Concealed Bike Anti-Theft Device

Electrical & Computer Engineering

Elizabeth Atkinson, Alex Wen, Srinidhi Raman

ECE 445 Spring 2022

4/28/2022

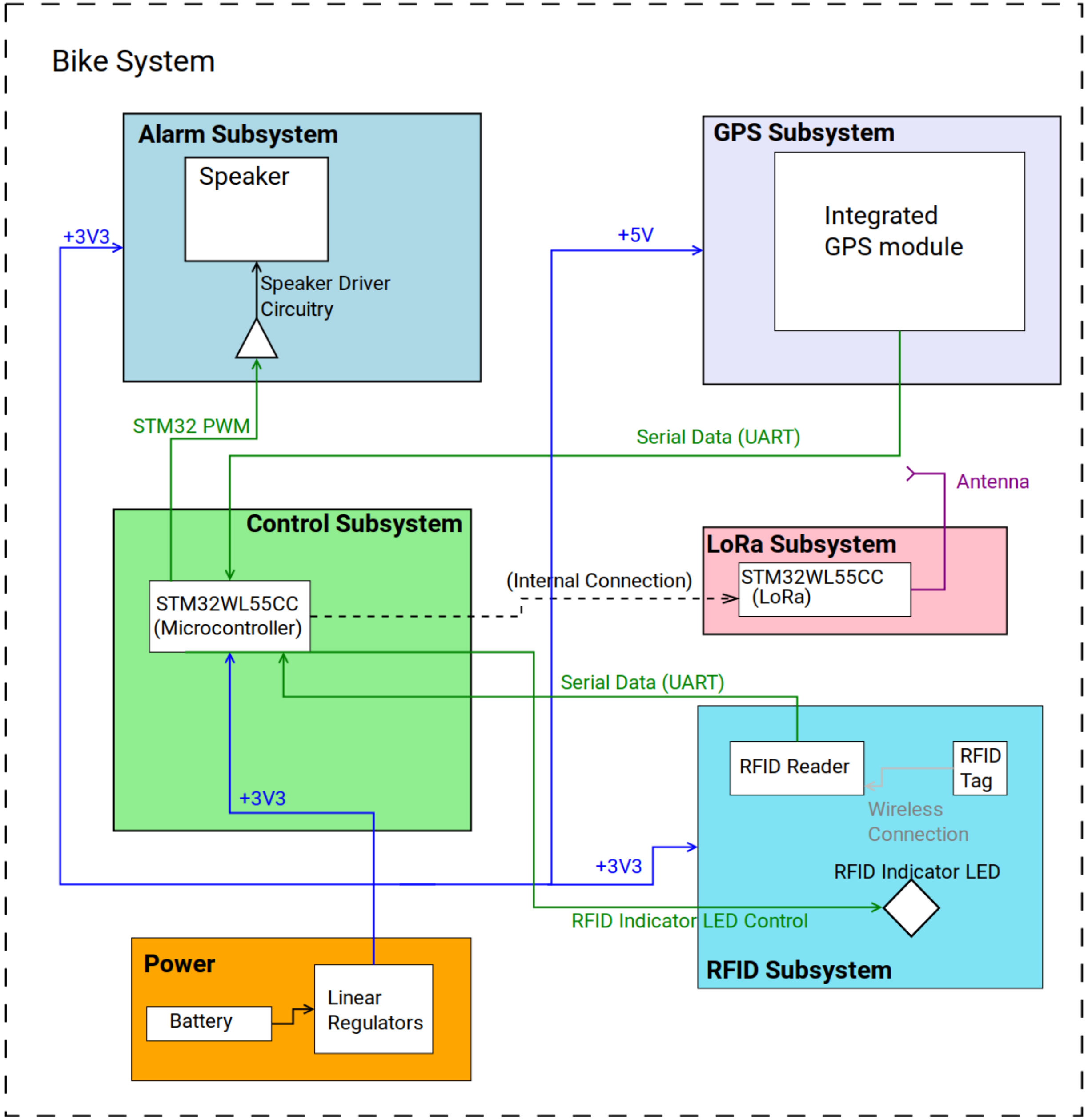
Background

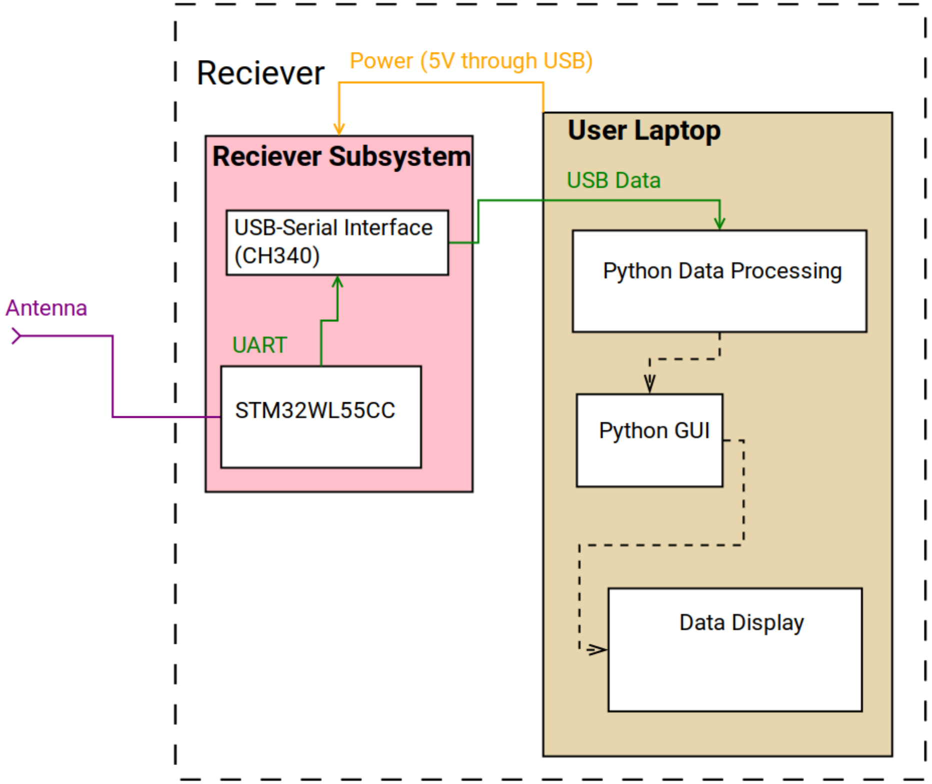
Problem:

