BikeTracker

Genevieve Heidkamp, Alison O’Connor, and Marisa Thompson

December 10, 2016

**Table of Contents**

[1 Introduction](#h.2ep9c74x4mfr) 2

[2 Problem Statement and Proposed Solution](#h.h9tppp2s5pxu) 2

[3 System Requirements](#h.lxea93vwywyz) 3

[4 System Block Diagram](#h.nrhdkl6rzcoy) 3

[4.1 Overall System:](#h.i7m3n0e77fow)

[4.2 Subsystem and Interface Requirements:](#h.a07cw1h9j4ri)

[4.3 Future Enhancement Requirements](#h.48urx5bxaab1)

[5 WiFi considerations](#h.xsnkrku50qu9) 7

[5.2 Data Flow](#h.tlfr3zke2uxa)

[5.3 ESP8266](#h.si2i1jg47vc1)

[5.4 Protocol for Data Exchange](#h.ywtluohz1tv2)

[6 High Level Design Decisions](#h.3475dynrnuzv) 8

[7 Open Questions](#h.g1haoydpe311) 10

[8 Major Component Costs](#h.vt16fp4fra7d) 10

[9 Conclusions](#h.1i1ku6ofio0f) 10

# **1 Introduction**

On college campuses across the nation, students are either constantly preoccupied with the location of their bikes or are hesitant to buy a bike in the first place. Why? Because of the high likelihood that their bikes could be stolen, even while using bike locks or storing their bikes in inconspicuous locations. In order to eradicate these fears, it would be beneficial to many college students if a GPS device could attach to their bike to help them locate their bike in case of theft. Not only would this help them find their stolen bike, but it could also alert them at the time of the robbery so they could take action if desired. This concern led to the creation of BikeTracker.

The following is a document which outlines the high level design of this BikeTracker including the many design specifications we needed to meet in order to bring a functional product to fruition. It outline the problem statement, our proposed solution, the overall system requirements divided into subsystem requirements, and some open ended questions that we still have about the project.

# **2 Problem Statement and Proposed Solution**

In the United States, it is estimated that between 800,000 and two million bicycles are stolen each year. Many of these thefts occur because cyclists fail to lock their bike or use bike locks that are broken easily. In addition, once a bike is stolen, it is generally difficult to recover, especially in a city or on a college campus, because bikes can be transported so quickly and/or can easily blend in with other bicycles on a bike rack. Only about 5% of bicycles that are stolen are returned to their owners. [[1]](#footnote-0)

In order to prevent bicycle theft and help more bikes be returned to their owners, we are proposing to make a smart bicycle finder. This product we are proposing is specifically targeted towards college campuses where wifi is prevalent and will be able to utilize a GPS signal.

First, the bike finder would include a GPS device that, when connected with a smartphone, could inform a user where their bike is located at any given time. This feature could also be used to track where the bike has been in order to track the most recent taken paths and whereabouts of it.

Finally, a battery will be used to power the device. A battery will be used because in order to keep the tracker in a discrete location on the bike, a smaller / more compact power source will need to be used, and a battery would fit this requirement.

# **3 System Requirements**

The following is a list of system requirements necessary for our BikeTracker to have full functionality:

* Device will interpret GPS data through NMEA sentences that are parsed and then sent through ESP device to mobile application by way of a WIFI network
* Device will be powered by a rechargeable battery.
* Device will need to support GPS, Battery, and ESP8266 components in order to function and communicate properly. The range of the device is to be approximately the size of a large college campus while communicating with open wifi networks in the area.
* The user interface must be a mobile application which shows the GPS location of the tracker at the last time a measurement was taken. This application will contain modes so that if the location of the bike changes while in “parked” mode, an alert is sent through the application to notify the user.
* To connect the specific BikeTracker to the user in mobile application, there will be a setup option in which the device will connect to the user application through MQTT protocol.
* Device will be installed under the bike seat using a screw apparatus while being mounted behind a reflector panel for disguise. Once the battery is connected, the device will be functional.
* Device will need to be an appropriate size and weight to successfully be hidden behind a reflector panel under the bike seat.

