Real-Time Asset Tracking System

Faculty Advisor: Professor Ridha Kamoua
Team: Ayman Azad, Muhammed Fareed, and Alwin Joseph
Electrical and Computer Engineering Department, Stony Brook University

Stony Brook University
Electrical and Computer Engineering

RealTrak
Real Time Cloud-Based Asset Tracking System

Abstract

According to the FCC, 44% of reported stolen devices or belongings occur because the owner leaves the device behind in a public setting. The RTS team wanted to solve a simple, yet difficult problem faced by many people today: how to keep track of one's belongings? Using a form of asset tracking prevents loss and/or theft of valuable personal belongings and expensive assets (i.e. Laptop, tablet, wallet, keys, etc.). From reminding you that you are forgetting your keys at home to alerting you when a pickpocket is stealing your wallet, the RTS System helps to mitigate loss of personal belongings, individuals, and high-value assets.

Solution Description

The project consists of three main components that connect with each other to accomplish the overall goal of protecting and tracking valuable assets. Each component uses the Bluetooth Low Energy standard for wireless close-range communication. The architecture used provides the reliability and precision of a military-spec tracking system but is portable & cost effective like consumer offerings.

The RTS System can support tracking for up to 5-7 user defined assets simultaneously and alert the user if any are missing. Each asset has a Transmitter node attached to it (with its own specific device ID) that will perform a 'handshake' with the Receiver module on a software defined interval. The asset will be able to roam freely from the Receiver module up to approximately 15 to 25 feet (roughly the distance between two corners of a standard bedroom) before alerting the user that it is missing. The user can see where their assets are in real using the mobile application

RTS System Architecture

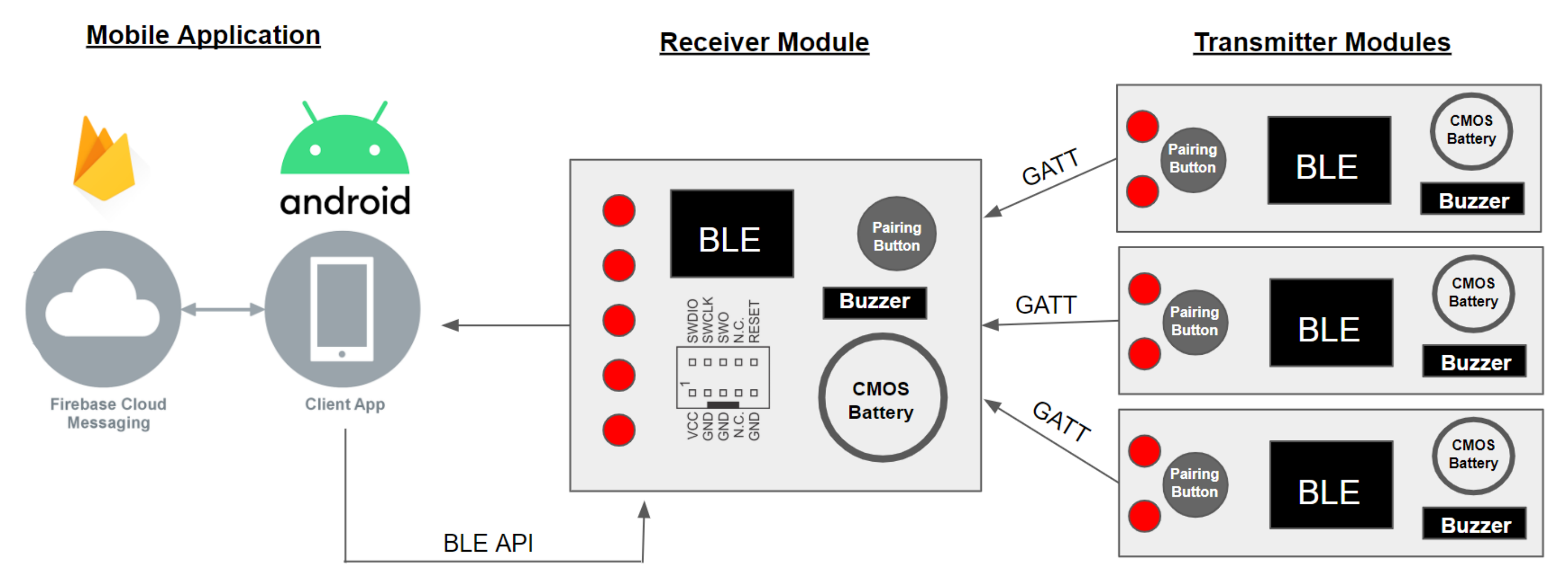
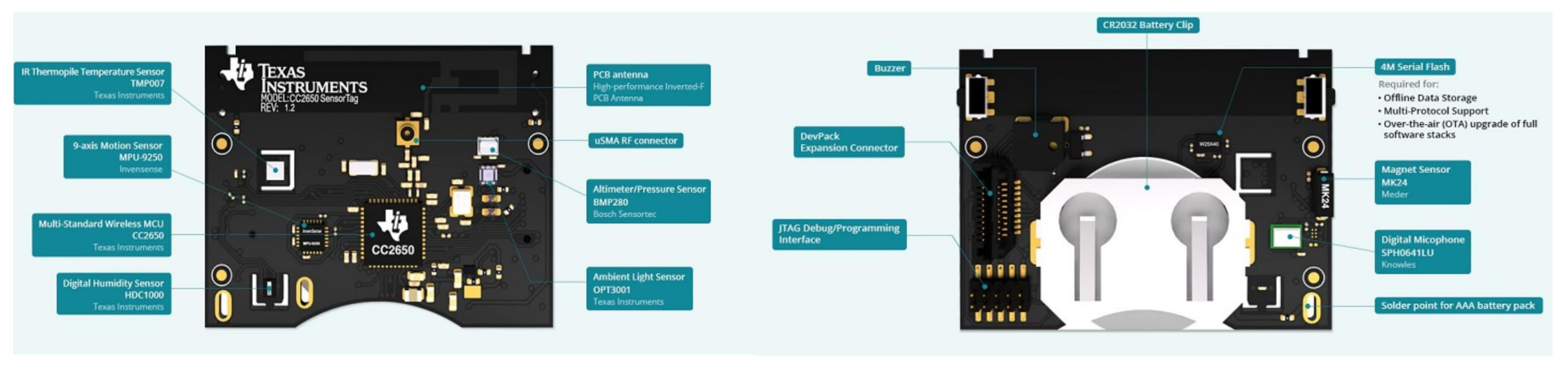