- Bike theft is very prevalent on campus
- Bike locks are not always adequate to protect bikes
- Once a bike is stolen, it is extremely difficult to find and retrieve the bike

Solution:

- GPS tracker to keep track of location of bike
- Alarm that will sound when an unauthorized user tries to remove the bike
- LoRa transceiver to communicate GPS data and other parameters to the user base station

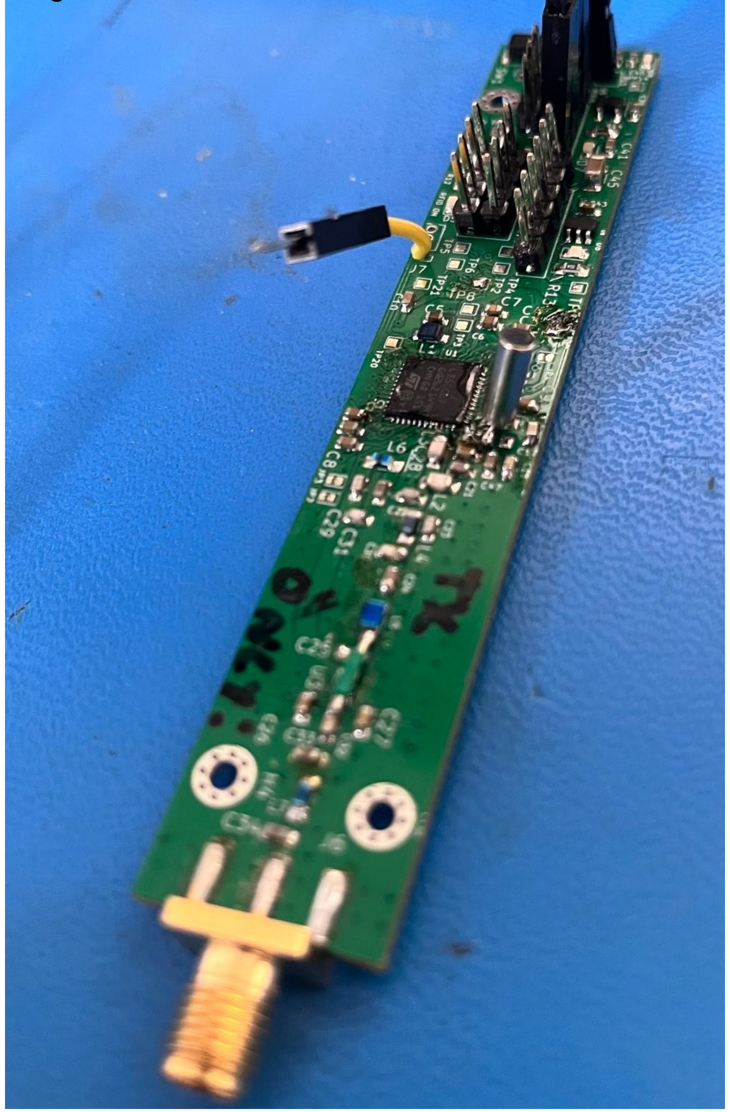




Concealed Bike Anti-Theft Device

Bike Unit: Control Subsystem

Fig 1: MCU/LoRa board.



Overview:

- STM32WLCC integrated microcontroller and SubGHz transceiver
- Dual core Cortex-m4/Cortex-m0

Requirements:

- The microcontroller implements a state-based control system with power saving measures in the idle/safe states.
- The microcontroller packetizes GPS data to be sent over LoRa.

