Concealed Bike Anti-Theft Device: Design Document

Elizabeth Atkinson (eatkinso)
Srinidhi Raman (nidhim2)
Alex Wen (acwen2)

April 25, 2022

1 High-Level Requirements

1. If a user tries to remove the bike from a stationary location without the RFID tag, the alarm will sound.
2. The device receives GPS data *at least* once per minute, and records its own position over time.
3. The device transmits its GPS location data and additional data over LoRa to be received by a base station.

2 Requirements and Verification

2.1 Control Subsystem Requirements & Verification

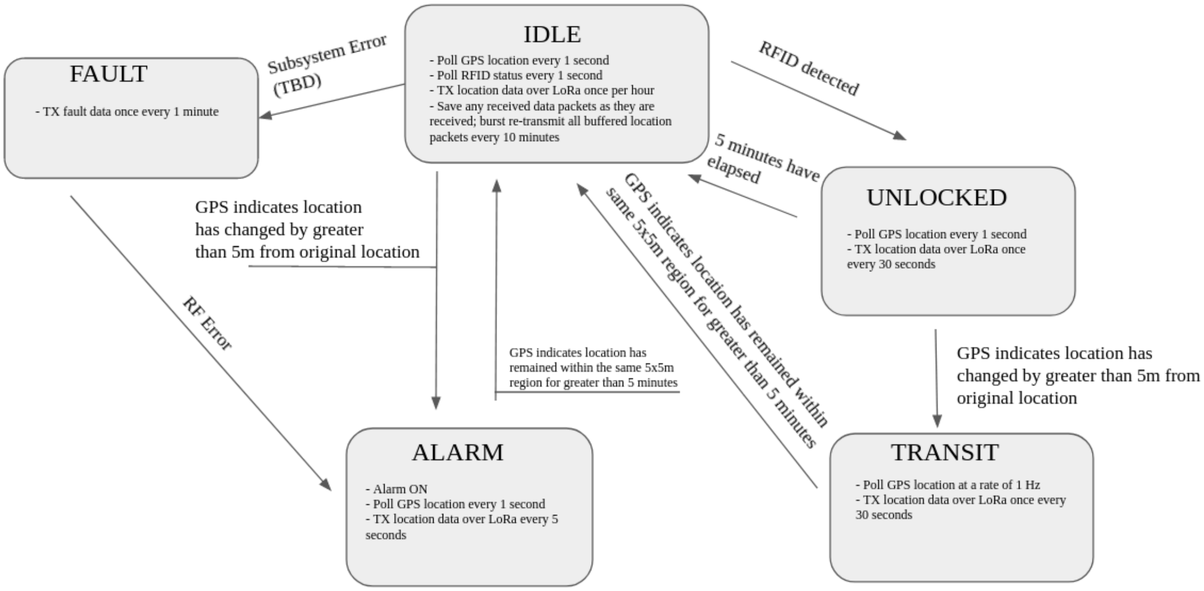
IDLE
- Poll GPS location every 1 second
- Poll RFID status every 1 second
- TX location data over LoRa once per hour
- Save any received data packets as they are received; burst re-transmit all buffered location packets every 10 minutes
FAULT
- TX fault data once every 1 minute
Subsystem Error (TBD)
RFID detected
5 minutes have elapsed
UNLOCKED
- Poll GPS location every 1 second
- TX location data over LoRa once every 30 seconds
GPS indicates location has changed by greater than 5m from original location
GPS indicates location has remained within same 5x5m region for greater than 5 minutes
RF Error
GPS indicates location has changed by greater than 5m from original location
GPS indicates location has remained within the same 5x5m region for greater than 5 minutes
ALARM
- Alarm ON
- Poll GPS location every 1 second
- TX location data over LoRa every 5 seconds
TRANSIT
- Poll GPS location at a rate of 1 Hz
- TX location data over LoRa once every 30 seconds

Figure 1: Control System State Machine

Requirement	Verification
1. The microcontroller implements a state-based control system with power-saving measures in the idle/safe states.	1. Flash the firmware to the device. 2. Monitor the state transitions during device operation.
1. The microcontroller packetizes data to be sent over LoRa at the update rate specified by the state diagram in figure 1.	1. Flash test firmware that cycles through each of the states. 2. Verify that the TX update rate is greater than once per hour in the IDLE, UNLOCK, and TRANSIT states.

Figure 2: Microcontroller R&V Table.

2.2 LoRa Subsystem Requirements and Verification:

Requirement	Verification
1. The LoRa subsystem matching network and antenna have a VSWR of less than 10 at the RF output of the STM32WL55CC.	1. Populate the RF matching network before populating the microcontroller. 2. Calibrate the VNA from 800-1200 MHz with PCB TRL standards. 3. Terminate the SMA output of the PCB with a 50 ohm load. 4. Measure the input impedance looking into the RF matching network at the RF output of the STM32WL55CC. 5. Calculate the VSWR from the measured input impedance and the specified PA output impedance in the STM32 datasheet.
1. The antenna is hidden in a common bike component such as a reflector so that it is not visibly obvious.	1. Objective visual verification. 2. Ask other senior design students.

Figure 3: LoRa Subsystem R&V Table.

Note: The LoRa Subsystem does not function as desired. When the radio is set to the TX state, there is no RF output (measured with a spectrum analyzer). Since identical code *does* successfully transmit CW on the Nucleo development board, this is likely due to a hardware error. We have several possible explanations and future solutions for this failure:

1. **Oscillator:** Our PCB had an error in the oscillator footprint, so we instead soldered a through-

hole crystal oscillator to the pads. However, this introduces significant parasitic capacitance and/or inductances compared to the original SMD oscillator. This could have caused internal errors in the sensitive PLL for the RF circuitry in the STM32WLCC. This hypothesis is supported by the fact that during testing, querying the radio error status sometimes resulted in PLL Lock error. We were unable to fix this error during the course because we did not have the time or resources to order a new PCB when we discovered the error. **Solution**: Order a new board with a properly routed SMD oscillator.

2. **Matching Network:** During the debugging process, we depopulated the RF switch and replaced it with a small piece of wire. Since the length of the wire (approx. 4mm) was much smaller than the wavelength at 915 MHz, we assumed this would slightly degrade the VSWR but probably not cause major issues. It is possible that this change actually significantly effected the VSWR and caused the internal PA output of the radio to be destroyed. Additionally, we did not verify the self-resonant frequency of the capacitors and inductors used in the matching network due to time constraints during the part ordering process. This could also have contributed to degrading the VSWR and damaging the internal PA of the STM32WLCC. **Solution:** Verify that the capacitors and inductors are appropriate for the matching network, and verify the matching network with a VNA before use.

Additionally, we were also unable to verify Requirement 1 due to the fact that the lab does not have a VNA available.

2.3 GPS Subsystem Requirements and Verification:

Requirement	Verification
1. The GPS module can communicate with the microcontroller at a update rate of greater than 1 Hz.	1. Flash a test program to the microcontroller that continuously polls the GPS module. 2. Verify that the GPS module transfers GPS data at an update rate of greater than 1 Hz.

Figure 4: GPS R&V Table.

2.4 RFID Subsystem Requirements & Verification:

Requirement	Verification
1. The RFID detector will detect the user's RFID tag within 10 seconds.	1. Place the RFID tag within 10 cm of the RFID receiver. 2. Monitor the RFID output data to verify it has detected the RFID card within 10 seconds.
1. The RFID subsystem will provide a visual indication that the RFID tag has been detected within 1 second of detecting the RFID tag.	1. Place the RFID tag within 10 cm of the RFID receiver. 2. Monitor the RFID reader output data and the RFID LED to verify that it turns on within 10 seconds of detecting the RFID tag.
1. The RFID subsystem triggers the alarm if movement occurs and the user's RFID is not detected.	1. Ensure that the RFID tag is out of range (> 1 m). 2. Move the bike 5m from the original location to verify the alarm is triggered.
1. The RFID indicator LED is concealed in an unobtrusive location on the bike.	1. Visual verification.

Figure 5: RFID Subsystem R&V Table.

2.5 Power Subsystem Requirements & Verification:

Requirement	Verification
1. The power system will provide 3V3 +/- 0.4 mV to all subsystems that require 3V3, and 5V to the subsystems that require 5V.	1. A multimeter will be used to ensure the voltage and current are as specified at different points on the board.
1. The power system will power the device for over 24 hours.	1. Connect a multimeter/oscilloscope to measure the voltage at the battery terminals. 2. Record data every two hours over two 12 hour periods. 3. The data from the device will be read to ensure that the values were reasonably constant over time.
1. The microcontroller can fully power off the GPS and RFID subsystems when not in use by turning off the load switches.	1. Flash a test program to the micrcontroller that toggles the control signals GPS_ PWR and RFID_ PWR. 2. Measure the output voltage on pin 1 of U7 and U8 to verify that the RFID and GPS subsystems are powered off.

Figure 6: Power Subsystem R&V Table.

Note: Since the GPS requires some time to lock, we have not implemented the power-saving feature in the Demo firmware. However, we did verify in the lab that the load switches turn off power to the devices.

2.6 Alarm Subsystem Requirements & Verification:

Requirement	Verification
1. The alarm makes noise when triggered by the ALARM_CTRL signal.	1. Flash a test program to the microcontroller that toggles the ALARM_CTRL signal. 2. Verify that when ALARM_CTRL=1, the alarm makes noise.
1. The alarm circuit draws < 1 uA of current when not making noise.	1. Set ALARM_CTRL to 0 through code. 2. Use multimeter to measure voltage and current value from the alarm.

Figure 7: Alarm Subsystem R&V Table.

Note: ALARM_CTRL is not a signal that is 1 or 0 but instead a PWM output that is either on or off.

2.7 Base Station Subsystem Requirements & Verification:

Requirement	Verification
1. The receiver board will interface with the computer over USB.	1. Flash a test program to the base station board that sends continuous data over the I2C/USB interface. 2. Connect the base station board to the user computer and view the data sent over USB. 3. Verify the received data matches the sent data.
1. The receiver board will receive LoRa packets from the bike-based device.	1. Flash a test program to the TX (bike system) board that transmits continuous data. 2. Flash a test program to the RX (base station) board that listens for data and continuously sends received data to the computer over I2C (USB to the computer). 3. Verify the data sent from the TX board is received by the RX board.
1. The GUI will display GPS data over the past day, week, month, or year on a map.	1. Flash a test program to the TX (bike system) board that transmits data with (false) timestamps that cover a 6-month period. 2. Transmit the data and receive it with the base station board. 3. Using the GUI, plot the data and verify the output plots match the data sent by the TX board.
1. The GUI will allow the user to view plots of the average speed and distance traveled.	1. Flash a test program to the TX (bike system) board that transmits data with (false) timestamps that cover a 6-month period. 2. Transmit the data and receive it with the base station board. 3. Using the GUI, plot the average speed and distance traveled and verify that it matches the data transmitted by the bike system board.

Figure 8: Base Station Subsystem R&V Table.