Figure shows the high-level overview of the RTS System Architecture & Components

Transmitter Modules

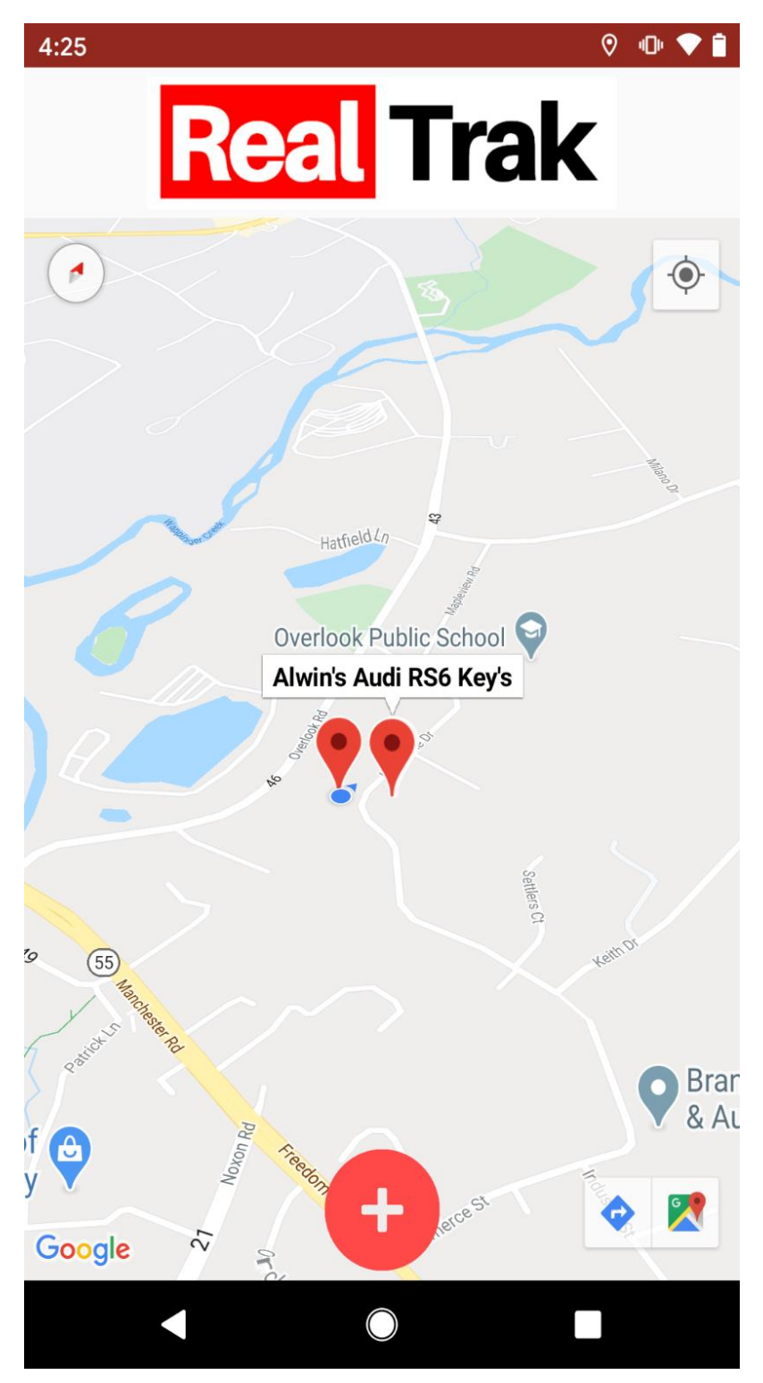
The Transmitter is very thin and portable; it attaches to assets (keys, wallet, medicine bottles, etc.) and transmits a 'handshake' to the Receiver board. The 'handshake' is a custom protocol developed by RTS to send metadata securely via BLE GATT Layers. If the Transmitter doesn't see a handshake back, an audible alarm is set to alert the user. The firmware was written in embedded C & the hardware uses the CC2541 BLE SoC from Texas Instruments. The board can operate for approximately 1495 days of continuous operation thanks to the low-power SoC.