Bike Unit: Control Subsystem

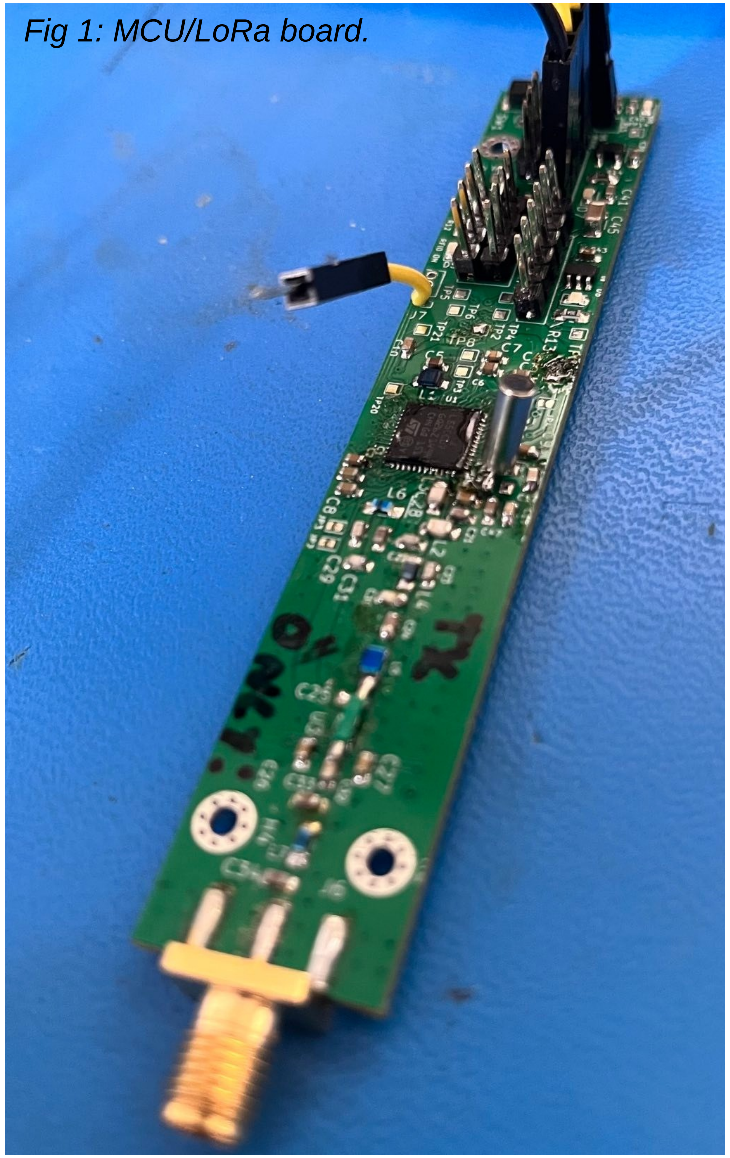


Fig 1: MCU/LoRa board.

Successes:

- The microcontroller implements the system state machine (refer to Appendix 1).
- The microcontroller communicates with GPS and RFID subsystems (UART) and controls the alarm with a PWM output.

Failures:

- None.

Bike Unit: LoRa Subsystem

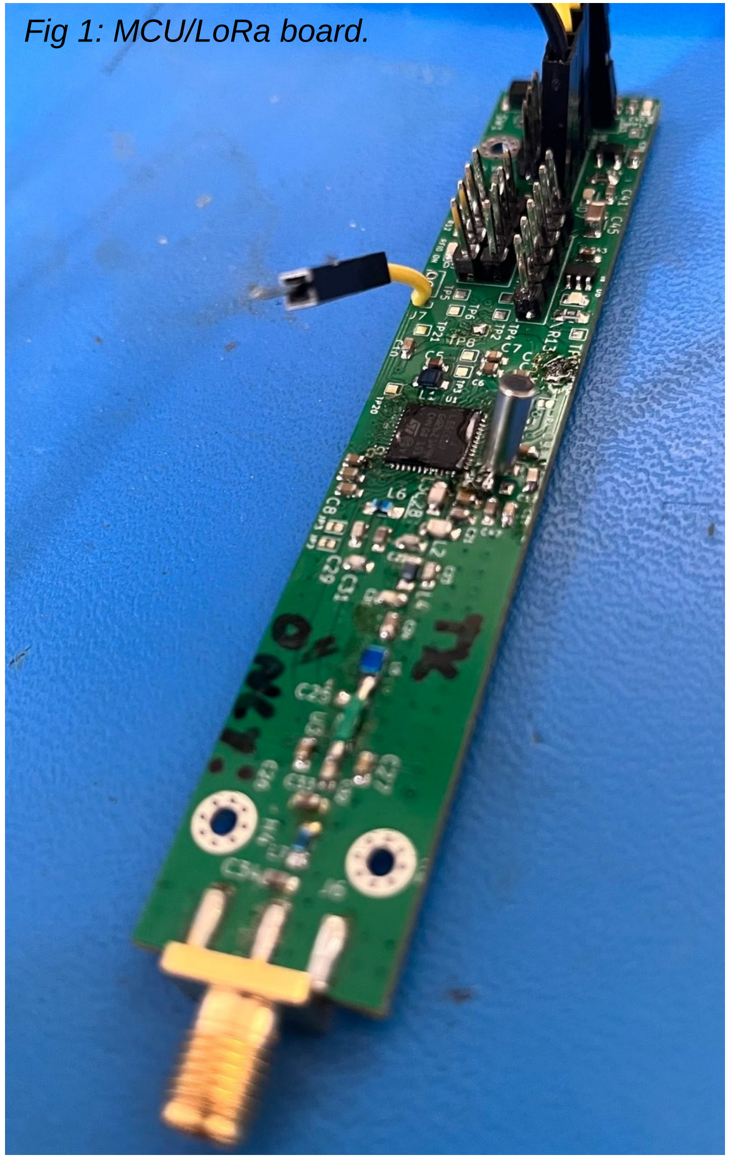


Fig 1: MCU/LoRa board.

Overview:

- Based on the STM32WL55CC integrated microcontroller and transceiver
- Firmware uses the ST low-level radio driver

Requirements:

- The LoRa Subsystem matching network and antenna have a VSWR of less than 10 at the RF output of the STM32WL55CC.
- The antenna is hidden in a common bike component so that it is not visibly obvious.

