Electrical Engineering Lab - Embedded System
Final Project Report

EE4 **Yun-Yao Tien** EE4 **Shu-Yu Lin**

1. Motivation

Bicycles are the main transportation for students in NTU, there exists some potential problems, thus we want to design an embedded system to solve

a. It is difficult to find your own bicycle in large parking lots if you forget where your bicycle is.
b. One’s hands are often contaminated with lubricating oil or mud while unlocking the bicycle.
c. It is often being stolen, and it’ll be too late when one notices it.
d. There are lots of bad riders, and they often cause accidents.
e. There are speed limits in NTU, which is 20 km/hr, we should follow the rules.

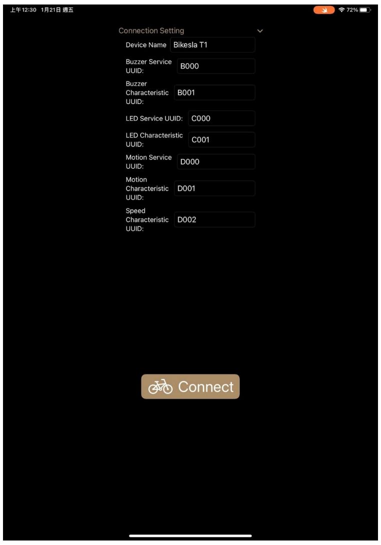
2. Method

We divide this part to categories of functions

a. Speed-checking: Using the Hall sensor, with the magnets on the wheel, we can calculate the speed of the spinning of the wheel and inference the speed of the bicycle.
   The method of calculating the speed of the bicycle is:
   The perimeter of the wheel / the time difference between the two times the magnet passes through the Hall sensor.
b. Lock/Unlock: Using the App, we can unlock and lock the bicycle.
c. Bicycle-finding:
   i. Using the App, pressing the Flash button to make the LED flicker. It is convenient for finding our bicycle at night.
   ii. Press the Ring button to make the buzzer ring for 2 seconds, users can locate where the bicycle is much easily with this function.
   iii. There is an additional drop down menu for users to adjust the lightness and the frequency of the LED.
d. Speeding detection: Based on the speed limit of Palm Avenue, we set the buzzer to ring for 2 seconds if the speed exceeds 20 km/hr.
e. Theft alarm: If the bicycle is locked, with the Hall sensor on the wheel, if the sensor detects that the wheel spins over some certain criteria, we regard it as abnormal usage, the buzzer will ring continuously.
f. Fall detection: Using the intrinsic triaxial accelerometer of the STM32 IoT Node, if we detect the z-axis acceleration is less than 500 m/s, we regard it as someone crashed, the buzzer will ring continuously, informing the people nearby.

3. Results

a. The page before the App connects to the bicycle.



上午12:30 1月21日 週五

72%

Connection Setting

Device Name: Bikesla T1

Buzzer Service UUID: B000

Buzzer Characteristic UUID: B001

LED Service UUID: C000

LED Characteristic UUID: C001

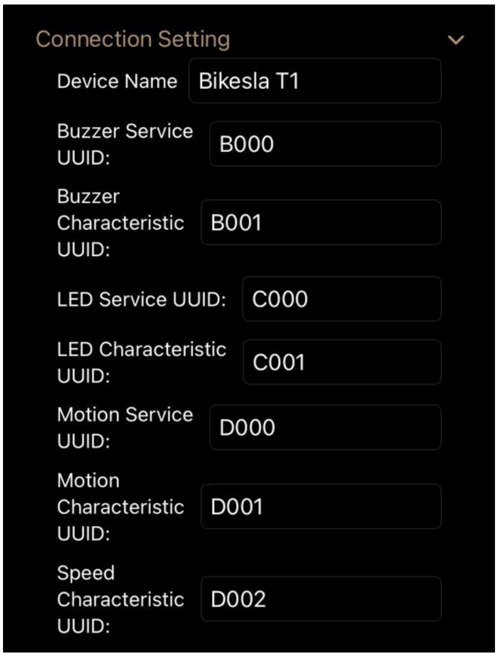
Motion Service UUID: D000

Motion Characteristic UUID: D001

Speed Characteristic UUID: D002

   i. drop down menu, one can check

      1. the default name of the STM32 IoT Node.
      2. Service UUID of Buzzer, LED, Motion



Connection Setting

Device Name: Bikesla T1

Buzzer Service UUID: B000

Buzzer Characteristic UUID: B001

LED Service UUID: C000

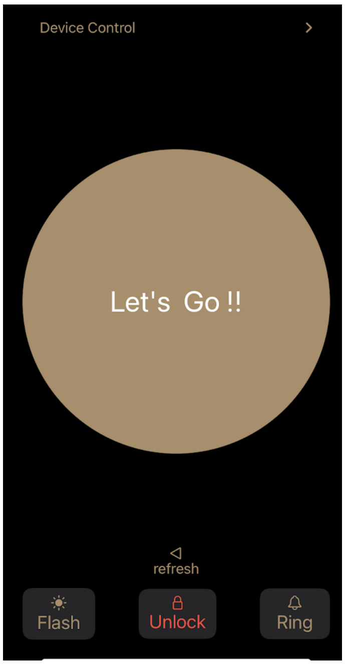
LED Characteristic UUID: C001

Motion Service UUID: D000

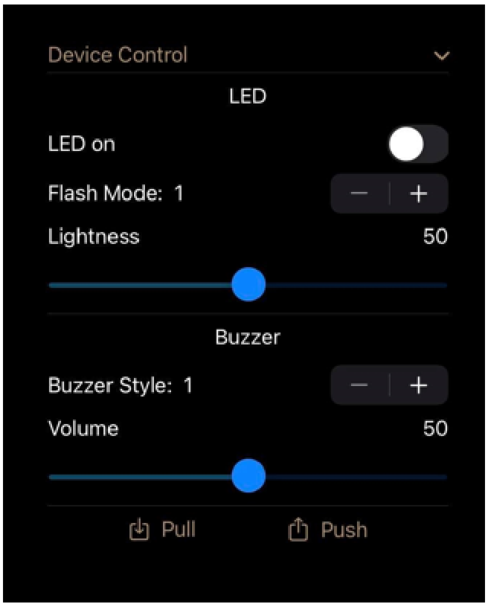
Motion Characteristic UUID: D001

Speed Characteristic UUID: D002

b. The page after pressing Connect and successfully connecting to the bicycle.



i. Drop down menu, one can control ON/OFF, lightness, frequency of the LED.



LED on

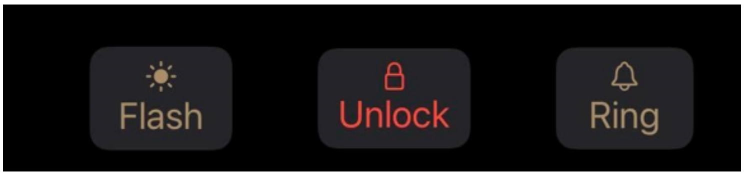
Flash Mode: 1

Lightness 50

Buzzer Style: 1

Volume 50

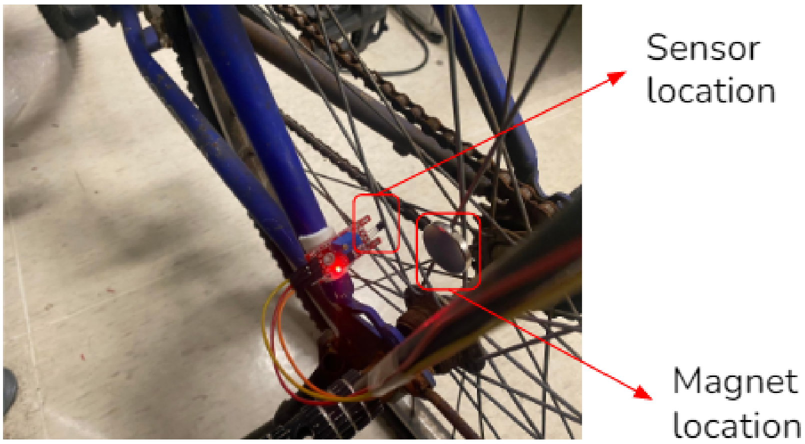
ii. Bicycle-finding toolbar.



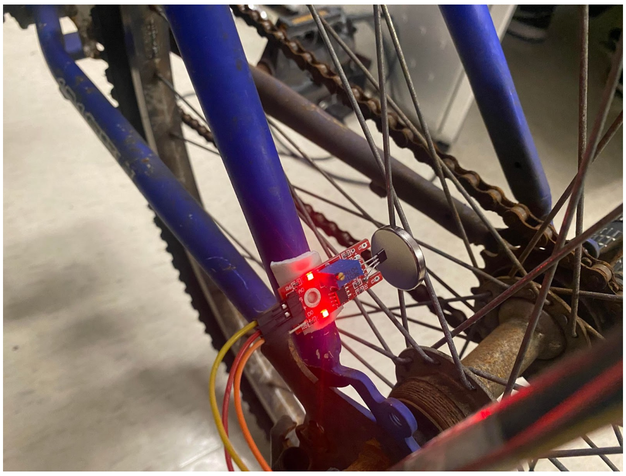
1. Flash: Pressing it to make the LED flicker.
2. Unlock: Pressing it to lock/unlock the bicycle.
3. Ring: Pressing it to make the buzzer ring for 2 seconds.

c. The configuration diagram of the Hall sensor and the magnet.

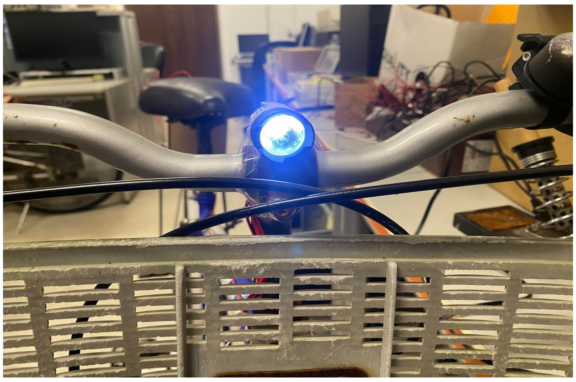
   i. The magnet is not in the region which the Hall sensor can sense.



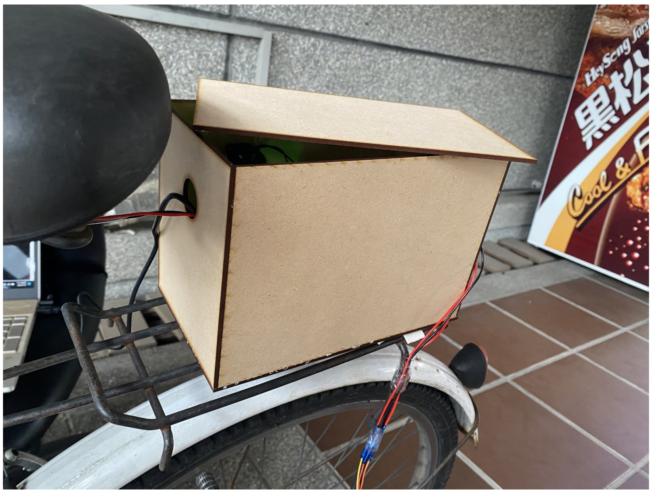
   ii. The magnet enters the region which the Hall sensor can sense.



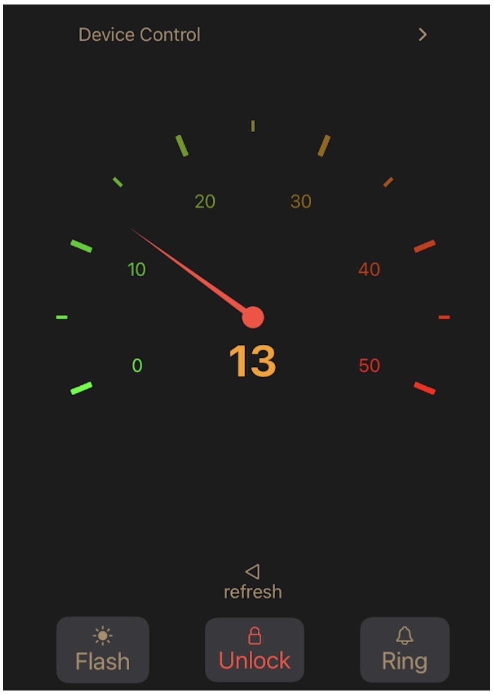
d. Using LED lights and one lens surface of a flashlight to compose a warning light.



e. Using the laser cutting machine in NTUEE Maker Space, we design our own container to put the STM32 IoT Node, circuits, and breadboard in it.



f. The speed of the bicycle is shown on the App.



4. Problems and Discussion

Power supply problem:

Since we wish that the embedded system we designed can work on a bicycle independently, we bought a small power bank, whose output is 5V/1.2A. The power bank is connected to the USB port of STM32, the power of other devices including LED, buzzer and Hall sensor are provided by STM32.

When we are testing, we encounter a strange problem: when all the devices are connected, the BLE service in our program cannot start successfully. After trial-and-error, we found that the reason might be the required voltage of the buzzer is larger than expected, therefore, we remove the ground line of the buzzer before we press the reset button. After pressing the reset button and ensuring that the BLE service starts successfully, then we'll put the ground line of the buzzer back, now that all the function works normally. During the demo, professor Wang gave us some advice, informing us that we can use relays to make our system more stable.

5. **Demo video link:**
    a. Theft alarm:Link
    b. Fall detection:Link
    c. Bicycle-finding:Link
6. **Reference**
    a. 指紋辨識模組:
    https://www.waveshare.net/wiki/Capacitive_Fingerprint_Reader
    b. STM32Cube and include files:
    https://community.st.com/s/question/0D50X00009XkXw4SAF/stm32cube-and-include-files
    c. 嵌入式12——STM32时钟系统讲解(此为STM32L4):
    https://zhuanlan.zhihu.com/p/91225332
    d. How to migrate STM32Cube based application between two STM32 series:
    https://www.youtube.com/watch?v=YU4jivUZJm8
    e. Swift Document
    https://developer.apple.com/documentation/swift
    f. Swift package for BLE
    https://github.com/jordanebelanger/SwiftyBluetooth
7. **The GitHub link of the project:** Link