Write Firmware for a BLE Module

Project Details
January 2019

The purpose of this project is to build the firmware and Android application needed to achieve persistent pairing and several pieces of user functionality.

The project currently revolves around a Silicon Labs BGM123 SiC. We are open to switching to a Nordic nRF52810 SoC model if it proves more efficient for battery use and form factor without decreasing necessary Bluetooth performance.

A copy of our PCBA design will be shared.

1. Project Overview

We are building an embedded device for drivers. The device latches onto the steering wheel and lights up to alert the driver when they need to slow down to avoid a speed trap.

This project, Write Firmware for a BLE Module, focuses on the embedded devices firmware and an Android application that guides the user through and maintains the persistent pairing process.

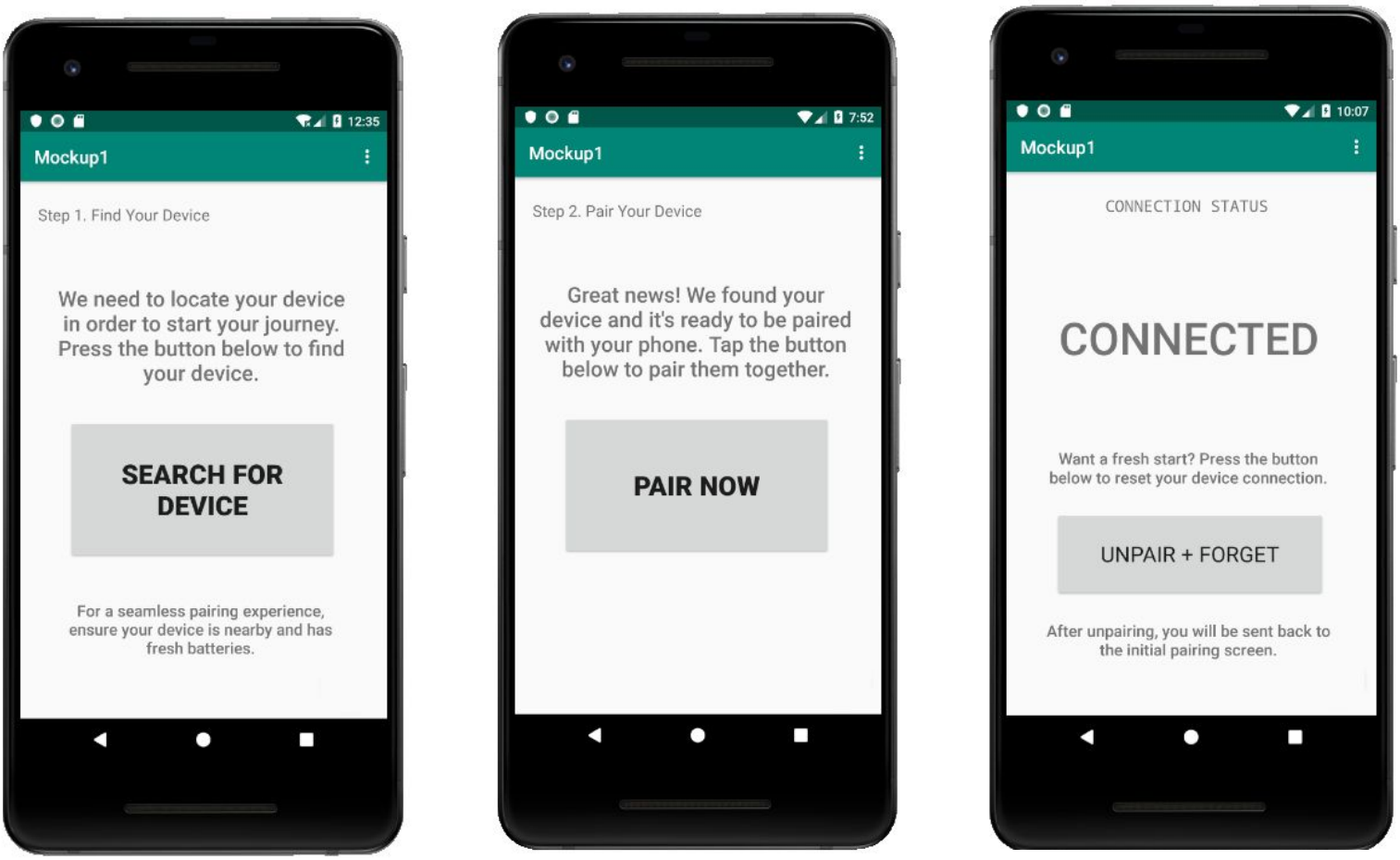
- Insert picture

2. Android Application

The purpose of the Android application is to guide the user through the persistent pairing process between the embedded device and the user's Android smartphone.

2.1 Mockup

The following mockup shows the flow of actions for the pairing process.



The desire is to have the app recognize one of our devices using a very unique identification method.

If the embedded device cannot be discovered by the app, then a simple toast message saying “Could not find device.” should pop up for 20 seconds.

2.2 Activation

We want to be good stewards of the user’s smartphone data and battery life. So, we want the application to go to “sleep” whenever the user is not driving their car. This sleep state would entail:

- Not using any GPS functionality

- Waking up every 90 seconds to see if the smartphone's accelerometer is over 10 miles per hour. If it is not, then continue to "sleep".

If the phone's accelerometer surpasses 10 miles per hour or if the user directly opens the app, then the app needs to try to ping the embedded device to see if it is nearby. It should keep trying to ping the device until:

- It's attempted 50 times
- The accelerometer goes below 10 miles per hour for more than 1 minute.

3. BLE Module Firmware

We are currently using a Silicon Labs BGM123 SiP as our BLE module. The firmware for this project is divided between two pools of the functionality, Pairing and User Functionality.

3.1 Pairing: Pair and stay paired

Identification: We need a way for the embedded device to be recognized as a unique product a part of our brand. We need a highly unique identifier assigned to our module so that the app can recognize it as one of our devices.

This goes for the app side too. We don't want our device to pair with just any old app - We want it to pair with only our app. If it can be done in 2-3 hours of work, then this should be pursued.

Initial Pairing: As soon as the user pops the batteries into our device, the device should be available for pairing, checking for pair requests every 5 seconds for 10 minutes. If no pairing has occurred, then the device should go into a deep sleep, checking for pair requests once a minute with very low millisecond cycles.

Persistence: We are very battery-constrained, so our device should be in a deep sleep state for 90 seconds at a time, waking up to see if the app is nearby and active. If it is, wake up and check for instructions from the app every 5 seconds. We want to balance message success accuracy and message reception time (low milliseconds).

Going to Deep Sleep: The app will send a goodbye message to our device to tell it to go into deep sleep.

3.2 User Functionality

LED Activation: When the app sends a “GO HIGH” message, our device’s LED should light up. It should fade in from dark gradually and remain lit until the app sends a “GO LOW” message.

Our device uses an RGB LED, documentation here.

We want the device to light up a dark magenta color, with **#e00437** as a target.

Button Press: When the user presses down on the device button for 3 seconds:

- The device should send the app a message saying the button was just activated.
- The LED should “breathe” out and in two times like this animation. Each “breath” should be 2 seconds from start to finish (4 seconds total).

4. Deliverables

- Android Studio package containing all project files necessary for achieving all of the functionality detailed in this document. The client is able to successfully import the package into Android Studio and run the app on an Android device.
- Firmware to be installed on a BGM123 that performs all of the functions detailed in this document.
- High-level documentation explaining what has been delivered and how it works on a basic level. Comments inside the code are appreciated.