Bike Unit: LoRa Subsystem

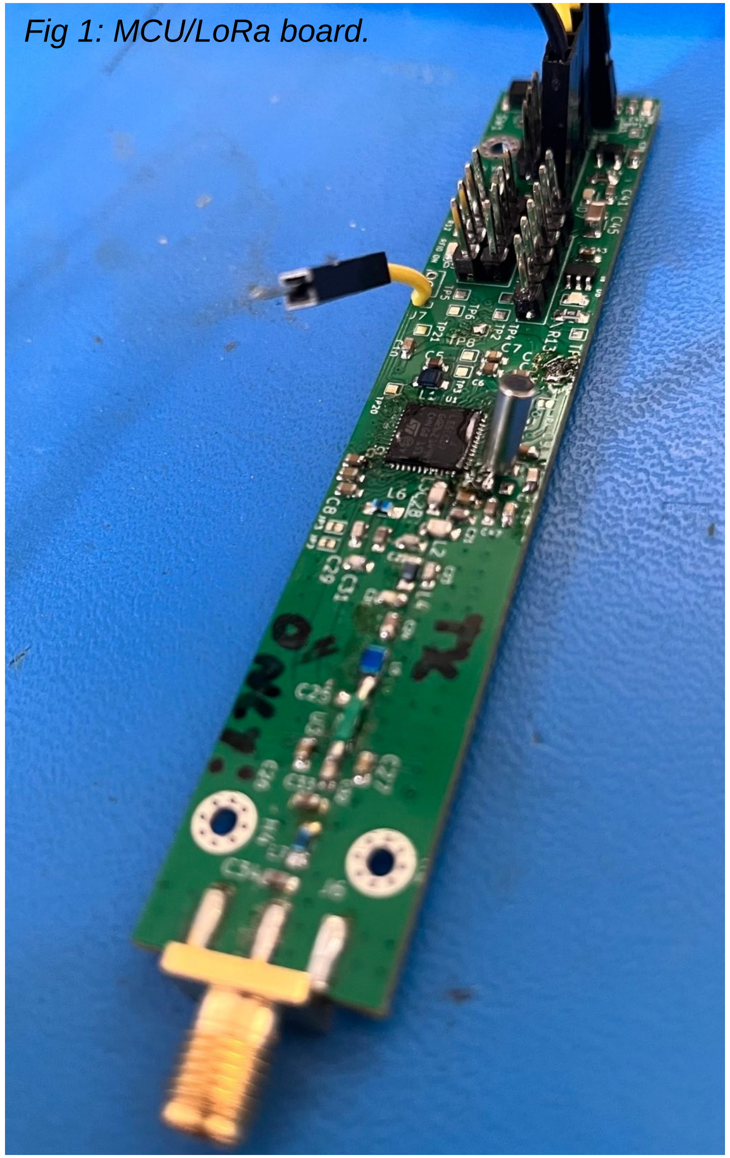


Fig 1: MCU/LoRa board.

Successes:

- Able to implement the ST-provide low-level radio driver.
- Radio driver code successfully transmitted on the STMWL Nucleo dev board (Fig 1).

Failures:

- Unable to transmit using LoRa on the PCB we designed. For a detailed engineering explanation of the suspected cause, refer to Appendix 2.

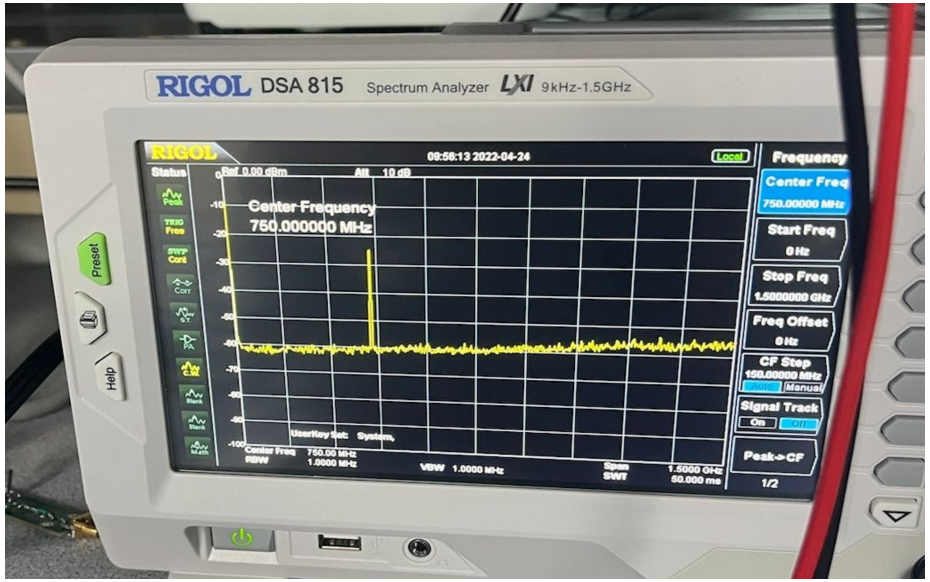


Fig 2: TX with nucleo board; note the frequency is 430 MHz. The nucleos are tuned for the 400-MHz band so we could not use them for LoRa which uses the 915 MHz band.

Bike Unit: RFID Subsystem

Fig 3: RFID carrier board.

Overview:

- Based around the ID-3LA RFID Module and external 1mH antena
- Communicates with microcontroller over UART

Requirements:

- The RFID detector will detect the user’s RFID tag within 10 seconds.
- The RFID subsystem will provide a visual indicator that the RFID tag has been detected within 1 second of detecting the RFID tag.
- The RFID subsystem triggers the alarm if movement occurs and the user’s RFID has not been detected.
- The RFID indicator detector LED is concealed in an unobtrusive location on the bike.

Bike Unit: RFID Subsystem

Fig 3: RFID carrier board.

Successes:

- All requirements were satisfied and demonstrated at the Demo.

Failures:

- Since the antenna was very small, the range was very low (<10 cm). In the future, a larger antenna could increase the range.

