SkatEE

Group members: Andrew Baker, Santiago Neira, Pierce Witmer, Hunt Wyman

Concept

The central idea behind this project is to build a product that has the sharing functionality of LimeBike (a popular bicycle sharing system) and the advantages of riding an electric skateboard, which include a more enjoyable riding experience without sacrificing transportation speed. Cellular technology is ideal for IoT integration in the LimeBike concept, because it allows internet connection that covers the operating range of the skateboard. Since there are companies that create programmable electronic devices that have both cellular capabilities and cloud platforms for data management, starting with an existing product such as the Particle E-Series allows us to implement our vision.

Design Requirements

1) Create a mobile app for user interface
2) Use cellular technology to send and receive data from the application to our skateboard
3) Build a circuit which responds to button pushes on the app
4) Creating a locking mechanism controlled by the circuit that secures the board when not in use

Results

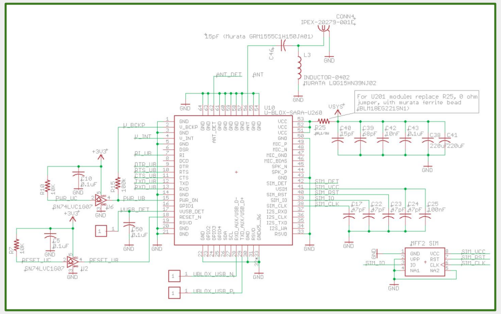
The overall success of this project relies heavily on a few key functionalities. First, the skateboard needs to accommodate the holster and electronics in a compact, efficient way. Although the current design is compact, the efficiency could be improved using custom electronics. Next, the iOS app has to have a way of communicating information to control the servo the right amount. CocoaPods (a dependency manager for Swift) and pulse width modulation directly solve this problem. Both of these solutions were implemented effectively. Lastly, the locking mechanism needs to mechanically allow and disallow access to the holster. The current design is functional, so this functionality also meets the requirements.

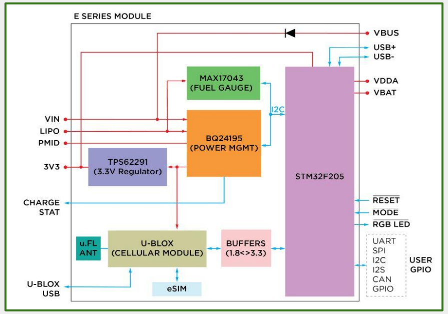
Future Goals

The most significant development to this project would be the complete integration of all components. For example, designing a new electric skateboard includes a custom circuit board integrated with the controls, battery, and charging of an electric skateboard. An exciting challenge for future students to work on would be discovering a method of charging the skateboard battery without relying on a charging station.

ublox Cell Module

The ublox Cell Module handles the telecommunication protocol for the Particle E-Series. An attached Taoglas antenna receives the HSPA/GSM signals and the module talks to the microcontroller over a full-duplex USART interface. The eSIM chip is directly connected to the u-blox. The power to the eSIM chip is also provided by the cell module.



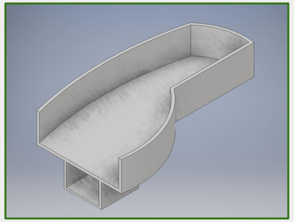


Particle Board

The particle E-Series board is used for testing and development of the controls and app. It also serves as a guideline for researching creating a custom board, as well as provides an antenna and lipo battery.

Assembly

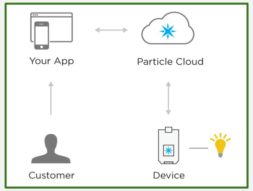
We have many different objects that needed to be in close proximity in order to function. The solution to combining these various components is a 3D-printed holster that holds the remote and servo lock combination as well as allowing for wiring. The holster and other components are attached via epoxy to avoid damaging the skateboards internal wiring. The particle board itself is lofted on stand-offs to prevent warping or snapping of layered connections. The Particle E-Series includes microcontroller and an antenna, so it can receive functions and store data. The servo is controlled from the Particle E-Series, both are powered by an external 3.7V lipo battery.



Deeper Dive into the App

To meet the design requirements set for this project, the team created the skatEE app, an application that ultimately serves as the user interface between the user and skateboard during a normal "ride interaction". Below are three main functionalities that users can perform when communicating with the the skateboard.

- Lock/Unlock:
  - *Use servo motor to toggle states*
- Read Board Status:
  - *Determine status based on stored variables*
- Locate:
  - *Triangulate position based on coordinates*



Software Architecture

The interactions between the skatEE app and the skateboard are possible through the use of the Particle E-Series development board and the Particle Cloud Console. The application itself was built using Apple's development platform, Xcode, whereas the functions that interact with the E-Series board were written using the Particle Cloud Web IDE. The two interfaces are then connected using the Particle-iOS SDK (software development kit). This allows function calls to be invoked from the skatEE App, transferred via the 3G cellular network, and finally received and performed by the E-Series development board which is mounted on the skateboard.

Google Maps Integration

To implement the location function, the skatEE App makes use of the Google Maps API. This process required development and integration on both the Particle Console as well as the iOS skatEE App. The Particle console is responsible for using the Google Geolocation API to triangulate the E-Series board's position using nearby-cell tower connections. From this, the latitude and longitude are returned and stored within the skatEE App. On the other side of the software, the skatEE app uses the Google Maps SDK to return a map depicting the position of the located device.

Storage and Billing

As of now, the application stores all of the key information on the Particle Console. This includes variables such as board status and location, but in the future could be expanded to ride duration, distance traveled, and other details that are necessary to calculate the cost of each ride. Ideally, once all of the data parameters are set, they would be stored on a real third party database (such as MongoDB or Amazon's AWS) so that the application could offer secure information storage and reliable retrieval to its users. Although the Particle platform already boasts a billing manager, a third party database would also provide a more effective solution especially if the project scales up.