BU Memo Senior Design ENG EC 463



To: Alan Pisano, Osama Alshaykh, Babak Kia

From: John McCullough, Shantanu Bobhate, Brian Tan, Bandhan Zishanuzzaman, Teng Zhang

Team: 2 aka Right Alert

Date: 4/14/2016

Subject: **Functional Test Plan**

**1.0 Introduction**

**1.1 The Problem**

Alan Pisano, our client and a professor at the College of Engineering at Boston University, realizes that biking in Boston is not as safe as it could be with today’s technological resources. Based on the biking statistics from city of Boston, there have been approximately 150 self reported bike collisions on or in close proximity to the Boston University campus each year. One of the most common types of collisions cars have with bikers are those that involve right turns across the bike lane. Our team, RightAlert, has been given the task to create an alerting system that will help to decrease the number of these types of specific incidences.

**1.2 Customer Requirements**

The requirements for our alerting system were as follows:

1. Detect bikers in the bike lane with the use of a camera.
2. Provide a simple and intuitive alerting system to alert drivers.
3. Collect valuable traffic data, such as date and time of bike detection.
4. Store and display the data on a central website.

**1.3 Setup**

For our demonstration we wanted to show the performance of the system on live video feed of Cummington Mall. However, we did not want to finalize the setup of the system yet since we still haven’t received the sign and so would like to install everything together during the customer installation period. For now, we have our entire detection system setup inside a weatherproof box. This includes the Nvidia Jetson, the battery, the battery charging circuit and other miscellaneous circuits that help monitor the battery percentage and control the sign. We gave the camera an separate enclosure to avoid drilling into the box and compromising its weather proofness. Currently, we have the box temporarily set up on a ladder looking down at Cummington Mall.

**NOTE:** The height that the camera is at is not the optimal height. Therefore, it’s view does not span the area we had hoped and for the purpose of this demonstration we will have to work around this restriction.

**2.0 Bike Detection System**

**2.1 Bike Detection Algorithm**

*Description and Goal*

For bike detection we are using a Machine Learning Algorithm. The procedure trains an SVM classifier with positive and negative training images taken at the installation site (Cummington Mall). In order to provide an accurate detection rate, training images with rich differences were taken. The negative images must focus on objects that could have very similar HOG features to that of a biker, eg: people, tree trunks.

Currently, the algorithm has been trained with **556** positive and **1173** negative training images taken on Cummington Mall. In order to test the performance of this system we tested the predictions of this system for numerous different dataset combinations.

For the demonstration, we plan to run our algorithm on a live feed of Cummington Mall and observe its accuracy.

The goal of having a well trained algorithm is to minimize the rate of false negative and false positive readings of oncoming bikers. The lower these rates are, the safer an experience we can guarantee drivers and bikers using an intersection that implements RightAlert.

*Procedure*

Use the **ssh -X** [**ubuntu@ra-jetson.duckdns.org**](mailto:ubuntu@ra-jetson.duckdns.org) command to connect to the Jetson and gain remote access to it and it’s graphical output.

To run the algorithm on the test video, first direct yourself to the directory **~/Desktop/Bike\ Detector\ 3.0/**and run the executable **./out**. This should provide you with a help menu with a guide to the flags that can be used. For the purpose of this demonstration we will be using the -**-detection** flag along with the **--video** flag followed by **test/test-video-3.avi**. This will run the algorithm on a test video which will allow us to observe the detection process. We will run this only for **\_\_\_** mins due to the time constraint. The detections will be saved to the **detection record** folder. These will be used to improve the algorithm.

For the second part of our demonstration we will be using the -**-detection** flag along with the **--camera** flag. This will run the detection program on the camera feed. We will have John ride down the street and observe the program’s accuracy in detecting him. The program will keep a record of the detected bike frames in the **detection record** folder in order to further improve the algorithm later on.

**NOTE:** There may be severe lag since the output would be over an ssh connection.

*Verifiable Result*

The algorithm will update the database and the website every time it detects a bike. It will capture a timestamp. This will allow us to evaluate the detection images and help us further improve our algorithm by re-training the SVM with any false positives.