Bike Unit: Alarm Subsystem

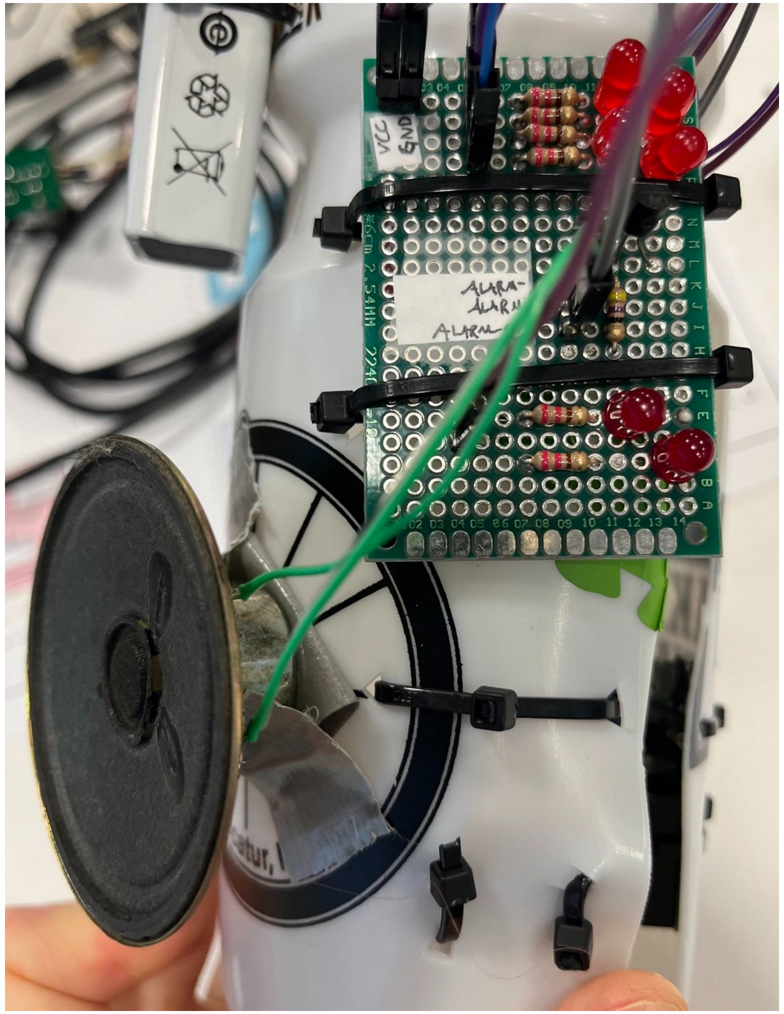


Fig 4: Speaker circuit in prototype device.

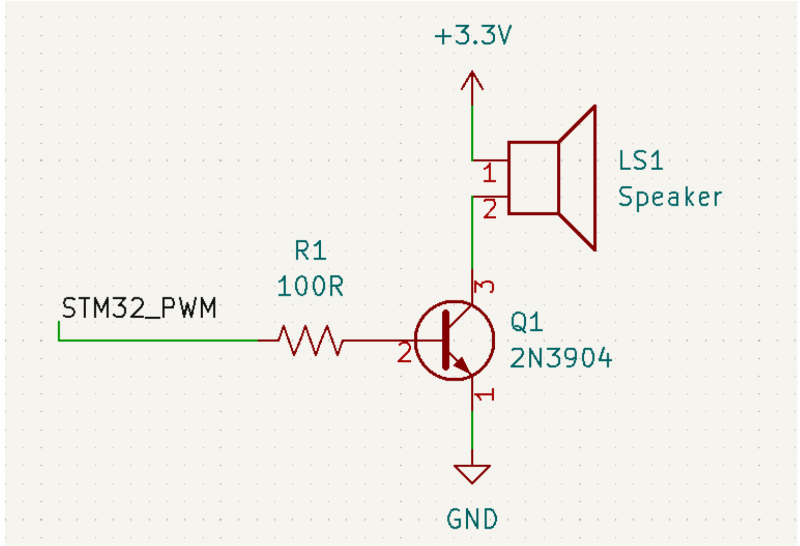
Overview:

- 5 kHz PWM output from the STM32 drives the base of a BJT
- BJT acts as a switch to drive the speaker

Requirements & Verification:

- The alarm makes noise when triggered by the microcontroller.
- The alarm circuit draws <1uA of current when not making noise.

Schematic:



Bike Unit: Alarm Subsystem

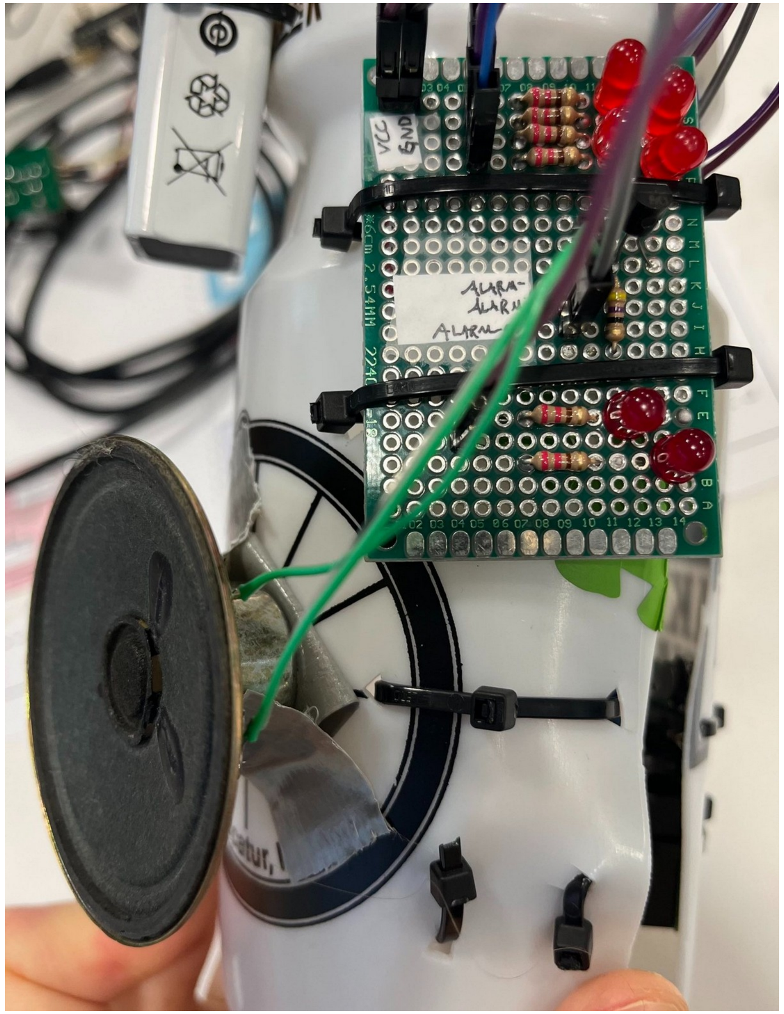


Fig 4: Speaker circuit in prototype device.

