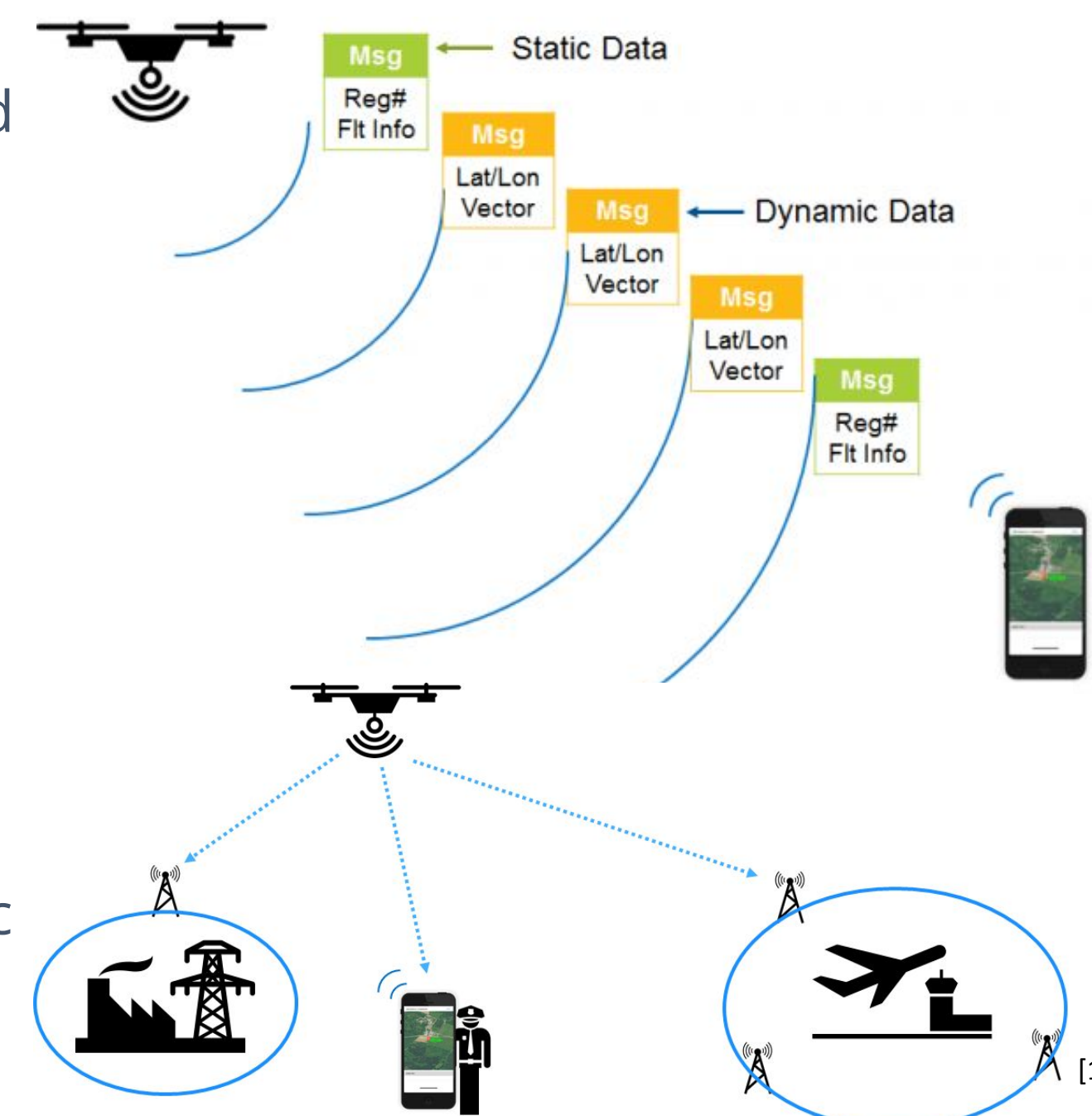


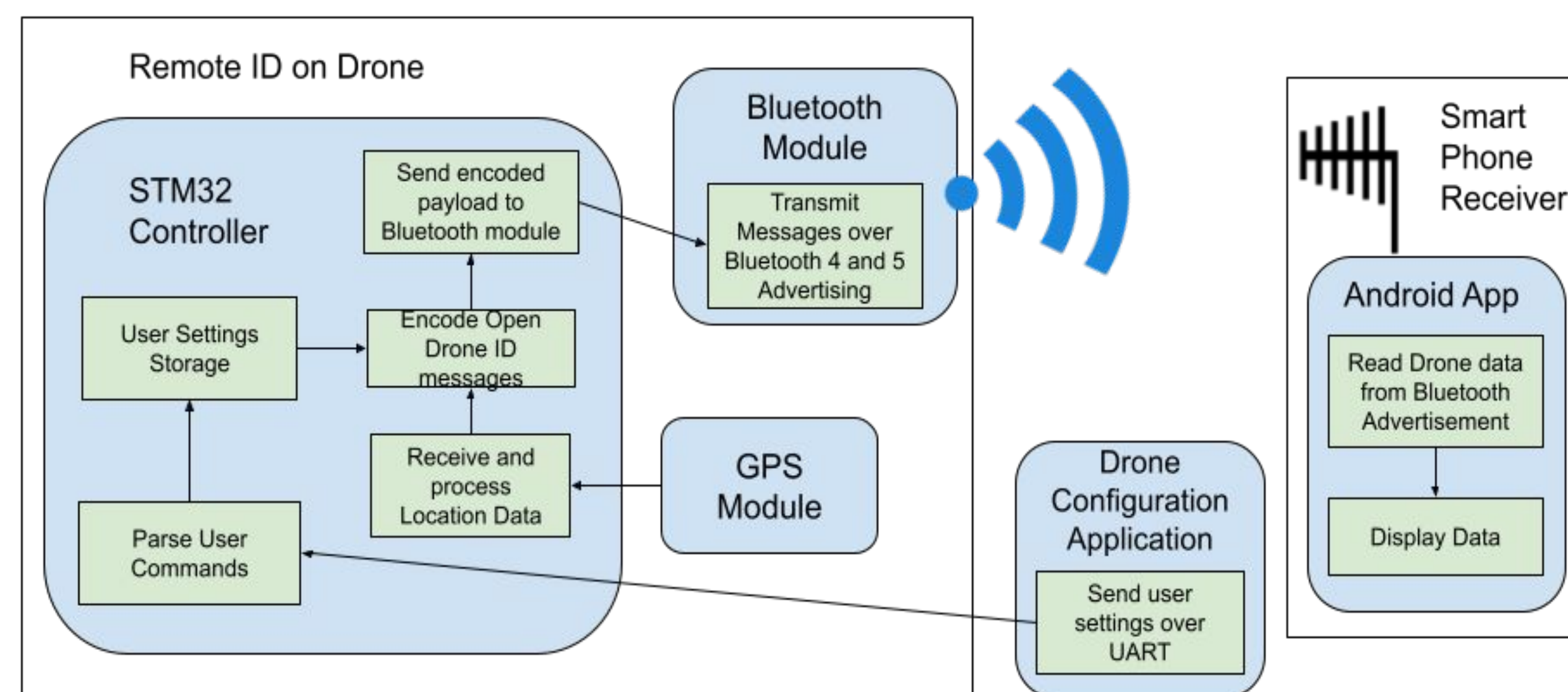
STUDENTS: JALEN BAUER, RYAN HALLGRIMSON, KING HONG, MARCO XU

Remote ID Standard

- Remote ID is a regulation instituted by the FAA that requires unmanned aerial systems (UAS), drones to **broadcast** their **location, identity, and status** via radio
- Remote ID information can be broadcasted using Bluetooth and WIFI
- These messages can be monitored by authorities and nearby air traffic