# **4 System Block Diagram**

## **4.1 Overall System:**

The system requirements must be met by the system as a whole. These requirements will be met through 4 subsystems: GPS location tracking, user app interface, WiFi capability, and battery power.



 These four subsystems will divide the project into manageable segments that will allow for a gradual development of the project with consideration to testing phases until the product is fully functional.

## **4.2 Subsystem and Interface Requirements:**

The requirements of each subsystem or major interface are described below by subsystem. These lower level requirements support the overall system requirements.



The GPS location tracking will utilize the GPS device to connect with the microprocessor in order to transmit the NMEA sentences to be converted and used in the User Application. The NMEA sentences will be parsed in order to obtain time, latitude, and longitude which will be passed into the ESP for transmission to the user interface.



The mobile app will be an iOS app. It will have two different modes: “park” mode and “ride” mode. If the bicycle moves during “park” mode, the app will send an alert to the user. The app will not send an update while in the ride mode. It will have the ability to display the location of the device and the time at which the location information was taken. The elements will comprise the user mobile interface.



The WiFi Capabilities of this device will rely upon the ESP8266 to be connect to the microprocessor while being able to utilize the MQTT protocol developed on Amazon server in order to communicate with the User Application. Power from the battery will drive the device and allow for the sleep/wake modes to occur on 5 minutes intervals. The parsed data from the GPS will be sent through this capability for full functionality of the application.



The battery in the system will be rechargeable so the user does not need to continually purchase batteries to power the system. It will connect to the system, and converters and regulators will provide the appropriate voltages to the different devices on the board.

## **4.3 Future Enhancement Requirements**

There are a number of features that are not going to be part of the initial release of our product, but we would be interested in adding them. There are several enhancements we think could be added in the future:

1. The first future enhancement for this project would be to add a feature that would allow the user to call the device at will and get a real-time location right then. This would require a different consideration of the battery usage, since the device would have to be on for a longer period of time.
2. A second future enhancement would be to add an accelerometer to the device. This could be included with the alert feature of the app. If the bike began to move in the “parked” mode, the motion of the accelerometer could be the alerting feature rather than the change of GPS location. Movement of the bike could be detected sooner.

# **5 WiFi considerations**

**5.1 Connection Considerations**

This device will be connected to a single WiFi network, utilizing the open guest networks on college campuses and an MQTT protocol developed in the Amazon server or through the Nomad network.

## **5.2 Data Flow**

Information is obtained in the form of NMEA sentences from the GPS device. This information is 8 bits. A measurement is taken once every five minutes. These sentences are transmitted to the microcontroller which then parses the information. Desired information from the GGA sentence of time, latitude, and longitude is obtained and sent through WiFi capability to the user application. The user application obtains the location and shows the user where the bike is.



##

## **5.3 ESP8266**

Our design will not require a separate microcontroller because the ESP8266 microcontroller has the capability to time, go into a low power mode, and it can transmit and receive the information necessary to communicate the GPS location through its own serial interface. The only consideration that will have to be taken into account while programming is that the device resets whenever it powers up after deep sleep, so most of the programming will most likely be in the setup function.

We will use the ESP 12 model of the ESP8266 device. It will be the easiest to use for our particular project, based on how many I/O pins there are and the method in which those pins are brought out to usable locations.

We will be using the Arduino environment for programming the device. We’ll use this environment because it contains the libraries for the ESP device and is set up for MQTT protocol. The way in which some things are programmed in Arduino may not be as elegant as they could be using other environments, but it has the largest capability for programming the ESP devices.

## **5.4 Protocol for Data Exchange**

We will be using the MQTT protocol to send information. The messages will be sent using QOS, a type of message within the MQTT protocol. It will be hosted either on Amazon server or on Nomad. For the battery, the power will be connected using DC-DC converters.

# **6 High Level Design Decisions**

After assessing the overall functionality of our system, we analyzed each subsystem by the high level decisions we considered in order to conceptualize our device up to this point.