Successes:

- Speaker output is very loud and annoying.
- STM32 PWM output successfully drives the speaker.

Failures:

- Our original plan of using a piezo buzzer was not loud enough, so we switched to the speaker.
- The speaker is not small enough to fit in the bottle as shown in the image.

Bike Unit: GPS Subsystem

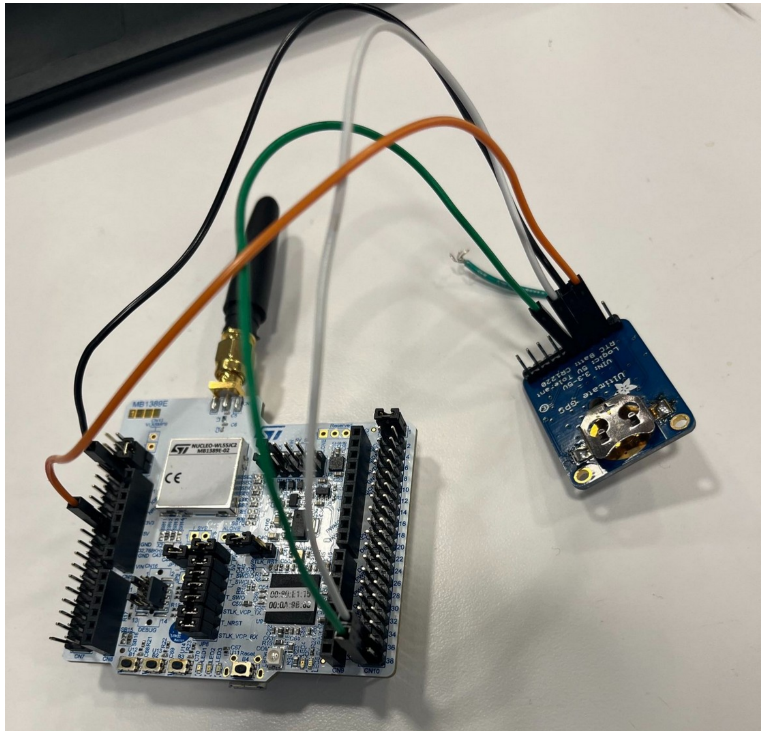


Fig. 5: GPS module (connected to Nucleo for testing).

Overview:

- Based on the Adafruit Ultimate GPS module V3
- Communicates with the microcontroller over UART

Requirements & Verification:

- The GPS module communicates with the microcontroller with an update rate of greater than 1 Hz.

Bike Unit: GPS Subsystem

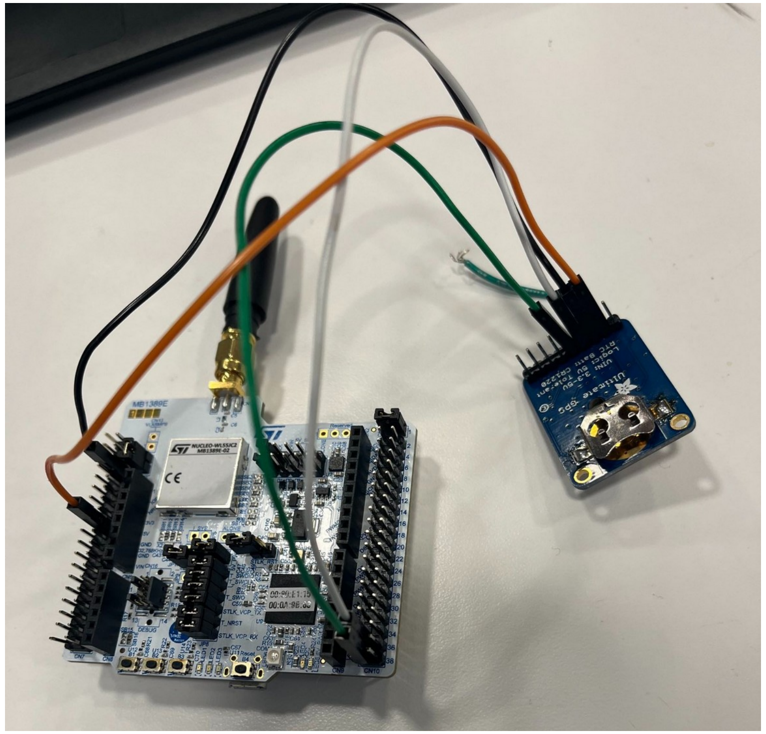


Fig. 5: GPS module (connected to Nucleo for testing).

Successes:

- Communicates with the microcontroller over UART.
- MCU firmware calculates distance between 2 points.

Failures:

- None.

```
(gdb) p gpsmsgbuf
$3 = {UTC_Time = "170921.000", latitude = "4006.9306", NS = 78 'N', longitude = "08813.6236", EW = 87 'W', pos_fix = 49 '1', sat_us
ed = 55 '7'}
(gdb) p/c gpsmsgbuf
```

Fig 6: Example of GPS data.