System Architecture



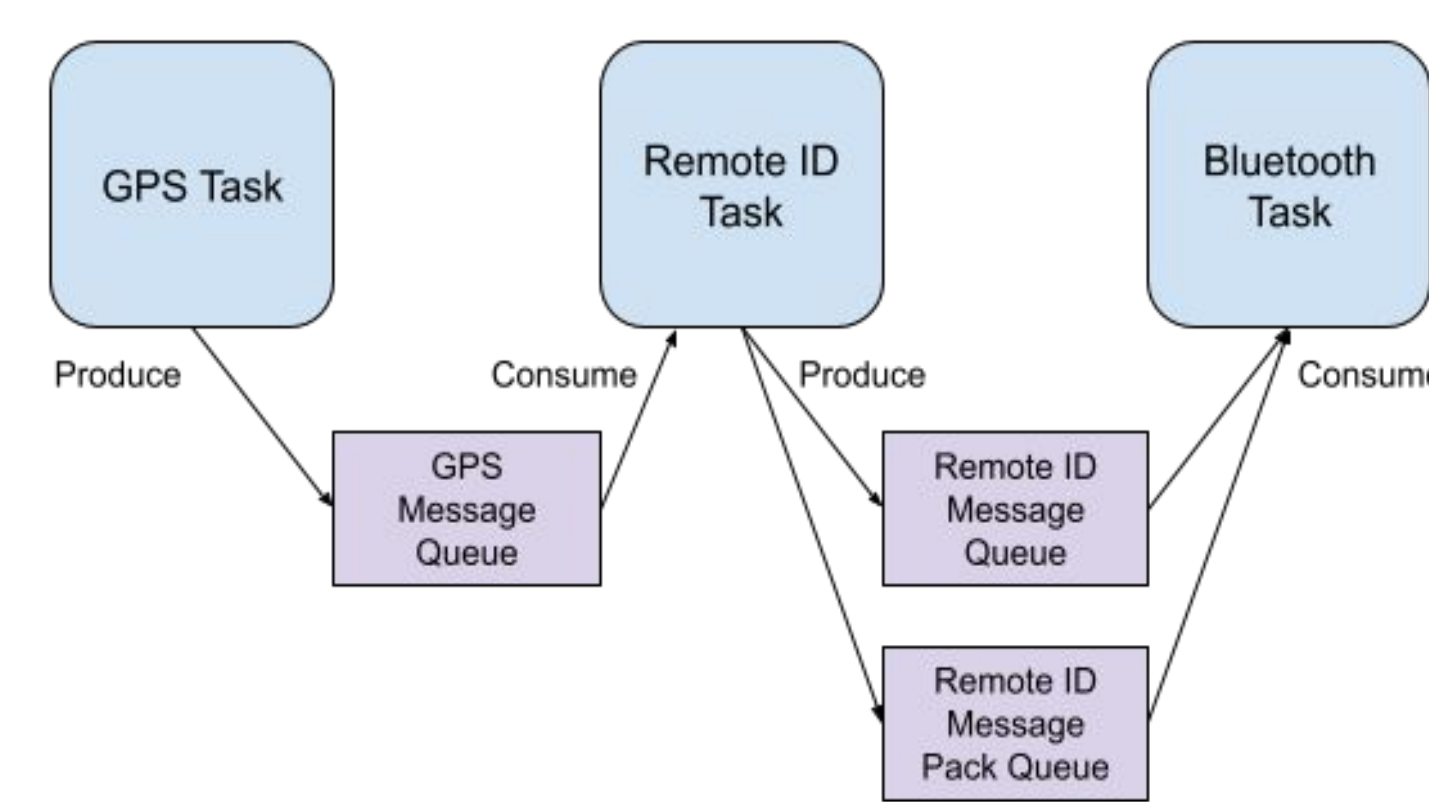
Software Design

Based on ARM Mbed:

- Modern C++, highly modular design
- Free, open-source RTOS
- Supports many targets, easy to port
- Community support

Modular and portable software design:

- Easy to test, maintain, and extend
- Easy to adopt to new hardware, resilient to supply chain disruptions



Task based architecture

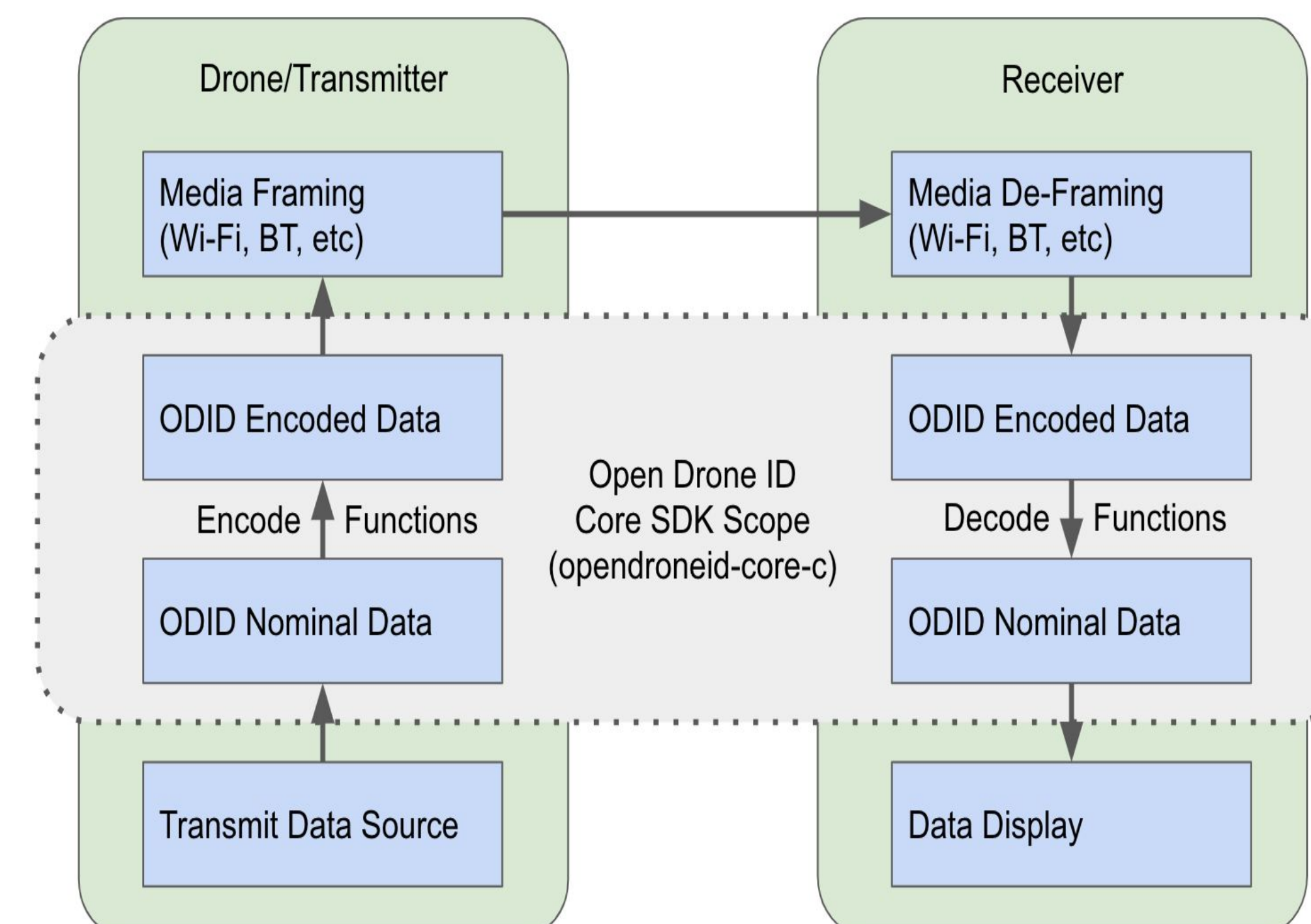
- Producer/consumer model
- Separation of concerns, highly modular
- Deterministic timing with Mbed RTOS

Independent GPS Receiver

- GPS module acquires geolocation data of the device with update rate once per second
- GPS task decodes and extracts location information required for the Remote ID data
- GPS task formats a message package and sends the package to the GPS message queue for the Remote ID task to process

Remote ID Encoding and Decoding

- The Remote ID messages have a standardized format they fit into
- Location, identification, etc are encoded into the Open Drone ID format messages
- The encoded messages will be sent over Bluetooth or WIFI
- We used `opendroneid-core-c` standardized library for encoding



Bluetooth Broadcasting

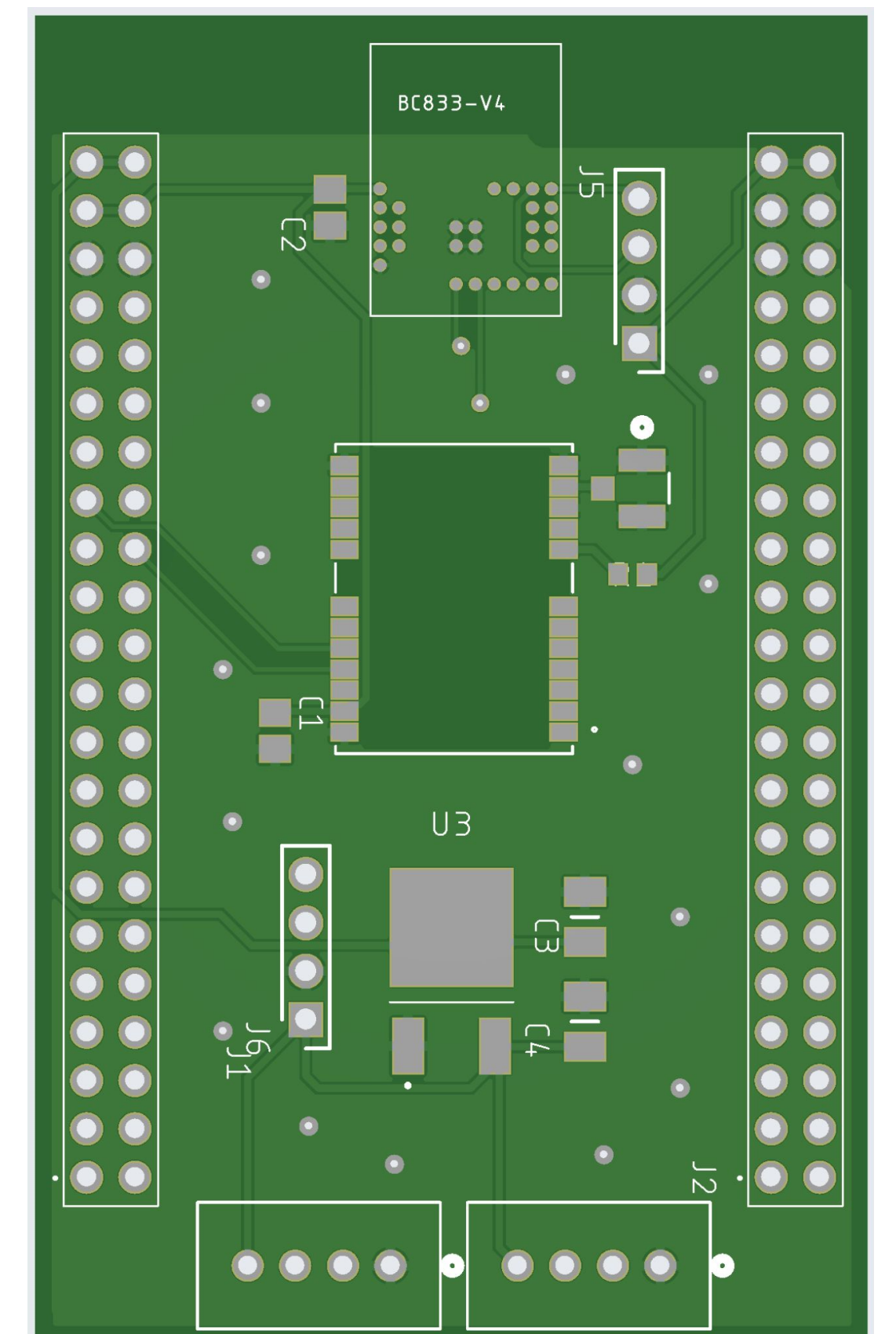
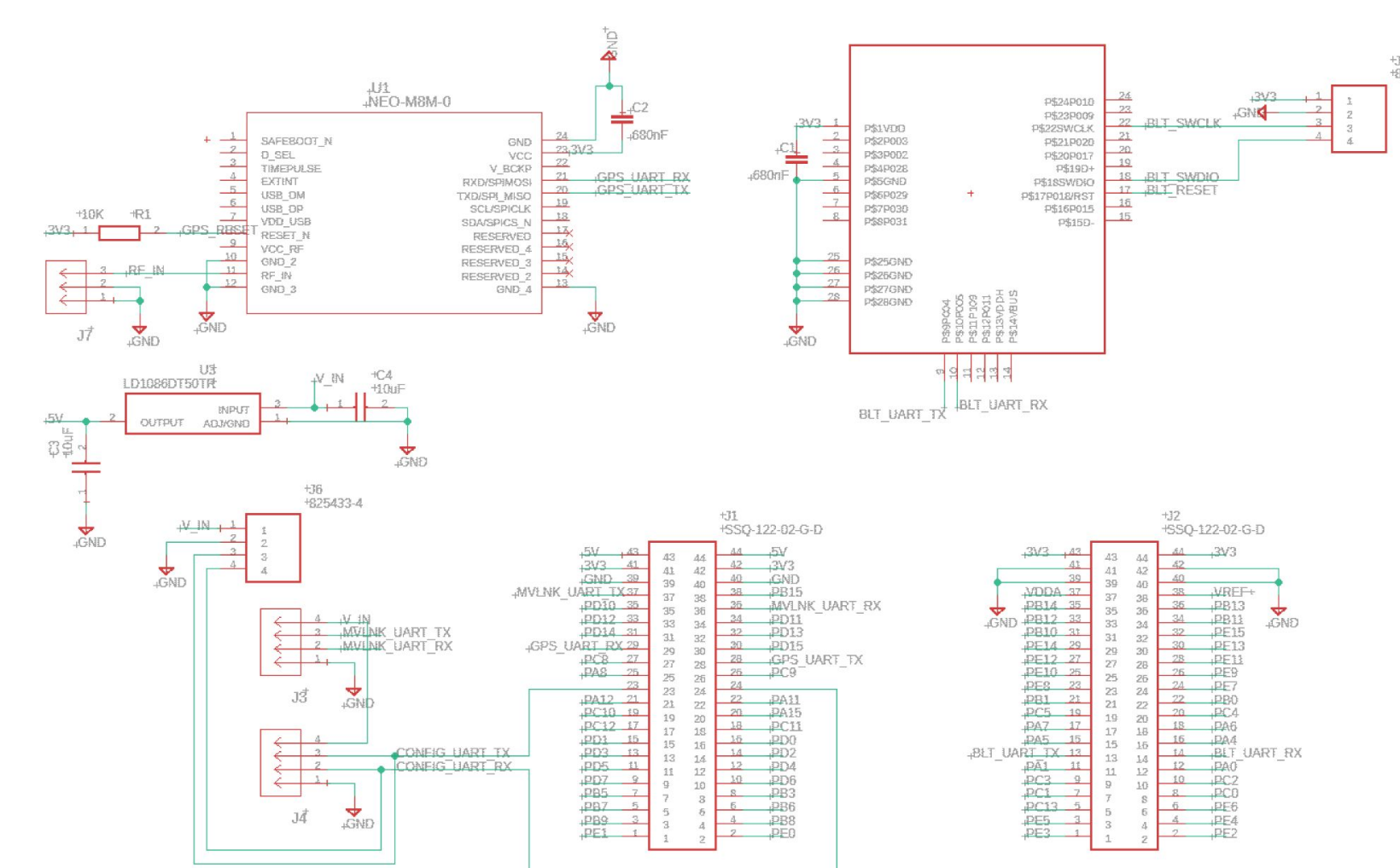
- Remote ID messages are transmitted using Bluetooth advertisement beacons (Bluetooth GAP)
- Compatible with both Bluetooth 4 (legacy) and Bluetooth 5 (extended)
- The Bluetooth firmware encodes then sends out both types of beacons
- The Remote ID receivers monitor the advertisement channels

Legacy Advertisement Frame									
Preamble (1 Byte)	Acc Addr (4 Bytes)	PDU Hdr (2 Bytes)	AD Addr (6 Bytes)	AD Len, Type (2 Bytes)	16-bit UUID	App Code	Counter (1 Byte)	Open Drone ID Message (25 Bytes)	CRC (3 Bytes)
0xAA	8E89BED6h	20h, 25h	<public>	1Eh, 16h	0xFFFF	0x0D	0x00 - 0xFF	<message>	<calculated>

Open Drone ID Message		
Msg Type (4 bits)	Version (4 bits)	Message (24 Bytes)
0x0 - 0xE	0x0 - 0xF	<Open Drone ID message>

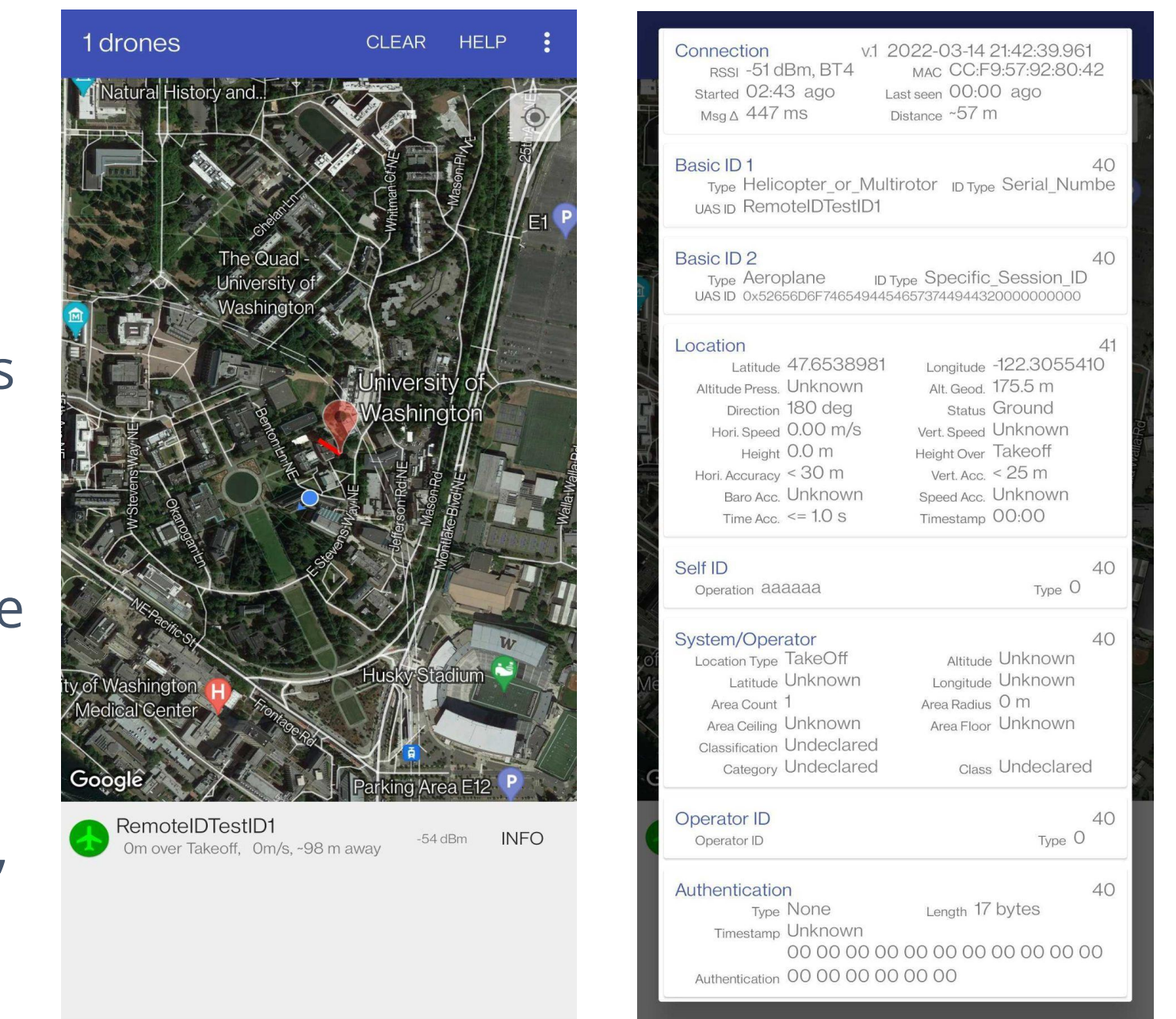
Hardware Design

- We design a printed circuit board to connect our Bluetooth and GPS modules to the board for the STM32 development board
- The shield design was chosen to minimize design and fabrication times and to reduce the impact of semiconductor shortages



Android Application

- We use example Open Drone ID Android receiver app to test our implementation
- Detects and receives Remote ID messages produced by nearby devices through Bluetooth
- Decodes the received message and displays the location of nearby Remote ID-enabled drones on the map
- We use this app to test compliance with the standard like format, interval, and range.



Future Work, References, and Acknowledgements

- Design a one-piece PCB with minimal size that encapsulates all modules
- Integrate Mavlink protocol for in-flight communication with flight controller of the drone
- Ability to broadcast over WIFI

[1] *Open Drone ID*. 11-Mar-2019. [Online]. Available: <https://www.opendroneid.org/>. [Accessed: 20-May-2022].

[2] ASTM standard F3411, 2019. "Standard Specification for Remote ID and Tracking", ASTM International, 2019

Faculty Mentor : John Reece
Industry Mentor : Matthew Hamilton
Teaching Assistant : Shruti Misra