Receiver Module



The Receiver module utilizes the low power CC2650 BLE SoC from Texas Instruments. It will be housed in a 3D-printed low-profile case that sticks to the back of a mobile smartphone. It receives BLE signals from multiple Transmitter nodes, stores characteristics from the nodes in a table, and broadcasts this table to Mobile devices. This adds redundancy and reliability. The Transmitter nodes will connect to the Receiver (central mode) and at the same time the Receiver will also connect to the mobile app (as a peripheral). The Receiver's firmware is developed in C.

Mobile Application



The mobile application acts as the main interface for the end user to manage tracking their assets and provides essential redundancy for the system. The application is built for the Android OS Platform using Java. The Android app allows users to create an account and add their devices to a tracking list. The app will utilize the Google Maps API from GCP to show users where their assets are on a Map interface based on data from the Receiver. The user will connect the Receiver to the app via BLE and connect Transmitters to the Receiver.

Results & Conclusion

Overall, the team was able to surpass many challenges that were faced throughout the year and successfully finish the software aspect of the project. However, due to the COVID-19 pandemic, the team was unfortunately unable to test the integration of software components with the hardware modules. Since BLE requires close-range communication between devices, testing and debugging the project as designed would require that all three team members be present together with the hardware components. Instead, the team decided to verify that each component would work as intended by developing minor testbenches.

As a result, the RTS team was successfully able to test the connection of the CC2541 Transmitter nodes and test the functionality of the app using Bluetooth-enabled devices such as headphones and smartwatches in place of the custom hardware components the team designed. The team learned invaluable skills regarding product development, the BLE Specification, mobile application development, and embedded systems design.

Glossary

- **API** – Application Programmer Interface is a software utility.
- **BLE** – refers to Bluetooth Low Energy Standard and Protocol.
- **Central Mode** – an operating mode in BLE to Receive data.
- **Characteristics** – data divided into declaration and value. Using permission properties (read, write, notify, indicate) to get a value.
- **Firebase** – Realtime Database service within Google Cloud.
- **GATT Layers** - Generic Attribute Profiles: Transmits the data that will be accessible to other connected devices an attribute table.
- **GCP** – Google Cloud Platform. Provides hosting & data analytics.
- **Peripheral Mode** – an operating mode in BLE to Transmit data.
- **SoC** – System on Chip (refers to the antenna & microcontroller)

Acknowledgment

The RTS group would like to express our deep gratitude to **Professor Ridha Kamoua** for his continued guidance and mentorship throughout the entire project as well as **Professor Harbans Dhadwal** for his coordination of the Senior Design program, and the **IEEE Student Chapter** for use of Lab facilities and equipment. Lastly, the RTS group would like to thank the **ESE/ECE Undergraduate Faculty and Staff** for their instruction, support, and mentorship over the course of the past four years.

RealTrack Systems (RTS)