Base Station: Base Station Subsystem

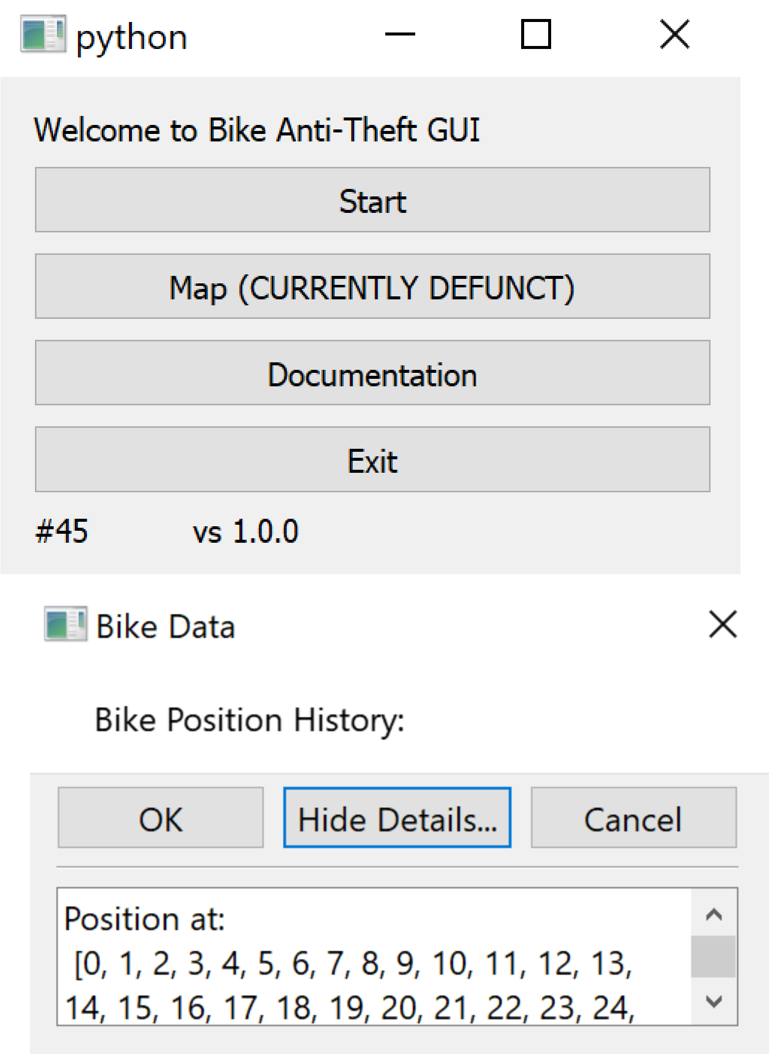


Fig 7: Base Station GUI Screenshots.

Overview:

- Same STM32WLCC microcontroller board
- Communicates with the computer over UART, using the CH340 interface IC.

Requirements:

- The RX board will interface with the computer over USB.
- The RX board will receive LoRa packets from the bike device.
- The GUI will display GPS data over the past day, week, month or year on a map.
- The GUI will allow the user to view plots of the average speed and distance travelled.

Base Station: Base Station Subsystem

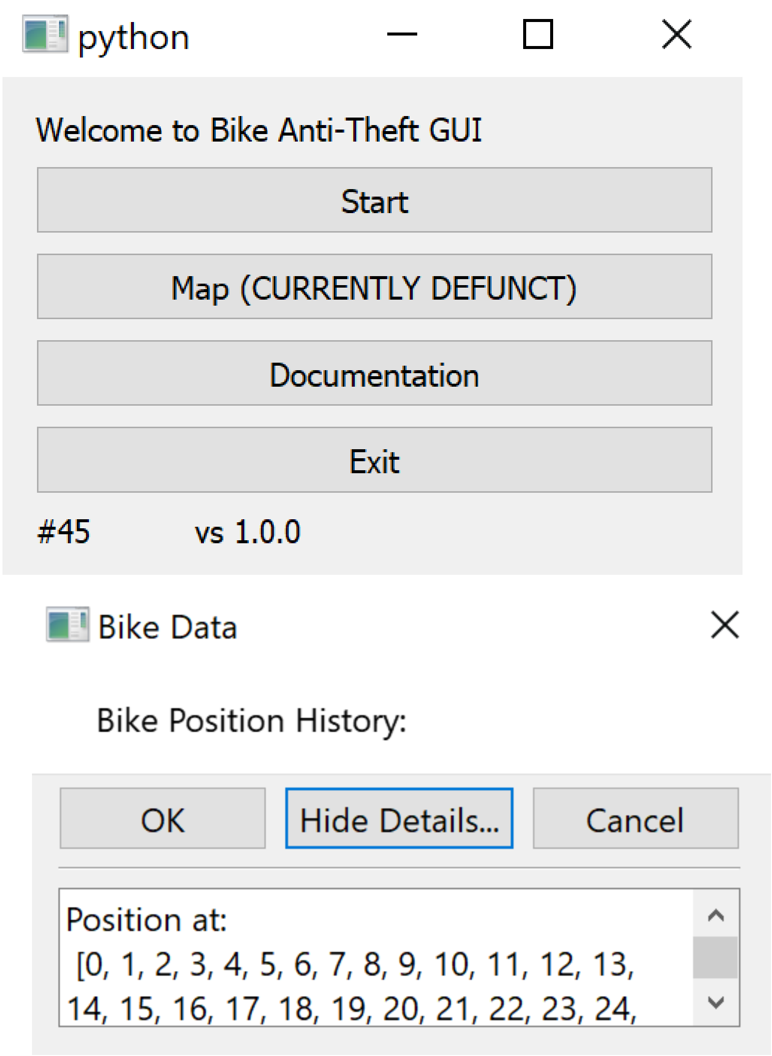


Fig 7: Base Station GUI Screenshots.