**2.2 Algorithm Interaction with Alert Sign LEDs**

*Description and Goal*

In order to simulate our regulated LED lit alert sign for the demonstration, we substituted a full-size mock-up alert sign with standard LEDs. The LEDs are connected in parallel in a simple circuit where their positive ends are connected to pin50 of the Jetson and their negative ends are connected the GND on the Jetson through an electrical relay. While the detection algorithm is observing thevideo, the LEDs will turn on everytime it has made a detection of what it believes is a biker. This test is to ensure the Jetson’s GPIO interact harmoniously with the OpenCV code it is running. This is an incredibly important portion of the alert system because it will be the visual cue drivers will use to stop for oncoming bikers.

*Procedure*

Since the LEDs are already connected to the Jetson’s GPIO, we need only to observe them when the algorithm has made a detection. When a biker is detected, we should expect the LEDs to turn on. When a biker is no longer detected the LEDs should turn off.

*Verifiable Result*

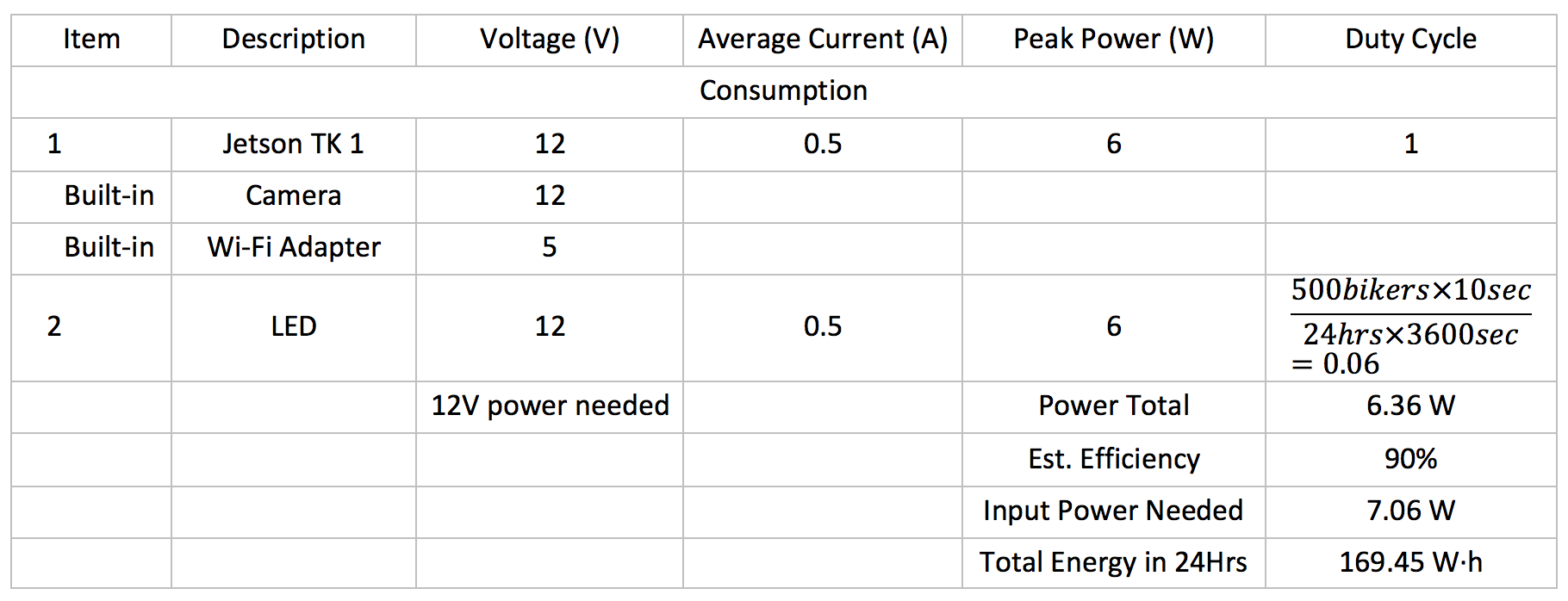
We will get desired results if the LEDs turn on the same time a biker is detected in the video and if the turn off when there is no longer a detected biker.

**3.0 Power Budget**

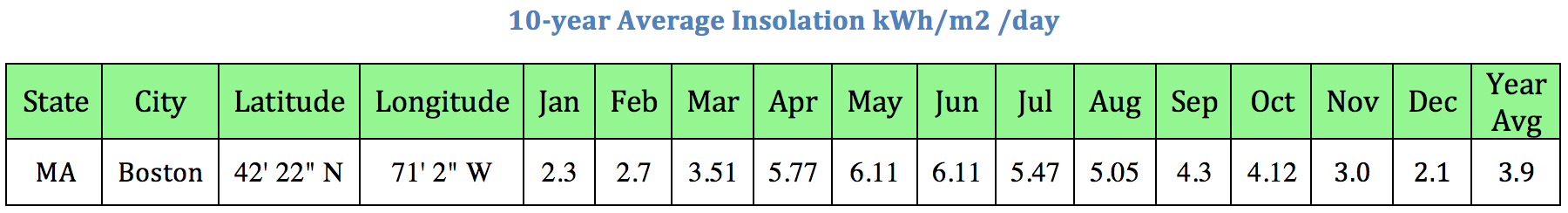
*Description and Goal*

The entire setup is completely powered by a self–sustaining solar power system. The power system mainly consists of two parts: solar panel and battery. While the battery provides the power to system at nights, the solar panel charges battery and support the system during daytimes.

To fully investigate the power consumption of entire system, we listed all components’ power consumption in the power budget listed below. For calculation, we assume there are 500 bikers turning right at the cross everyday, and LED sign is triggered and turned on for 10 sec for each warning.



Considering the size limitation and electrical performance under extreme weather conditions in Boston area, we choose to use Lipo(lithium polymer) battery. The capacity of battery is 6400 Ampere-hour, which can support our system for roughly 11 hours without charging. Since the lipo battery is sensitive to the charging current and stability of charging voltage, a charger regulating the charging conditions is needed in our system.



According to NASA Surface meteorology and Solar Energy data, the average solar radiance in Boston area is 3910 Wh/m2/day, which means the solar panel can receive 3.9 hours of sun per day in 1000 W/m2 in average. Thus, we choose our Renogy 50W 12V solar panel, which can provide 12V/2.84A power to the system in 1000 W/m2 condition. In this case, the solar panel can provide 195 W·h in the daytime, which is larger than the total power consumption of entire system 169.45 W·h in 24 hours.

*Procedure*

To calculate the power budget, power consumption of each electrical component is required. Simply using the power data from specification sheet is not accurate enough and the power consumption of Jetson TK1 varies significantly when algorithm is running in different periods. Since Jetson TK1 can take direct DC power supply from the battery and working voltage is constant 12 V, the power consumption can be calculated by multiplying the 12V voltage by the current, which is measured by connecting the Ampere-meter in series with the Jetson TK1.

*Verifiable Result*

The power budget is easily verified when the system is given power and works. This is really the base requirement of the entirety of the system, so if the camera, sign and other components turn on, then the power is verified.

**4.0 RightAlert Website**

*Description and Goal*

The RightAlert Website provided a central administration portal for the customer to view collected data, verify the system is working properly, and provide remote administration. Upon visiting <http://admin.rightalert.net> the user is asked to login. Administrators will be provided with a default admin account to login with. After successful login, the customer is presented with the homepage which displays a brief visual of recent data, an overview of current system information, and a live activity feed. The Overview section provides information such as number of bikers that have passed throughout the day, number of overall bikers that have passed since the system was implemented, and the RightAlert system battery level. The activity feed is a live reflection of a database which contains recent system events. The website is coded with C# and uses the ASP.NET framework.

*Procedure*

Similar to the LEDs, the algorithm’s update to the RightAlert website are automatic and rely on no input from the user. We will simply wait for the system to make a detection and observe if the website is updated. This procedure is used on the homepage for the bike counters, battery level, and the activity feed. For the Advanced page, the battery information panel and lastest detection image are also tested through observation. The page also contains an Advanced Stats tool where a user may inquire about number of bikers that have crossed an intersection in a user defined timeframe. Through the Terminal panel the user is also able to execute commands directly over SSH.

*Verifiable Result*

After the algorithm detects a bike, in addition to the LED lights, a script is run to update the SQL database and upload an image to AWS S3. By visiting <http://admin.rightalert.net> it can be confirmed that the database was updated with a record of a biker and the latest captured image is visible on the webpage. Further verification can be completed by direct view of database tables and local Jetson storage.

**NOTE:** A large portion of data on the website relies on the Jetson to be connected to the Internet. There are scripts running on the Jetson to validate network connectivity and force reconnect attempts if the “Right Alert’ network is in range.