**GPS Location Tracking**

 The main requirement of our system lies in tracking the location of the bike via a GPS device which can determine latitude and longitude at a specific time. The GPS device will need to be powered by the battery with enough voltage to drive the device while simultaneously not completely draining the battery. The device will be programmed to turn on once every 5 minutes in order to avoid this constant power draw and conserve the battery. The GPS data is transmitted in NMEA sentences which provided a multitude of information about the location of the transmitter. For our purposes of obtaining latitude, longitude, and time, we will utilize the GGA and GGA sentences. These sentences contain the necessary information and once parsed, they will be able to provide necessary information to the microcontroller which will send data through WiFi Capability to the User Interface.

**User Application Interface**

 The user application interface serves the purpose of communicating with the user and the device to send and receive necessary messages between the two. In order to make the device have maximum utility with a user friendly interface, we will make a mobile application compatible with iOS software. Within this application, there will be an option to select the mode that the user wishes the device to operate in. This will allow for maximum battery life so there is not a large amount of unnecessary battery used. The ESP8266 will transmit data signals between the BikeTracker device and the user. When requesting a location, the “park” mode will be active and request activated in order to indicate that location data should be received. Due to the network considerations, we will utilize an MQTT protocol to connect the user inputs with the device.

**WiFi Capability**

 The WiFi capabilities necessary for this device rely heavily upon the networks available to be used as the device travels. Due to power considerations combined with complications in connecting to secured networks, we will connect to open networks available on college campuses. Specifically at Notre Dame, we will connect to the ND-Guest network while running the MQTT network through either Nomad or Amazon server. The ESP8266 will be able to transmit and receive signals from the device and from the user application due to its ability to send messages that can wait to be picked up.

**Battery Power**

 Within the battery power subsystem, the power considerations for maintaining battery are of high importance. Since the user will not want to worry about the battery often, we have elected to use a rechargeable battery that will need to supply 5 V to the device in order to power all the of the various components. Also, we elected to design an element on the board that can indicate when the battery is close to empty so the user is aware. We elected to use this type of system as it optimizes the balance between using the battery enough to get reading while also conserving the system so it is not constantly sampling.

#

#

# **7 Open Questions**

In designing our BikeTracker device, we have several concerns about how we will meet some of the requirements necessary. Below is a list of these concerns that will be taken into serious consideration while furthering the design.

* Connectivity to spotty/inconsistent WiFi networks and how we will manage to keep the signal continuous while measurements are being taken
* Transmission of information from the GPS to application so that user will be able to easily interpret (must have a map or some more concrete way to present information)
* Battery life for the device and how we will not overuse it in order to require constant replacement of the batteries
* Size of the device being small enough to not be noticeable on the bike
* Connecting specific devices to the user--necessity for a log-in/serial number input of sorts in order for product to be connected to user application

# **8 Major Component Costs**

* For making the circuit for the device, the cost will be around $50.
* For GPS technology, there are different GPS modules that we could incorporate into the receiver. GPS modules usually run in the realm of $40
* For wi-fi connections, a wifi transmitter can be purchased for around $7.
* For powering the device, a battery holder will cost under $5.
* For disguising the device, a set of reflectors is needed which will cost under $10.
* This brings our total cost to around $120.

# **9 Conclusions**

**References**

* <https://www.sparkfun.com/pages/GPS_Guide>
* <https://www.sparkfun.com/products/13678>
* <http://www.dreamincode.net/forums/topic/92034-nmea-file-parsing/>
* <http://www.instructables.com/id/Using-the-ESP8266-module/>
* [http://www.amazon.com/REFLECTOR-SUNLT-FT-WHT-RR-RED-SP-27-2/dp/B009BFEQ5A/ref=sr\_1\_2?ie=UTF8&qid=1449345303&sr=8-2&keywords=front+reflector+bicycle](http://www.amazon.com/REFLECTOR-SUNLT-FT-WHT-RR-RED-SP-27-2/dp/B009BFEQ5A/ref%3Dsr_1_2?ie=UTF8&qid=1449345303&sr=8-2&keywords=front+reflector+bicycle)
* <http://www.nxp.com/documents/data_sheet/P89LPC901_902_903.pdf>
1. http://www.bicyclelaw.com/p.cfm/bicycle-safety/about-bike-theft [↑](#footnote-ref-0)