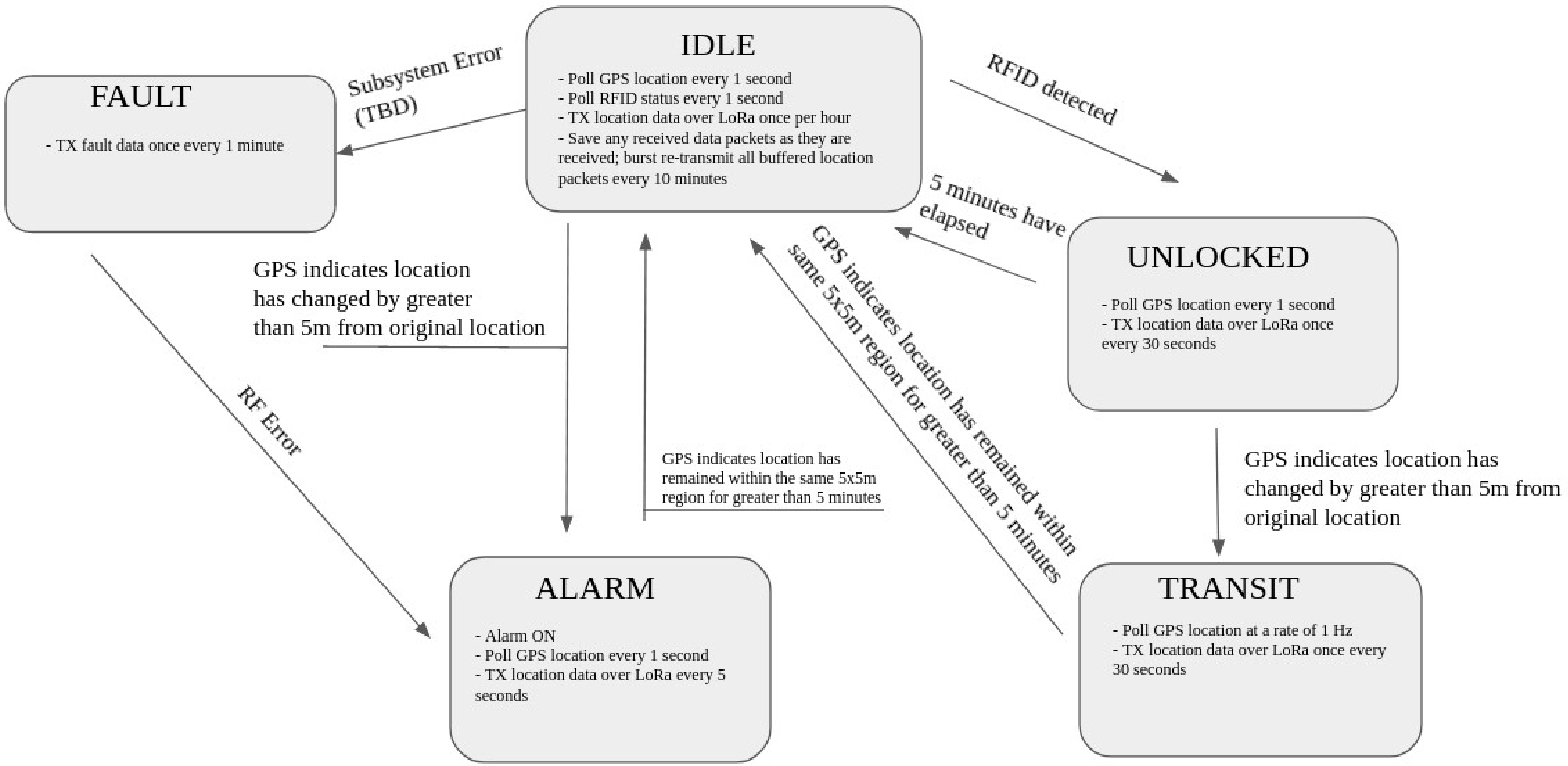
Successes:

- Microcontroller board can communicate with the user laptop over USB.
- GUI displays data from the microcontroller.

Failures:

- As mentioned previously, the LoRa link does not work.
- Mapping feature not fully functional.

Appendix 1: Control System State Machine



Appendix 2: Engineering Explanation of LoRa Subsystem Failure

1. Inadequate Oscillator

In our intial design, we used a 32 MHz external crystal since the RF transceiver in the device requries relatively high frequency accuracy. However, we used the wrong footprint, and due to the short and thin traces it was not possible to bodge in another oscillator. We tested the RF performance in this configuration and saw no output; we queried the error state and found that there was a PLL lock error. Although we had configured the STM32 to use the internal oscillator, we realized that the SubGHz radio peripheral autonomously enables the external crystal regardless of whether the system clock uses the internal or external oscillator. This explains the PLL error, since it essentially meant the radio was not getting a clock at all. We then tried to bodge in a through-hole 32MHz crystal (shown in fig. 1 as the silver cylinder, which did successfully give the device a clock (verified by probing the MCO pin with HSE32 routed to it), but the radio still did not have any output. At this point we queried the error state and found inconsistent results; at times there would be no errors but still no RF output, an at other times there would be PLL errors or frequency lock errors, which made us suspect the issue was option (2) shown at right.

2. Other hardware and/or PA failure

We were eventually able to initialize the radio, query the error state, and see no erorrs, but the radio still did not transmit. Additionally, we verified that our use of the low-level radio driver was correct since we were able to transmit with the Nucleo board using identical code (the only change was a change in frequency from 430 MHz to 915 MHz, since the Nucleo has a 400-MHz band matching network and ours was designed for 915 MHz). Since the radio no longer had any internal errors, and our code worked on known hardware, it's likely that the reason there wasn't any output was due to a hardware issue on our board. We suspect we may have either damaged the internal PA (relatively likely due to the extensive amount of time spent trying different things on the board) or made an error in the external matching network (relatively unlikely since we closely followed the RF design guide, and regardless, an error in the matching network would likely result in somewhat degraded RF performance but not zero RF output at all). To further investigate this issue, we could have populated a fresh board and tested it again, but unfortunately a lack of time and resources prevented us from doing so in this class.

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN