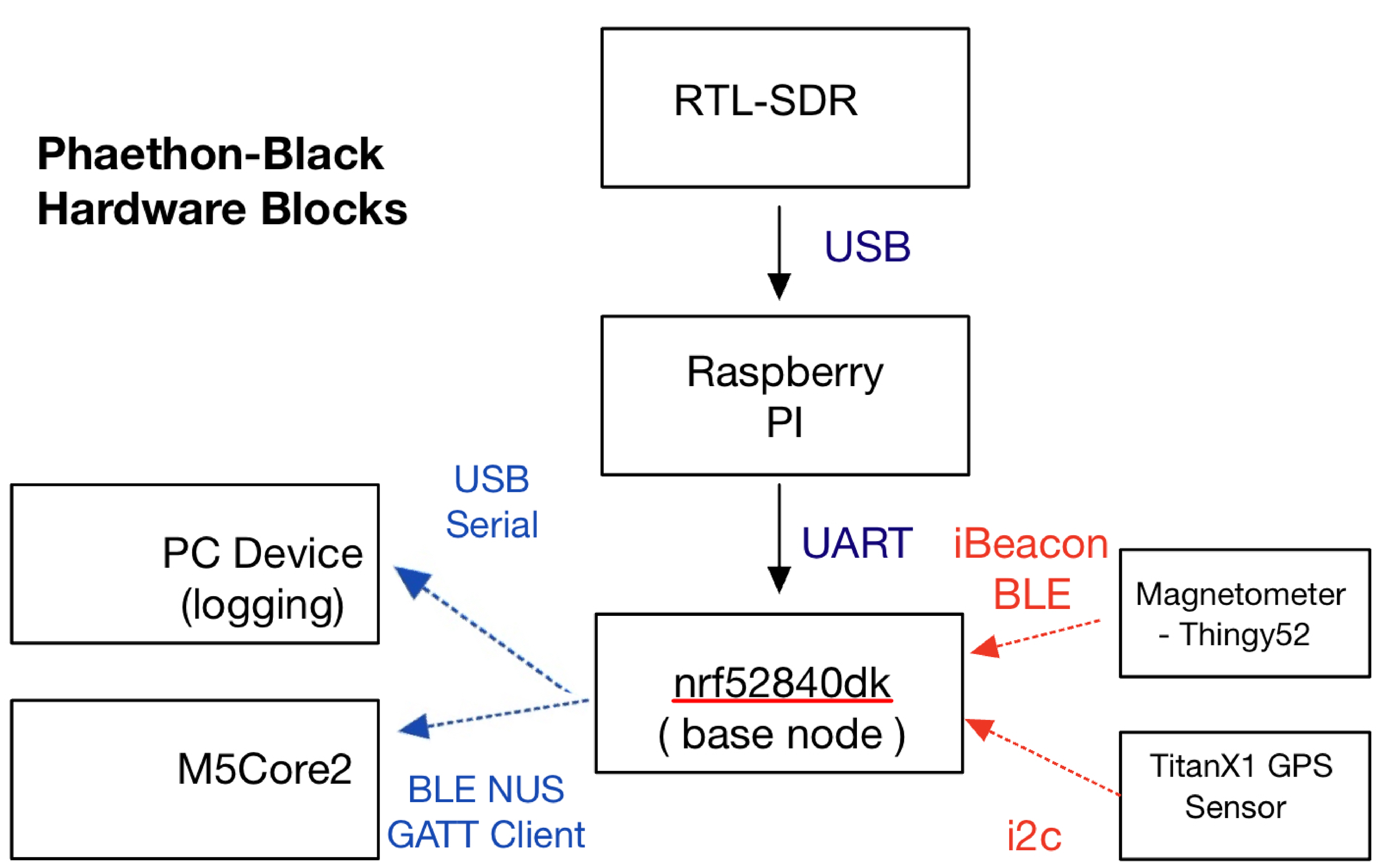

Figure 1. System Overview

Importantly, the arrows of the block diagram also reflect communication protocols, including Sam's custom DLT protocol.

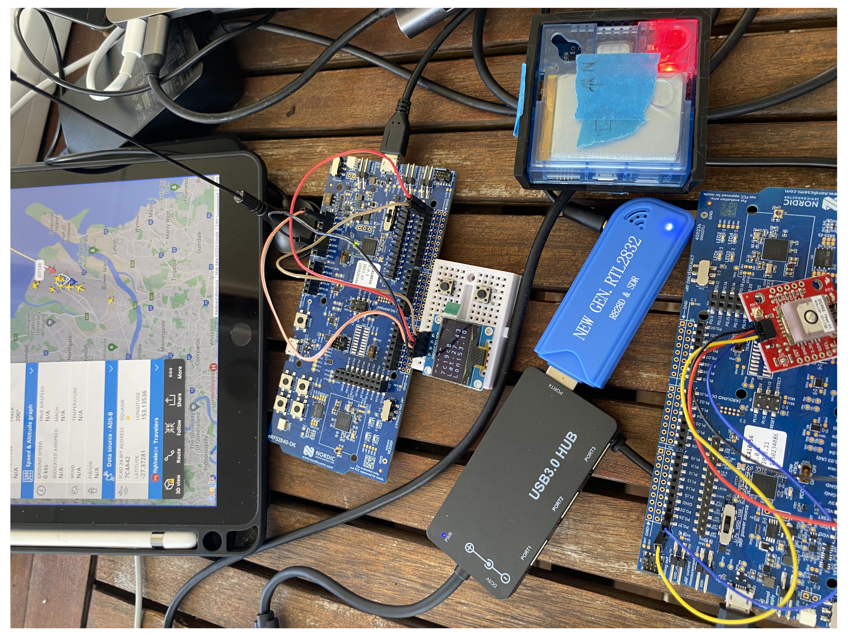

Figure 2. Setup

Methodology

The successful delivery of this project revolved around a lot of moving parts which had to be designed in a modular way but integrated into a larger system. The below design approaches reflect a broad overview of the design and implementation process;

1. KPI breakdown into achievable sections
2. High level architecture of system (Figure 1)
3. Sensor & driver module development
4. Sky selection through fusion of GPS & Magnetometer values
5. NRF52840dk, m5stack core2 and thingy52 configuration
6. Development of UART & BLE communication & standards including custom DLT protocol
7. Development of SDR to embedded data structures and communications
8. Front end display of currently tracked planes
9. Product testing, integration & validation

Design Limitations

1. Magnetometer Interference: the magnetometer on the thingy52 is prone to EMI approx 10' offset.
2. SDR ADSB Signal Reception: small antenna, better outisde.
3. GPS Sensor Accuracy: needs to get a lock - better outside.
4. Localisation: To accurately track planes in a chosen region of sky the system should capture both location and heading values from the system.
5. M5Stack Core2 Zephyr DRAM Bottleneck: Confronted this problem in final testing, when pre implemented user interface was combined with zephyr BLE GATT API's resulted in DRAM overflow on the ESP32. Supplemented via a simple LCD Display.

Field Deployment Plan

As mentioned previously, for maximal sensor performance, the system should be configured outdoors. To obtain a relevant reference, the thingy52 should be orientated North, and the GPS should be allowed at least 30 seconds to obtain a lock. Laptop as proof of concept, can be run on any embedded linux / SBC with access to dump1090. Refer to wiki for config guides.

Results

Overall, the system demonstrates an effective proof of concept, combining a range of complex protocols and interfaces. These include, localization, sensor fusion through a madgwick filter, DSP (completed through dump1090 library), and GATT BLE communications.Arguably the most impressive result was Sam's successful implementation of DLT functional interface; custom protocol.

Tracking in a selected window

Figure 2, along with the following link https://youtu.be/sCdQtdMZH74 show the system accurately tracking ADSB data from planes in a chosen window (based on magnetometer orientation). Other than slight offset from the magnetometer readings, the sky selection functioned well, with the GPS providing accurate readings.

M5 RAM issue

The project was not without its flaws, late in testing we discovered a problem which altered the deliver-ability of our front end visualisation plans.

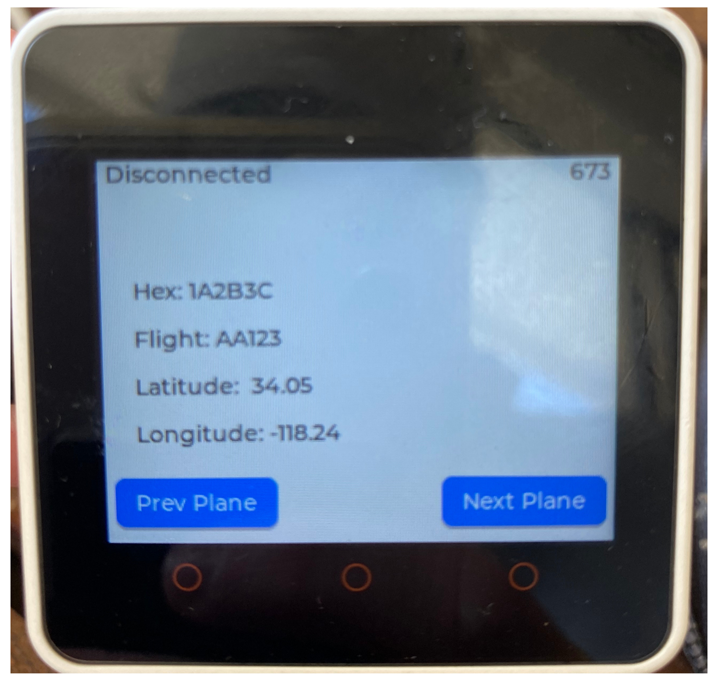

Figure 3. M5 Front End

Due to zephyr kernel and API bloat, it was not possible to have BLE GATT NUS service and LVGL front end visualisation build on the m5 at the same time (46% and 77% usage respectively).

Conclusions

1. System Performance Successfully real time tracks targets.
2. User Interface and Experience User interface not on m5 but is appropriate substitute.
3. Sensor 'Sky-Selection' Accuracy Magnetometer and GPS effectively select zone.
4. Connectivity and Communication Flawless connection, DLT protocol through BLE NUS and BLE iBeacon.
5. Deployability and Scaleability Effective proof of concept, pi script is scaleable to any SBC / embedded linux.

References

[1] Air Services Australia. Ads-b coverage. https://www.airservicesaustralia.com/about-us/projects/ads-b/ads-b-coverage/#:~: