DEPARTMENT OF TRANSPORTATION

Deploy and Test a Smartphone-Based Accessible Traffic Information System for the Visually Impaired

Chen-Fu Liao, Principal Investigator
Department of Mechanical Engineering
University of Minnesota

OCTOBER 2020

Research Project
Final Report 2020-28

To request this document in an alternative format, such as braille or large print, call 651-366-4718 or 1-800-657-3774 (Greater Minnesota) or email your request to ADArequest.dot@state.mn.us. Please request at least one week in advance.

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
MN 2020-28			
4. Title and Subtitle Deploy and Test a Smartphone-Based Accessible Traffic Information System for the Visually Impaired		5. Report Date October 2020	
		6.	
7. Author(s) Chen-Fu Liao and Brian Davis		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Mechanical Engineering University of Minnesota 111 Church Street SE Minneapolis, MN 55455		10. Project/Task/Work Unit No. CTS#2019004	
		11. Contract (C) or Grant (G) No. (C) 1003325 (WO) 64	
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Office of Research & Innovation 395 John Ireland Boulevard, MS 330 St. Paul, Minnesota 55155-1899		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes http:// mndot.gov/research/reports/2020/202028.pdf			

16. Abstract (Limit: 250 words)

An increasing number of Accessible Pedestrian Signals (APS) have been installed at new or upgraded intersections to assist people with vision impairment to navigate streets. For un-signalized intersections and intersections without APS, people with vision impairment have to rely on their own orientation and mobility skills to gather necessary information to navigate to their destinations. Previously, a smartphone-based accessible pedestrian system was developed to support wayfinding and navigation for people with vision impairment at both signalized and un-signalized intersections. A digital map was also created to support the wayfinding app. This system allows a visually impaired pedestrian to receive signal timing and intersection geometry information from a smartphone app for wayfinding assistance. A beacon using Bluetooth Low Energy (BLE) technology helps to identify a pedestrian's location when he or she travels in a GPS-unfriendly environment. A network of Bluetooth beacons ensures that correct traffic information is provided to the visually impaired at the right location. This project leverages the previous work by installing the system at a number of intersections in downtown Stillwater, Minnesota, where MnDOT operates the signalized intersections. In this study, researchers interface with the traffic controllers to broadcast traffic signal phasing and timing (SPaT) information through a secured and private wireless network for visually impaired users. The aim is to test the smartphone-based accessible system and evaluate the effectiveness and usefulness of the system in supporting wayfinding and navigation while the visually impaired travel through signalized and un-signalized intersections.

17. Document Analysis/Descriptors		18. Availability Statement	
Traffic signals, Visually impaired persons, Bluetooth technology,		No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312	
19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	51	

Deploy and Test a Smartphone-Based Accessible Traffic Information System for the Visually Impaired

FINAL REPORT

Prepared by:

Chen-Fu Liao and Brian Davis
Mobility Technology Laboratory
Department of Mechanical Engineering
University of Minnesota

October 2020

Published by:

Minnesota Department of Transportation
Office of Research & Innovation
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

ACKNOWLEDGMENTS

This project is sponsored by the Minnesota Department of Transportation (MnDOT). The authors would like to acknowledge the support provided by members of the technical advisory panel (TAP), including their invaluable feedback and assistance in making this study possible.

Special thanks to Mike Fairbanks, Bob Olson, Derek Lehrke, Timothy Johnson, and Martin Carlson from the MnDOT traffic operations office for their support at the field test sites and providing access to traffic control cabinets. The authors would like to thank the staff and engineers at Miovision for their technical support and providing Application Programming Interface (API) access to their cloud server.

Finally, the authors would like to thank the Center for Transportation Studies (CTS) for providing administrative assistance for this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADA	Americans with Disabilities Act
AFB	American Foundation for the Blind
API	Application Programming Interface
APS	Accessible Pedestrian Signals
ATTRI	Accessible Transportation Technologies Research Initiative
BLE	Bluetooth Low Energy
BSM	Basic Safety Message
CTS	Center for Transportation Study
CV	Connected Vehicles
DOT	Department of Transportation
DSRC	Dedicated Short Range Communication
EAR	Exploratory Advanced Research
FHWA	Federal Highway Administration
FTA	Federal Transit Administration
GPS	Global Positioning System
GUI	Graphical User's Interface
HTTP	Hypertext Transfer Protocol
I2P	Infrastructure to Pedestrian
I2X	Infrastructure to Anything
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IoT	Internet of Things
IP	Ingress Protection
JSON	JavaScript Object Notation
M2M	Machine 2 Machine
MAPS	Mobile Accessible Pedestrian Signal
MnDOT	Minnesota Department of Transportation
MNIT	Minnesota Information Technology
MQTT	Message Queue Telemetry Transport
NEMA	National Electrical Manufacturers Association
NFB	National Federation of the Blind
NHIS	National Health Interview Survey
PED	Pedestrian

SAE	Society of Automotive Engineers
SLAM	Simultaneous Location and Mapping
SPAT	Signal Phasing and Timing
TAP	Technical Advisory Panel
TTS	Text-to-Speech
UMN	University of Minnesota
USB	Universal Serial Bus
USDOT	United States Department of Transportation
UWB	Ultra-wideband
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
WHO	World Health Organization
WZDI	Work Zone Data Initiative
WZDx	Work Zone Data Exchange

EXECUTIVE SUMMARY

An increasing number of Accessible Pedestrian Signals (APS) have been installed at new or upgraded Minnesota intersections to assist people with vision impairment navigate streets. For un-signalized intersections and intersections without APS, people with vision impairment have to rely on their own orientation and mobility skills to gather necessary information to navigate to their destinations.

The research team previously developed a smartphone-based APS system to support wayfinding and navigation for people with vision impairment at both signalized and un-signalized intersections. A digital map was created to support the wayfinding app. This mobile traffic information system allows a visually impaired pedestrian to receive signal timing and intersection geometry information from a smartphone for wayfinding assistance. A beacon using the Bluetooth Low Energy (BLE) technology was also developed to help identify a pedestrian's location when he or she is traveling in a GPS unfriendly environment such as an urban canyon. A network of Bluetooth beacons was deployed to make sure that corresponding traffic information is provided to the visually impaired at the right location.

This project leverages the team's previous effort by installing the mobile traffic information system at 6 intersections (including 4 signalized and 2 un-signalized intersections) in Stillwater, Minnesota, where MnDOT operates the signalized intersections. The goals are to (1) install a data monitoring system to interface with the traffic controllers and broadcast traffic signal phasing and timing (SPaT) information through a wireless network, (2) refine the smartphone-based accessible traffic information system to display SPaT information, and (3) validate system performance in supporting wayfinding and navigation for the visually impaired while they travel through signalized and un-signalized intersections.

The research team has installed a collection system (SmartLink from Miovision) to acquire real-time SPaT data from each traffic controller through MnDOT's secured infrastructure network. A cloud server broadcasts the SPaT data through a subscription-based cellular network. The SmartLink system has the capability of monitoring and managing traffic signal control remotely. It enables SPaT data broadcast for this application and provides real-time pedestrian signal information for pedestrians.

The research team initially developed a hypertext transfer protocol (HTTP)-based interface on an Android smartphone to receive live SPaT information from a cloud server. However, the initial test results indicated that the HTTP-based interface has a data latency of 3 to 5 seconds due to the data overhead. In addition, the HTTP data latency was inconsistent and depended on Internet network traffic. To address the latency issue, a Message Queuing Telemetry Transport (MQTT) interface was later implemented to timely transmit real-time signal phasing and timing data.

An Android app called *PedNav* was then developed to provide intersection geometry and signal timing information and wayfinding assistance to visually impaired pedestrians. It uses embedded smartphone sensors to determine the user's position and heading. An Internet connection communicates with the MQTT server that provides live signal phase information. The PedNav app uses the SPaT and location information to notify the user when the walk or don't walk sign is displayed at a particular crosswalk. A digital map stores the geometry information of an intersection, such as positions of street corners and

their associated crosswalks. The map is used by the PedNav app to determine which crosswalk's phase and timing information is requested by the user.

To validate system performance, the research team tested the phone location, wireless data transmission latency, and correctness of the displayed SPaT information (both visual and audio displays) at each of the corners at all of the intersections when using an Android smartphone. For example, after arriving at an intersection, a user points the smartphone in different directions where a crosswalk may or may not exist. The user can then perform a single or double tap on the phone screen to request intersection geometry and signal timing information, respectively.

The phone-reported position accuracy at each test location ranged from 4 to 8 meters with an average reported accuracy of 5.6 meters. This measure provided by the smartphone's GPS sensor is an indicator of its current position accuracy. The team also tested the correctness of text and audible messages displayed on the smartphone when a pedestrian performs single and double taps. Out of 137 message correctness tests, the system successfully provided 132 (96%) correct feedbacks on intersection geometry and signal status information. Incorrect information was presented to users 5 times (4%) due to incorrect headings measured by the magnetometer sensor on the smartphone. Orientation information provided by the digital compass on the smartphone could be distorted when the phone is near a large ferrous metal object in the environment.

To evaluate the MQTT data latency, the research team used a camera to record the pedestrian walk phase transition on the pedestrian signal head and the display on the smartphone at different corners of the intersections. On average, the MQTT protocol had a data latency of 0.88 sec with a standard deviation of 0.38 sec. The maximum latency was 1.3 sec observed at the intersection of Main Street and Myrtle Street, while the minimum latency of 0.33 sec was observed at the intersection of State Highway 36 and Greeley Street South.

This proposed approach will allow state, county, or local agencies to provide a more complete and accessible solution for the visually impaired at both signalized and un-signalized intersections, thus improving the mobility and independence of visually impaired pedestrians in using the transportation system. In addition, the mobile APS approach can also be applied to provide the visually impaired with a sidewalk work zone bypass or transit arrival information at bus stops and light rail stations.

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

An increasing number of Accessible Pedestrian Signals (APS) have been installed at new or upgraded intersections to assist people with vision impairment to navigate streets. For un-signalized intersections and intersections without APS, people with vision impairment have to rely on their own orientation and mobility skills to gather necessary information to navigate to their destinations. We previously developed a smartphone-based accessible pedestrian system to support wayfinding and navigation for people with vision impairment at both signalized and un-signalized intersections. A digital map was also created to support the wayfinding app. This system allows a visually impaired pedestrian to receive signal timing and intersection geometry information from a smartphone app for wayfinding assistance. A beacon using Bluetooth Low Energy (BLE) technology was also developed to help identify a pedestrian's location when he or she is traveling in a GPS unfriendly environment. A network of Bluetooth beacons allows us to make sure that correct traffic information is provided to the visually impaired at the correct location.

1.2 OBJECTIVE

This project leverages our previous effort by installing the mobile accessible traffic information system at six intersections in downtown Stillwater, Minnesota, where MnDOT operates the signalized intersections. The objectives are to (1) install a data monitoring system to interface with the traffic controllers and broadcast traffic signal phasing and timing (SPaT) information through a cellular network, (2) refine the smartphone-based accessible traffic information system, and (3) validate the system performance in supporting wayfinding and navigation for the visually impaired while they travel through signalized and un-signalized intersections.

1.3 LITERATURE REVIEW

In 2019, the World Health Organization (WHO) estimated that at least 1 billion people have a vision impairment or blindness, globally [1]. The 2017 National Health Interview Survey (NHIS) data [2] reported that an estimated 26.9 million adult Americans (about 10% of all adult Americans) reported they either "have trouble" seeing, even when wearing glasses or contact lenses, or that they are blind or unable to see at all [3]. In 2016, Minnesota had more than 86,000 individuals who reported having a visual disability [4].

Individuals with visual impairments face significant challenges traveling in the physical environment. Wayfinding is difficult for an individual with blindness [5]. It involves the ability to learn and recall a route as well as to update one's orientation as he or she navigates along the route [6 & 7]. Though not always reliable, many environmental cues are available to support the decision making of people with vision impairment on various components of wayfinding. They often use auditory and limited visual information that they gather to make safe decisions while traveling in a transportation network.

Although there are many aids (such as electronics, Braille maps, etc.) to assist wayfinding in addition to using a cane or guide dog, blind people tend to use their cognitive map and spatial knowledge as primary guidance [8]. Giudice and Legge [9] reviewed various technologies developed for blind navigation and concluded that no single technology can provide both indoor and outdoor navigation and guidance for the blind. It is critical to gain more insight from studying perception to have a clear understanding of the cognitive demands on the blind when they interpret information received by the sensory system.

For example, some blind pedestrians sometimes have difficulty crossing intersections at some locations due to the lack of information available to them about traffic, geometry at intersections [10 & 11], and intersection types (signalized, un-signalized or roundabout). Guth et al. [12] initially found that site-specific characteristics (for example, treatments such as rumble strips or speed countermeasures) appeared to have a greater impact on reducing the number of conflicts between pedestrians and vehicles than did a mobility device such as cane or guide dog. However, subsequent studies conducted by Bourquin et al. [13-16] found that a mobility cane had an extremely high impact on drivers' yielding and the potential to significantly reduce conflicts.

In 2012, the Accessible Transportation Technologies Research Initiative (ATTRI), a joint federal initiative led by the Federal Highway Administration (FHWA) and the Federal Transit Administration (FTA), sponsored three research studies to develop situational awareness and guidance solutions for people with vision impairment and other disabilities. The first study sponsored by the Exploratory Advanced Research (EAR) program focused on navigation and wayfinding for the visually impaired in unfamiliar environments. Rose et al. [17] developed a prototype system that incorporates wearable components such as GPS, Inertial Measurement Unit (IMU), stereo camera, pedometry, tactile belt, smartphone, and a wireless radio to provide navigation solutions to the visually impaired.

One approach for indoor navigation developed by TRX Systems uses sensor fusion techniques and a proprietary mapping software for cloud-based indoor navigation. The system uses a wearable beacon, an Inertial Navigation System (INS), paired with a smartphone. The system includes a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis magnetic field sensor, a barometric pressure sensor, and both Bluetooth low energy and Ultra-wideband (UWB) transceivers. They used UWB transceiver to provide a time-of-flight ranging corrections for indoor positioning.

The third study used computer vision on a smartphone to provide situation awareness and simultaneous location and mapping (SLAM)-based navigation assistance [18-20] based on the Google Project Tango technology. This third study used a fish-eye camera to detect sparse features from an environment and stored the detected features in a database. A user's position was then determined when the same set of image features were detected and matched with stored features during navigation. Image processing techniques were also implemented for signage detection and recognition. The Google Project Tango system integrated a motion tracking camera and depth sensing for navigation without using GPS or other external signals. It incorporated advanced computer vision, image processing, and special vision sensors for spatial perception. However, the Google Tango technology is only available on a limited number of Android smartphone and tablet devices with a fisheye-lens camera for motion tracking.

Over the last decades, the development of navigation devices capable of guiding people who have vision impairment through indoor and outdoor scenarios has remained a challenge. Smartphones and wearable devices with built-in sensors could provide potentially feasible options to support wayfinding and monitoring of the surroundings of the visually impaired [21].

Coughlan and Shen [22] used computer vision on a smartphone to help people who are visually impaired achieve proper alignment with the crosswalk and read the status of walk lights to know when it is time to cross. Later, Fusco et al. [23] used the same platform to help the visually impaired pedestrian find important features in the intersection, such as walk lights, pushbuttons, and crosswalks, and achieve proper alignment to these features. However, this approach may be limited in challenging lighting conditions such as dusk or nighttime, or in the presence of rain or snow, or when the paint marking a crosswalk is peeling.

We previously developed a Mobile Accessible Pedestrian System (MAPS) to provide navigation and signal information to the visually impaired [24]. MAPS is a personal system based on a smartphone carried by the user as compared to the existing infrastructure-based APS system installed at an intersection. The MAPS system integrates information from sensors commonly available on a smartphone and then wirelessly communicates with an intersection's traffic signal controller to obtain real-time Signal Phasing and Timing (SPaT) information, which together then inform the user where they are and when to cross streets. An automated 'pedestrian call' request (i.e., a signal sent to the intersection traffic controller when a pedestrian presses a pushbutton) can be sent to a traffic controller wirelessly from a smartphone of registered users after confirming the direction and orientation that they intend to cross. With the automated 'pedestrian call' function, MAPS eliminates the need of physically locating and pressing a pushbutton near a crosswalk and provides intersection geometry information, such as street name, number of lanes, and signal types, to the blind at an intersection crossing. The automated 'pedestrian call' feature was previously implemented and tested at the intersection of Highway 55 and Rhode Island Avenue in Golden Valley, Minnesota [24]. The smartphone app provides an auditory or vibrotactile warning message to pedestrians when the walk sign is on and when the walking time has about 5 seconds left.

We also developed a smartphone app, in connection with Bluetooth beacons placed at key decision points near a work zone, to provide situation awareness along with routing or bypassing information to people with vision impairment [25 & 26]. A geospatial database of the locations of the Bluetooth beacons was developed to allow the smartphone app to query audible messages associated with discovered Bluetooth beacons.

1.4 REPORT ORGANIZATION

This report is organized as follows. Chapter 2 discusses test sites, system installation and the Bluetooth assembly. Chapter 3 describes the smartphone app development and the API interface to receive live traffic signal status data. Chapter 4 shows results from field experiments and system verification and validation. Chapter 5 contains a summary and a discussion on potential connected vehicles (CV) applications for pedestrians.

CHAPTER 2: SYSTEM DESIGN AND INSTALLATION

Six intersections in Stillwater, Minnesota, were identified to install the Mobile Accessible Pedestrian Signal (MAPS) systems. A data collection controller was installed at each signalized intersection to monitor and broadcast live signal phasing and timing (SPaT) information. A smartphone program was developed to receive the SPaT from a server and provide signal timing information to pedestrians. Solar-powered Bluetooth devices were placed at intersection corners to allow the smartphone app could correctly determine its location to ensure the system would provide the correct SPaT information.

2.1 TEST SITES

The research team worked with the TAP to identify 6 intersections (see Table 2.1) in Stillwater for system implementation and testing. We first met with a MnDOT traffic technician to examine each signal cabinet and ensure each controller cabinet meets the minimum hardware requirements for our application needs. The signal controllers in Downtown Stillwater have been upgraded to Econolite ASC3 controllers after our site visit in July. We also received drawings of the signal phasing and timing assignment at each signalized intersection from the traffic operations.

Table 2.1 List of Test Sites in Stillwater.

Site No.	Intersection Type	Controller Type	Cabinet Type	Description	# of Lanes
1	Signalized	ASC3 Controller	TS-1	Nelson Street & Main Street (MN-95)	2
2	Signalized	ASC3 Controller	TS-1	Chestnut & Main Street (MN-95)	2
3	Signalized	ASC3 Controller	TS-1	Myrtle & Main Street (MN-95)	2
4	Signalized	ASC3 Controller	TS-2	Greeley St & Highway 36	4
5	Un-Signalized	NA	NA	Nelson Alley & Main Street (MN-95)	2
6	Un-Signalized	NA	NA	Olive & Main Street (MN-95)	2

Figure 2.1 illustrates the location of the 6 intersections in the City of Stillwater. The circular red dots represent the signalized intersections and the green square marks represent the un-signalized intersections.

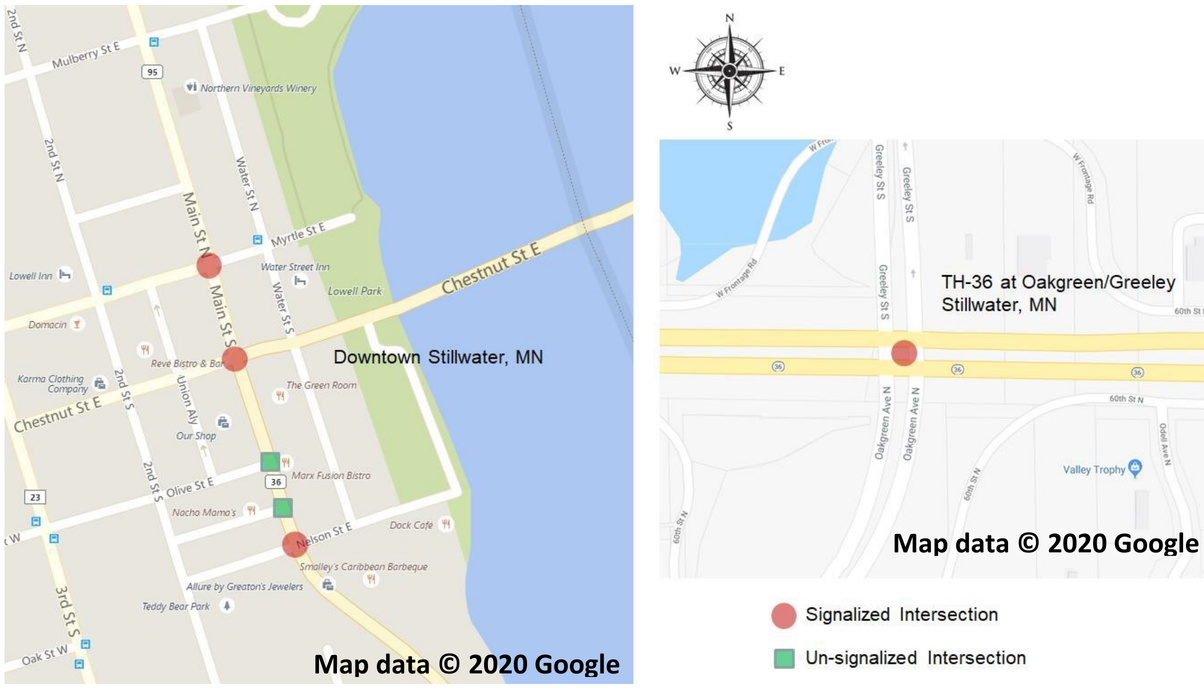

Figure 2.1 Selected Installation of MAPS at 6 Intersections in Stillwater.

In addition to the field visit, the research team had a meeting with MnDOT traffic engineers and staff from Minnesota IT services (MNIT) regarding receiving real-time signal information from the central traffic system. We discussed the need to obtain real-time traffic data from the MnDOT field network and were advised to deploy and test our system using the signal info from the local signal controllers. Using this approach, additional hardware was required to be installed in each controller cabinet (4 signalized intersections) in order to broadcast real-time signal timing and phasing information. A contract amendment including the hardware request was submitted to MnDOT for review and approval.

2.2 INTERFACE SIGNAL CONTROLLER

The research team has deployed 4 Signal Phasing and Timing (SPaT) data collection systems, i.e., SmartLink from Miovision for this project. The SmartLink system has the capability of monitoring and managing traffic signal control remotely. It enables us to collect and broadcast SPaT data for our application to provide pedestrian signal information for people with vision impairment.

We initially developed a HTTP-based interface on an Android Smartphone to receive live signal and timing information from the 4 traffic controllers through the Miovision SmartLink devices [27]. However, we later found the HTTP-based interface has a data latency over 3 seconds due to the data overhead of the protocol. In addition, the HTTP data latency was not consistent and varies depending on the Internet network traffic. To address the latency issue, we have worked with the Miovision team to implement a Message Queuing Telemetry Transport (MQTT) interface [28] to transmit real-time signal phasing and timing data.

2.2.1 SmartLink System Installation

We first install and test a SmartLink system in a test cabinet in the traffic operations division in MnDOT's metro district office. Tim Johnson from the Minnesota IT services department also helped us create a subnetwork that allows the SmartLink systems to be placed inside MnDOT infrastructure network for communicating with local traffic signal controllers. The network settings of the SmartLink systems and traffic controller at each intersection are configured and listed in Table 2.2 as follows.

Table 2.2 Network Settings of SmartLink Systems and Traffic Controllers.

Intersections	Highway 36 & Greeley South	Highway 95 & Nelson	Highway 95 & Chestnut	Highway 95 & Myrtle
Latitude	45.03598	45.054226	45.055794	45.056598
Longitude	-92.82218	-92.805104	-92.805792	-92.806127
SmartLink Serial ID	J000500	J000347	J000531	J000621
Cabinet Type	TS-2	TS-1	TS-1	TS-1
Controller Type	Econolite ASC-3	Econolite ASC-3	Econolite ASC-3	Econolite ASC-3

With the assistance from the traffic operations office, we installed the SmartLink hardware in the controller cabinets and connected them to the network switch inside each cabinet. The integrated antenna was temporary mounted inside the cabinet. Figure 2.2 illustrates the connection panel of a SmartLink system and a system installed at Greeley and Highway 36 intersection. The SmartLink Ethernet port #3 is connected to cabinet network switch port #6 as shown in Figure 2.3 at all 4 intersections.

The front panel of the SmartLink system has 4 LED indicators. The LED status is described as follows.

- POWER LED - Solid green confirms unit is powered
- STATUS LED - Blinking green indicates 'heartbeat'
- SERIAL LED - Solid green confirms Internet connectivity
- NET LED - Solid green confirms connectivity with Miovision server

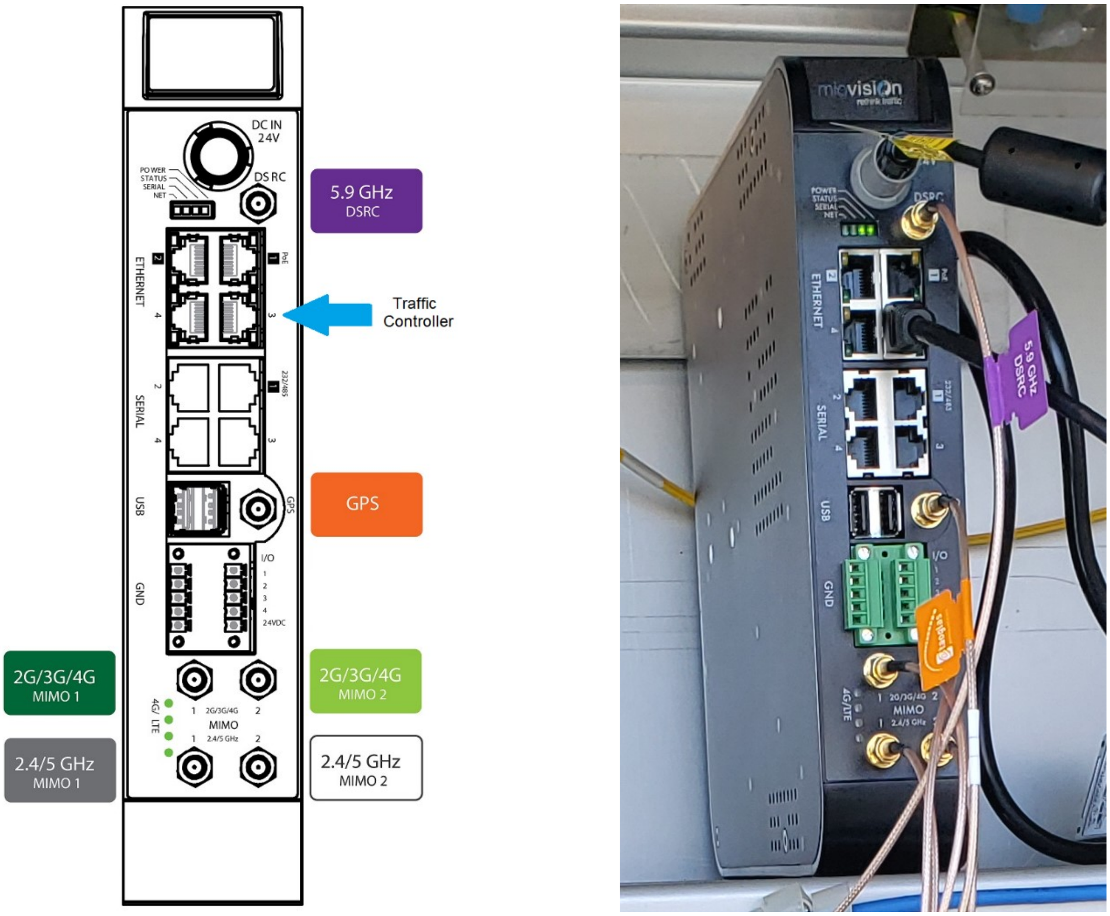

Figure 2.2 A SmartLink System.

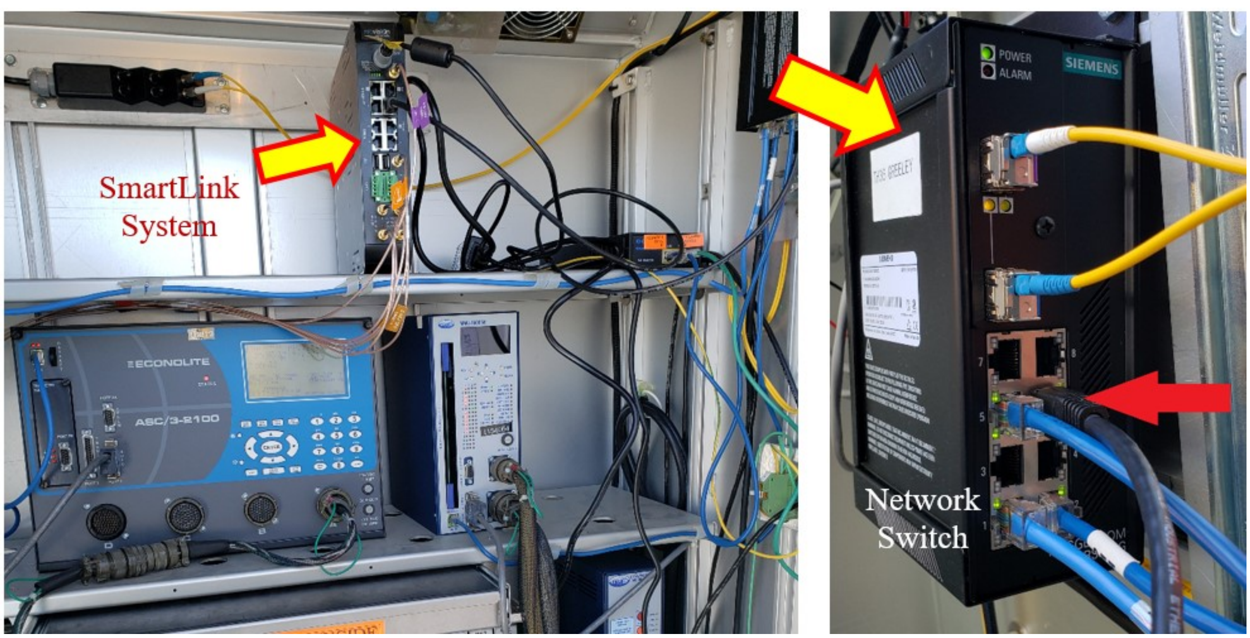

Figure 2.3 Installation of SmartLink System in Greeley & Highway 36 Controller Cabinet.

2.2.2 HTTP API

We developed a software interface on an Android smartphone to receive live signal and timing information from traffic controllers from the SmartLink devices. For data security consideration, a private key was generated to request SPaT data from an authorized remote client. Figure 2.4 illustrates the SPaT data flow from signal controller to SmartLink then to a smartphone app through a wireless Internet connection.

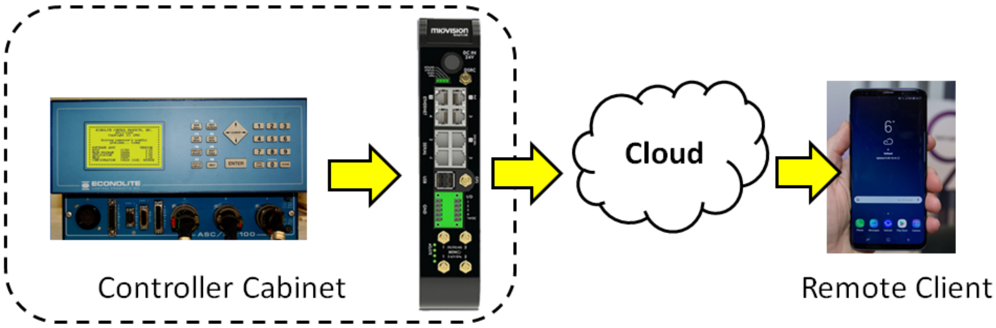

Figure 2.4 Illustration of Receiving Broadcasted SPaT data from a SmartLink System.

The received traffic data is formatted based on the Indiana Traffic Signal High Resolution Data Logger Enumerations. The SmartLink system supports the following event codes: 0, 1, 7, 8, 9, 10, 11, 12, 21, 22, 23, 61, 63, 64, 65, 81, 82, 89, 90. The SmartLink system monitors the controller signal states at 10 Hz rate through the wired Ethernet connection in the cabinet. A sample signal status data received from the SmartLink system is displayed in Table 2.3.

Table 2.3 Sample Signal Status Data.

Label Name	Value	Description
timestamp	1553858418924	Epoch time. # of milliseconds elapsed since 1/1/1970 UTC.
eventParam	6	Phases #1 to 16
eventCode	21	Event code, e.g., 21 – pedestrian begin walk

Tables 2.4 and 2.5 respectively list the signal phasing data codes and parameters for vehicle and pedestrian phases.

Table 2.4 List of Active Vehicle Phase Event Codes and Parameters.

Vehicle Phases			
Event Code	**Event Description**	**Parameter**	**Description**
0	Phase On	Phase # (1-16)	Set when NEMA Phase On becomes active, either upon start of green or walk interval, whichever occurs first.
1	Phase Begin Green	Phase # (1-16)	Set when either solid or flashing green indication has begun. Do not set repeatedly during flashing operation.
2	Phase Check	Phase # (1-16)	Set when a conflicting call is registered against the active phase.
3	Phase Min Complete	Phase # (1-16)	Set when phase min timer expires.
4	Phase Gap Out	Phase # (1-16)	Set when phase gaps out, but may not necessarily occur upon phase termination. Event may be set multiple times within a single green under simultaneous gap out.
5	Phase Max Out	Phase # (1-16)	Set when phase MAX timer expires, but may not necessarily occur upon phase termination due to last car passage or other features.
6	Phase Force Off	Phase # (1-16)	Set when phase force off is applied to the active green phase.
7	Phase Green Termination	Phase # (1-16)	Set when phase green indications are terminated into either yellow clearance or permissive (FYA) movement.
8	Phase Begin Yellow Clearance	Phase # (1-16)	Set when phase yellow indication becomes active and clearance timer begins.
9	Phase End Yellow Clearance	Phase # (1-16)	Set when phase yellow indication become inactive.
10	Phase Begin Red Clearance	Phase # (1-16)	Set only if phase red clearance is served. Set when red clearance timing begins.
11	Phase End Red Clearance	Phase # (1-16)	Set only if phase red clearance is served. Set when red clearance timing concludes. This may not necessarily coincide with completion of the phase, especially during clearance of trailing overlaps, red revert timing, red rest, or delay for other ring terminations.
12	Phase Inactive	Phase # (1-16)	Set when the phase is no longer active within the ring, including completion of any trailing overlaps or end of barrier delays for adjacent ring termination.

Table 2.5 List of Active Pedestrian Phase Event Codes and Parameters.

Pedestrian Phases			
Event Code	Event Description	Parameter	Description
21	Pedestrian Begin Walk	Phase # (1-16)	Set when walk indication becomes active.
22	Pedestrian Begin Clearance	Phase # (1-16)	Set when flashing don't walk indication becomes active.
23	Pedestrian Begin Solid Don't Walk	Phase # (1-16)	Set when don't walk indication becomes solid (non-flashing) from either termination of pedestrian clearance, or head illumination after a pedestrian dark interval.
24	Pedestrian Dark	Phase # (1-16)	Set when the pedestrian outputs are set off.

In order to verify and validate the signal data received from the controller, a graphical user interface (GUI) on the smartphone was developed to test the live signal application programming interface (API). In addition, a pseudo signal generator was also developed on the phone to test the functionality of the GUI interface. The GUI interface allows research team to select an intersection and displays the live vehicle and pedestrian phases for a simplified 8-phase intersection as illustrated in the bottom right corner of Figure 2.5. Bock diagram of signal API implementation and GUI data flow is illustrated in Figure 2.5. This GUI interface is intended to be used for testing and debugging purposes. Eventually, the SPaT signal will be processed in the background of the smartphone to provide appropriate signal information to the pedestrian users based on their location and intended direction of crossing.

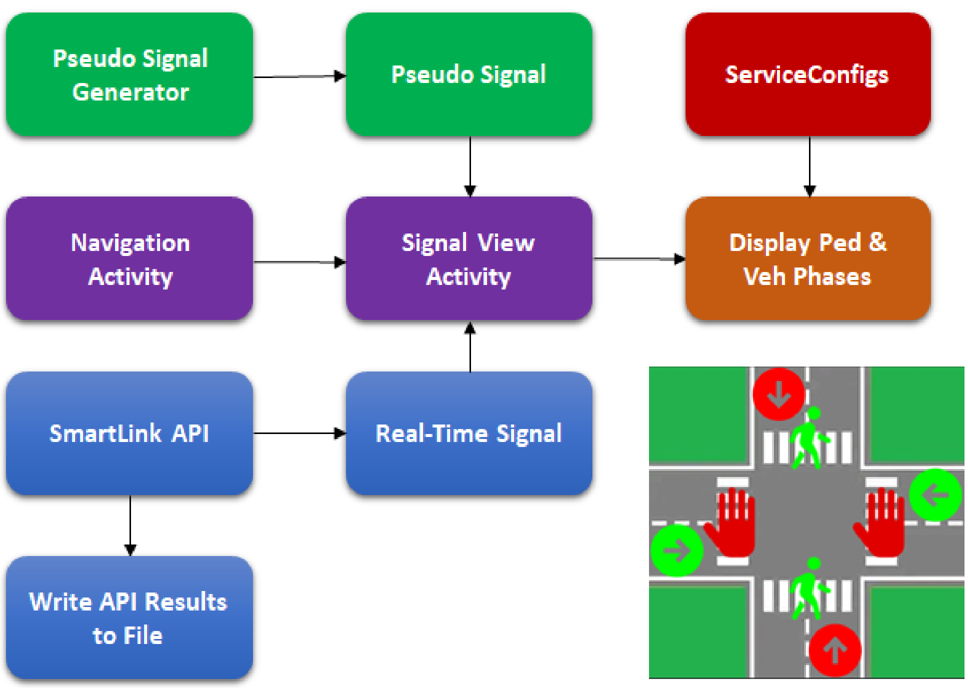

Figure 2.5 Block Diagram of Receiving Signal Status Data.

The research team took the smartphone app to each test site and verified the received signal phasing and timing information is corresponding to the signal light visualized in the field. We also noticed that the data latency for the signal status data to be received on the smartphone is about 3-sec.

2.2.3 MQTT Protocol for SPaT Data

MQTT is a machine-to-machine (M2M) or Internet of Things (IoT) [29] connectivity protocol. It is designed as an extremely lightweight publish/subscribe messaging transport. MQTT is designed for only delivering the data a client subscribes to. The publish/subscribe model provides clients with independent existence from one another and enhances the reliability of the whole system. That is, when one client is out of order the whole system can still keep on working properly.

The MQTT API exposes real-time data streams that are published to an MQTT broker running on the cloud server. The Miovision MQTT SmartLink API [30] provides signal channel, detector channel, preempt channel, timing plan changes, and connectivity topics for data subscription. The signal channel sends the red, green, or yellow light status on each channel when the status of any channel changes. The system can handle up to 64 traffic signal phases. Here is a sample channel message where phases 2 and 6 are green.

{"timestamp":"1593824094229","reds":[1,0,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"yellows":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],"greens":[0,1,0,0,0,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0]}

Currently, phase 1 to 16 are assigned for vehicles and 17-32 are pedestrian phases. Pedestrian phases on the traffic controller are assigned the same as the corresponding vehicle through phase. That is, pedestrian phase 2, 4, 6, 8, etc. The channel output from the SmartLink MQTT for pedestrian signals, walk (green), flashing don't walk (yellow), and don't walk (red), are mapped to channel 18 = PED 2, 20 = PED 4, 22 = PED 6, 24 = PED 8, respectively.

2.3 BLUETOOTH DEVICES

We have designed a solar panel mounting assembly and integrated Bluetooth sensors in an IP65 rating and waterproof enclosure. The mounting assembly and Bluetooth enclosure are respectively described as follows.

2.3.1 Solar Charger and Enclosure

Figure 2.6 shows a USB/DC/solar battery charging circuit board [31], a 3.7V 4400 mAh lithium ion battery pack [32], and a Bluetooth low energy module, NORDIC nRF52832 [33 & 34], packaged in an IP65 rating and waterproof NEMA enclosure [35]. The enclosure size is 2.5" W x 4.5" L x 1.75" H. The solar lithium ion polymer battery charger was designed specifically for solar charging. It will automatically draw the most current possible from the panel in any light condition. It uses the input power when available and keeps battery from constantly charging or discharging.

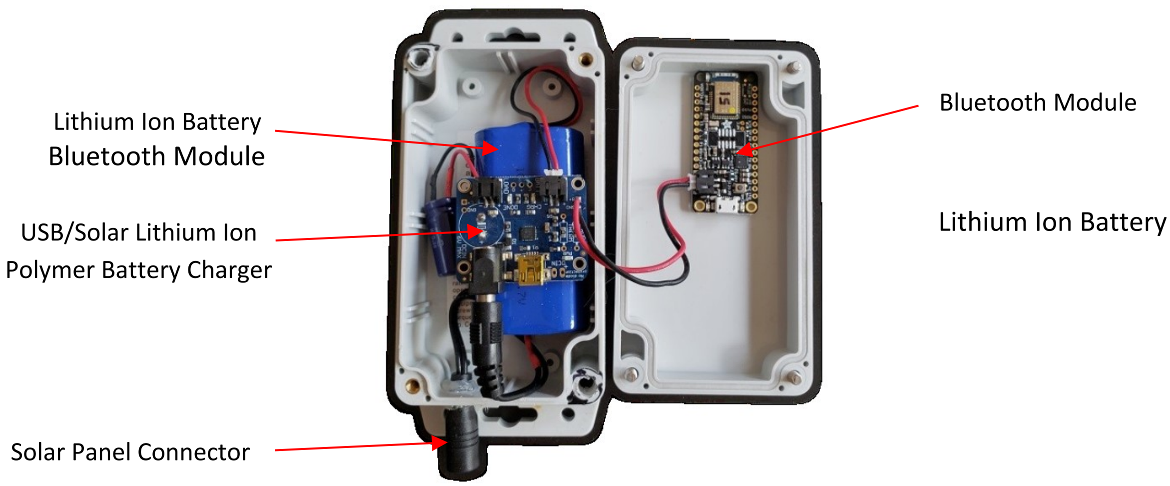

Figure 2.6 A Bluetooth Module and Solar Charging System.

2.3.2 Solar Panel Mounting Assembly

Figure 2.7 displays the drawings for two 1-inch angle bars that were directly attached to the solar panel which is 4.4 in x 5.4 in (11.2 cm x 12.7 cm) in size. Another two ¾ in angle aluminum bars (as illustrated in Figure 2.8) are used for adjusting the tilt angle of the solar panel. The entire solar panel mounting assembly (see Figure 2.9) can be attached to a lamp post using zip ties or a stainless steel worm gear clamp.

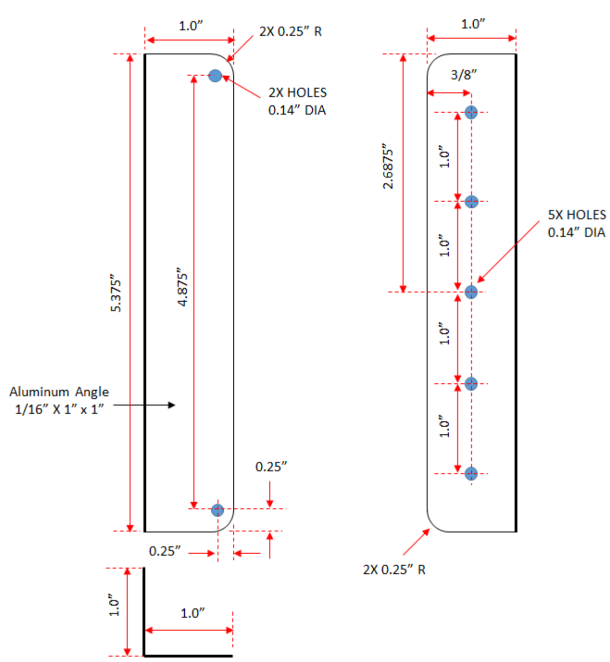

Figure 2.7 Drawings of Angle Bar for Holding Solar Panel.

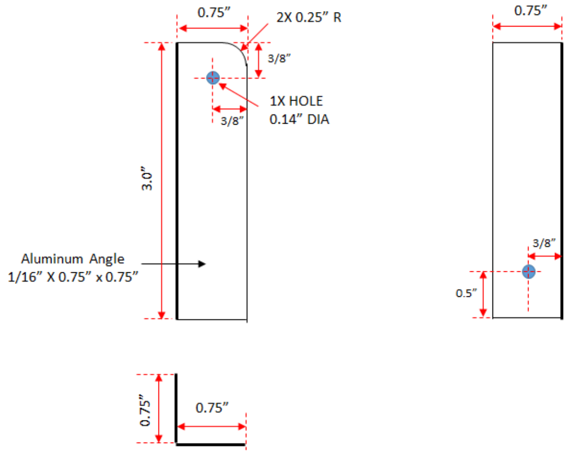

Figure 2.8 Drawings of Angle Bars for Tilt Angle Adjustment.

Figure 2.9 A Solar Panel with Mounting Assembly on a Signal Pole.

2.3.3 Installation of Bluetooth Devices

Figure 2.10 displays a few images showing that the Bluetooth enclosures and solar panels were attached to the traffic signal poles using zip ties at our test intersections. The entire assembly (solar panel and the Bluetooth device) weighs about 1 lb. The research team has installed 18 solar-power Bluetooth devices on light poles as shown in Figure 2.11 & 2.12. The red dots indicated the locations where Bluetooth devices have been installed. The research team was not able to find alternative solutions to attach the remaining 3 Bluetooth devices at 3 locations (blue dots indicated in Figure 2.12) where the Xcel Energy refused to permit attachment of our devices on their street light poles. Stores near the Blue dot locations also declined to provide contact information of their property owner.

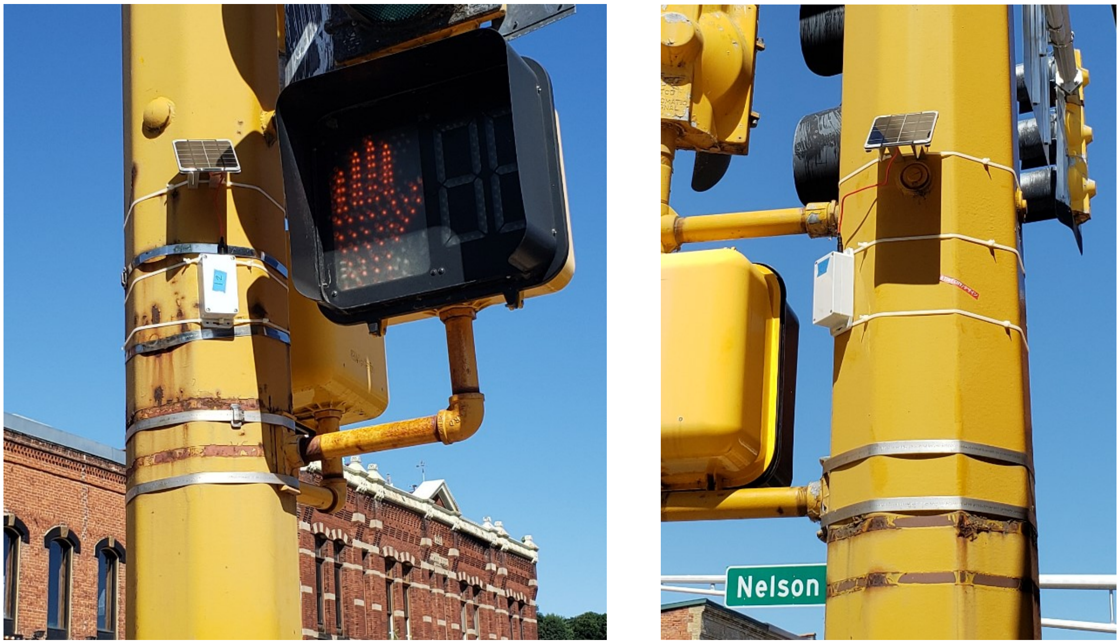

Figure 2.10 Images of Bluetooth Devices Attached to Signal Poles at Nelson St.

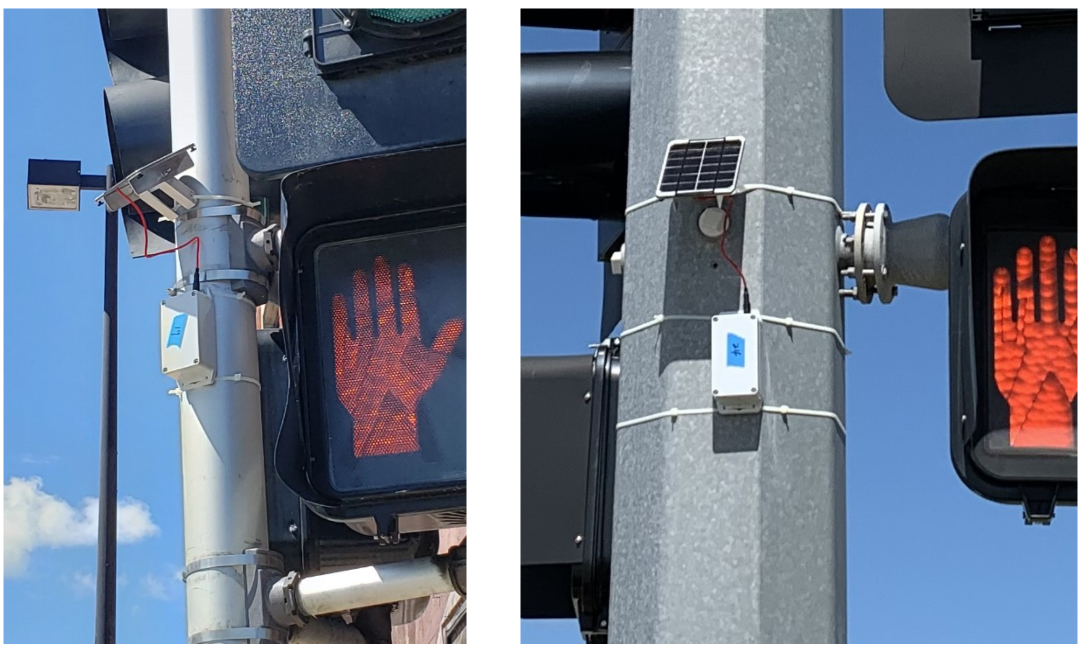

Figure 2.11 Images of Bluetooth Devices Attached to Signal Poles at Greeley & Hwy 36.

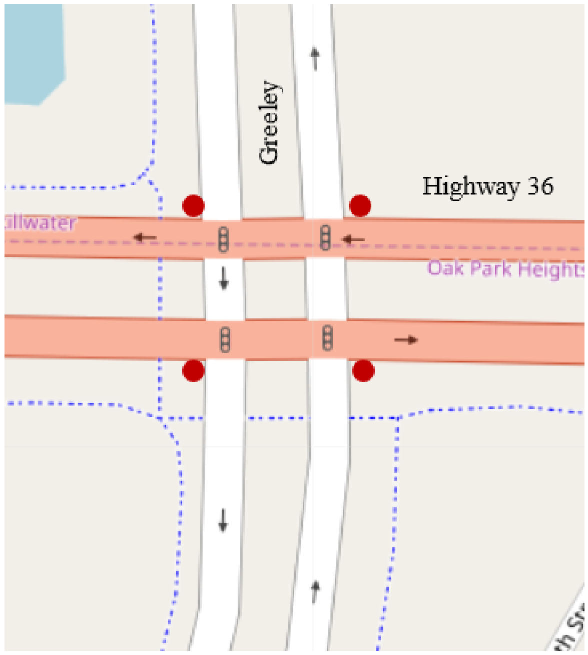

Figure 2.12 Location of Bluetooth Beacons Installed at Highway 36 and Greeley Intersection.

(Background Image from OpenStreetMap, 2020)

Figure 2.13 Location of Bluetooth Beacons Installed in Downtown Stillwater.

(Background Image from OpenStreetMap, 2020)

CHAPTER 3: SMARTPHONE APP

An Android app called *PedNav* was revised to provide wayfinding guidance to visually impaired pedestrians. It utilizes on-phone sensors to determine the user's position and heading. An Internet connection is used to communicate with an MQTT server that provides live signal phase information which is used to notify the user when the walk or don't walk sign is displayed. A map stores the positions of street corners and their associated crosswalks which is used by the app to determine which crosswalk's phase info to display to the user.

3.1 APP FUNCTIONALITY

3.1.1 Navigation View

The main screen of the app is the navigation view (Figure 3.1) which shows debug information and allows the user to interface with the screen through different taps and presses to get navigation information about their surroundings. For testing purposes, the debug information at the bottom of the screen includes the heading in degrees, latitude, longitude, and the provided location's estimated accuracy in meters. The "start logging" button begins logging application state data to a log file.

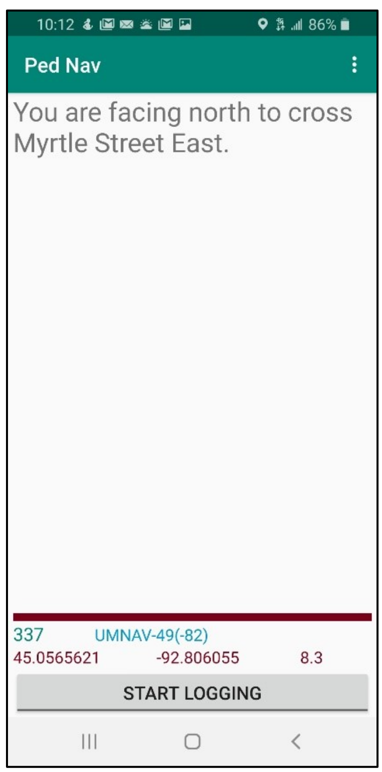

Figure 3.1 Sample text information.

Navigation feedback is provided to the user through text, a text-to-speech (TTS) voice message and when applicable, with icons. It is noted that the text-to-speech messages always match visually displayed text message. Except where noted below, messages are displayed visually for 5 seconds. Figure 3.1 shows a text only display. Figure 3.2 and Figure 3.3 shows a text display with an associated walk or don't walk icon.

There are three gestures that can be used to interact with the app on this screen. A single tap will prompt information about the crosswalk the user is currently facing which would result in a message such as "You are facing west to cross Main Street."

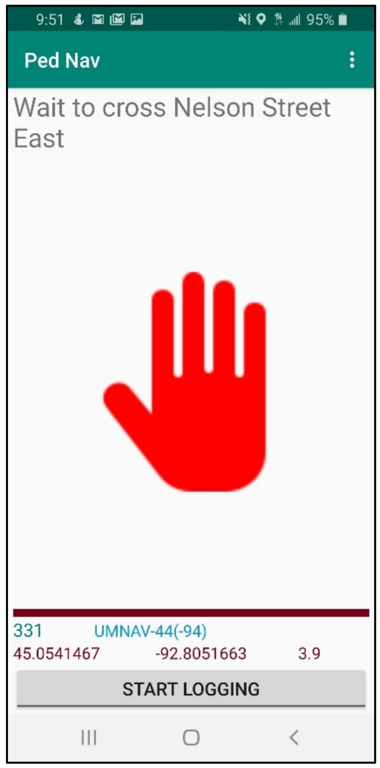

Figure 3.2 Don't walk text and icon.

A double tap prompts for information about the signal state for the crosswalk the user is currently facing. The app will either display information such as in Figure 3.2 or Figure 3.3. Here, the "don't walk" message is shown whether the real pedestrian signal is showing a solid or flashing "don't walk." When the user requests signal information and the crosswalk is in the flashing "don't walk" and "don't walk" phases, the "don't walk" message is shown on the screen until the crosswalk changes to the "walk" phase at which point the app updates to the "walk" message and then clears after 5 seconds.

A long press prompts for information about the user's current location. This would result in a message such as "You are at the north west corner of Union Street SE and Beacon Street SE."

Other possible text and voice feedback include messages warning the user when the phone can't determine their location, they aren't located at a known street corner, they aren't facing a known crosswalk, the crosswalk is unsignalized, or signal data is not currently available for a signalized crosswalk.

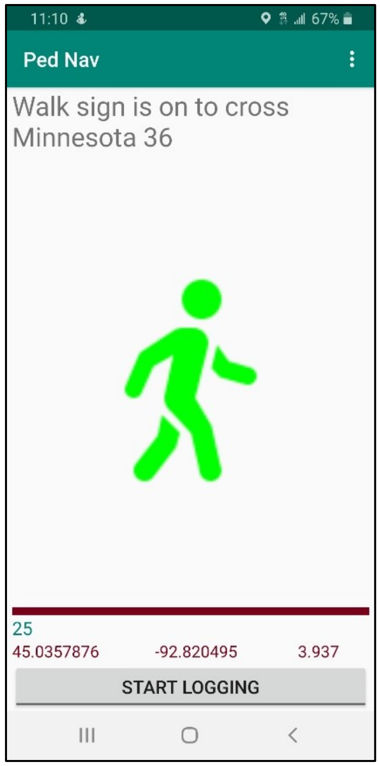

Figure 3.3 Walk sign text and icon.

3.1.2 Map View

A secondary, debug view is also available that shows the state of all known crosswalks superimposed on a Google Maps satellite view. This interface is helpful us to monitoring the pedestrian signal status at a system level. A sample of this view is shown in Figure 3.4. Here the crosswalks are drawn as line segments in green for the "walk" phase, yellow for the flashing "don't walk" phase, or red for the solid "don't walk" phase. Unsignalized crosswalks are drawn in black.

Figure 3.4 A Sample Map View.

3.2 APP COMPONENTS

The app architecture was designed to encapsulate similar functions within single classes such that no single class would need to perform multiple, disparate functions. This section describes the key components of the app noting that more detail can be found in the source code.

3.2.1 Data Object

3.2.1.1 NavigationUI

This is a data object that stores navigation activity UI elements which include the text to be displayed on the screen, the text to be played as speech, the icon/sign to be displayed (if any), and whether or not the screen should clear after displaying the information. It is used to pass UI updates to the activity responsible for managing the UI.

3.2.1.2 Node

This is a data object that represents the details associated with a single street corner. Here, the term node is used in the node-edge graph sense of the word noting that in the future there could be nodes that do not correspond to street corners. It stores key node/street corner information such as the associated Bluetooth MAC and device name, the intersection ID, and its latitude/longitude pair. The object also stores a list of the connected edges (i.e. crosswalks) and has a public method to return a particular crosswalk by heading. This method returns the crosswalk with the heading that most closely matches the provided heading but returns null if there is no crosswalk within 45 degrees.

3.2.1.3 Edge

This is a data object that represents the details associated with a single crosswalk. Here the term edge is used in the node-edge graph sense of the word noting that in the future, there could be edges that do not correspond to a crosswalk. It stores key edge/crosswalk information such as the signal phase and Miovision ID for the crosswalk and the cross street it traverses. It also contains a list of latitude - longitude geometry points corresponding to the nodes the edge connects. This is used for drawing the edges for the Map View. The Edge object also stores a heading value which is only used when considering an edge as belonging to a particular node. The heading value has no meaning when an edge is considered as a standalone entity.

3.2.2 Data Sources

3.2.2.1 LocationLiveData

This class extends the LiveData class such that when it is instantiated, it automatically connects to the operating system's FusedLocationProviderClient by submitting a LocationRequest. When the object is active (i.e. has an observer) it automatically requests location updates and conversely when it goes inactive (i.e. has no observers), it similarly stops location updates. When initialized, the class sets the location to null. This behavior is used to determine when a valid position has been determined. The class specifically extends LiveData<Location> which is an android.location.Location object wrapped in a LiveData object.

3.2.2.2 HeadingLiveData

This class extends the LiveData class such that when it is instantiated, it creates a SensorManager object. When active, the HeadingLiveData object uses the SensorManager to register itself as a listener to the Sensor.TYPE_ROTATION_VECTOR sensor which is used to calculate the phone's heading. Similarly, when inactive, the class unregisters itself as a listener to save system resources. When initialized, the class sets the initial heading to -1. This behavior is used to determine when a valid heading has been determined. Headings will always be in the range 0 to 359 degrees. The class specifically extends LiveData<Integer> which is an Integer wrapped in a LiveData object.

3.2.2.3 SignalClient

This class initializes the MQTT service that connects to the Miovision server in order to receive SPaT updates. This is done by the Paho library from Eclipse.org [37]. When the service receives an update, it triggers a callback that allows the SignalClient class to parse the message and extract the pedestrian phase information. These updates are exposed to listeners with a MutableLiveData <Map<String, String>> object. The map contains the state update for a single intersection at a time. The keys are the intersection ID combined with the phase using a hyphen (e.g. 12345-2). The values are the phases signal state represented by red, yellow, or green which correspond to “solid don’t walk”, “flashing don’t walk”, or “walk” signs respectively. The class also maintains the state of all signals in a non-live map that can be queried through the public getSignalState method.

3.2.2.4 DataRepository

This class abstracts access to the JSON files that contain the master list of edges and nodes. It uses Google’s GSON library to parse the JSON files into lists of Edge or Node objects. There is also a method that returns a map that links Miovision IDs to intersection IDs which is used in the Map View. In the future, this class could be expanded to include other data sources such as a persistent on-device database or external, network calls for data.

3.2.2.5 NearbyData

This class contains the list of all known Nodes and is used for querying for the closest node or the current node (i.e. the closest node that is within 10 meters). This class abstracts the process of searching through the list of nodes in addition to maintaining the list of nodes. In the future, this class could be responsible for tiling the current set of nearby data as a user moves through a city.

3.2.3 Activities and View Models

3.2.3.1 Map Activity

The Map View shows a Google Maps satellite view with superimposed lines that represent known crosswalks. They are colored red, yellow, or green corresponding to their respective signal states. The Google Maps fragment uses the research team’s Google Maps API. The Map View is controlled by MapActivity which handles displaying the map UI by observing the crosswalk phase updates Live Data provided by a SignalClient instance. On initialization, MapActivity queries SignalClient for the state of all signals to set the initial colors. Crosswalks with unknown signal states are colored blue and unsignalized crosswalks are colored black.

3.2.3.2 NavigationViewModel

This class serves as the view model for MainActivity (the main Navigation View) and contains most of the app’s business logic. It also instantiates most of the other helper classes. It communicates UI information to MainActivity.java through live data which MainActivity.java observes. The view model passes UI updates to the activity by exposing a MutableLiveData<NavigationUI> object that MainActivity observes.

3.2.3.3 MainActivity

This class handles the Navigation View's UI which includes both capturing the user input (i.e. gestures) and displaying information. All displayed information is updated asynchronously through LiveData observers that respond to changes in the observed data. All the LiveData instances that MainActivity observes are provided by the NavigationViewModel. The MainActivity itself doesn't instantiate anything but the view model. MainActivity creates an instance of TTSClient which is a simple helper class that handles the text-to-speech functions.

3.3 DATA REPRESENTATION

Data representing static information about the intersections, their street corners, and the crosswalks are stored in two ways, a no-redundancy relational database representation and an easily consumable document-style JSON representation.

3.3.1 Server-Side Data Representation

The goal of the server-side data representation is to model the data using a relational database schema that is easily maintainable. This means tables should represent a single type of object and not overlap fields with other tables. Additionally, tables should avoid excessive null or blank fields and instead split entities until sparse representations are not needed.

This representation is named the server-side data representation because it optimized for storage needs and query speed. In future iterations, this representation could easily accommodate spatial queries. Currently the "server" is approximated by storing the data in .csv files. The "API call" is approximated with a python script that transforms the data into a representation that is more suitable for serialization and consumption in the app. Presently, the data is processed using the script and then included in the source code for the app. Table 3.1 to 3.3 describe the schema for the server-side data representation.

Table 3.1 Sample data representation of intersections.

Intx_id	latitude	longitude	description	is_signalized
20646	45.054237	-92.805096	Nelson Street East & Main Street	TRUE
45054523	45.054543	-92.805257	Nelson Alley & Main Street	FALSE
45054912	45.05491	-92.805422	Olive Street East & Main Street	FALSE
20647	45.055791	-92.805792	Chestnut Street East & Main Street	TRUE
21353	45.05659	-92.806126	Myrtle Street East & Main Street	TRUE
20644	45.035985	-92.822186	Greeley Street South & MN-36	TRUE

Table 3.2 Sample data of the corners of an intersection.

intx_id	corner	bluetooth_name	latitude	longitude
20646	NE	UMNAV-42	45.054346	-92.805057
20646	SE	UMNAV-47	45.054206	-92.804982
20646	SW	UMNAV-44	45.054113	-92.805148
20646	NW	UMNAV-43	45.054235	-92.805219

Table 3.3 Sample data of intersection crosswalks.

intx_id	corner_1	corner_2	num_lanes	phase	cross_street
20646	NE	SE	2	2	Nelson Street East
20646	SE	SW	2	4	Main Street
20646	SW	NW	2	6	Nelson Street East
20646	NW	NE	2	8	Main Street

Table 3.4 lists the messages that are currently implemented in the PedNav app to provide navigational and signal information to pedestrians.

Table 3.4 List of implemented messages.

Message Name	Message
NO_LOCATION	Your location is not currently known.
NO_CROSSWALK	You are not facing a known crosswalk.
NO_CORNER	You are not at a known street corner.
NO_SIGNAL_DATA	Signal data is not currently available.
UNSIGNALIZED	Crosswalk is unsignalized.
UNKNOWN_ERROR	Unknown error.
CORNER_ANNOUNCEMENT	You are located at the [%s] corner of [%s].
CROSSWALK_ANNOUNCEMENT	You are facing [%s] to cross [%s].
WALK_ANNOUNCEMENT	Walk sign is on to cross [%s].
WAIT_ANNOUNCEMENT	Wait to cross [%s].

In the app, the data is represented in JSON [36] documents in order to maximize the ease of consumption and parsing into java objects. Here, data is repeated multiple times instead of linked using foreign keys in order to reduce the complexity of finding necessary information. Additionally, the models use “node” and “edge” to represent street corners and crosswalks respectively. The goal in using these terms is to allow for flexibility in the future, for example if an edge needed to correspond to a non-crosswalk path.

3.3.2 App Data Representation

The same data is represented differently in the app. Here, the data is represented in JSON documents in order to maximize the ease of consumption and parsing into java objects. Here, data is repeated multiple times instead of linked using foreign keys in order to reduce the complexity of finding necessary information. Additionally, the models use “node” and “edge” to represent street corners and crosswalks respectively. The goal in using these terms is to allow for flexibility in the future, for example if an edge needed to correspond to a non-crosswalk path.

3.3.2.1 Nodes

intersection_id	Intersection id
corner	Directional corner id
bluetooth_mac	MAC address of beacon

bluetooth_name	Beacon device name
location	
latitude	Latitude in decimal degrees
longitude	Longitude in decimal degrees
description	Textual description of intersection
edges (list of one or more)	
intersection_id	Intersection id
phase	Pedestrian crossing phase assignment
distance	Distance between nodes
heading	Bearing of crosswalk relative to parent node
cross_street	Cross street name
is_signalized	Boolean, indicating if intersection is signalized
miovision_id	Miovision ID for intersection

3.3.2.2 Edges

It is noted that the edge data object is used in two slightly different contexts. The first is when an edge is considered relative to a particular node (i.e. has a heading from the parent node to the partner/connected node) as is the case in the Node model. The other context is when an edge exists as a directionless segment connecting two hierarchically equivalent nodes as is the case in the Edge model. In both cases edges map to the same Edge object in the code. Fields are used or nulled based on what makes sense given the context.

intersection_id	Intersection id
phase	Pedestrian crossing phase assignment
is_signalized	Boolean, indicating if intersection is signalized
miovision_id	Miovision ID for intersection
geometry (list of two points, each with latitude and longitude)	
latitude	Latitude in decimal degrees
longitude	Longitude in decimal degrees

Figure 3.5 illustrates a sample data representation in JSON documents.

```
[ ...
{
  "intersection_id": 20646,
  "corner": "NE",
  "bluetooth_mac": "xxxx",
  "bluetooth_name": "UMNAV-42",
  "location": {
    "latitude": 45.054346,
    "longitude": -92.805057
  },
  "description": "Nelson Street East & Main Street",
  "edges": [
    {
      "intersection_id": 20646,
      "phase": 2,
      "distance": 17,
      "heading": 159,
      "cross_street": "Nelson Street East",
      "is_signalized": true,
      "miovision_id": "xxxx"
    },
    {
      "intersection_id": 20646,
      "phase": 8,
      "distance": 18,
      "heading": 226,
      "cross_street": "Main Street",
      "is_signalized": true,
      "miovision_id": "xxxx"
    }
  ]
},
... ]
```

Figure 3.5 Sample data representation in JSON format.

3.4 NAVIGATION VIEW ARCHITECTURE

A rough dependency graph for the main Navigation View is shown in Figure 3.6. Here arrows represent a “dependency on” or an “instantiates” relationship where the arrows origin creates an instance of the arrow’s destination. Custom classes are represented with blue, external dependencies are green, and system provided services are orange. An MQTT client library, called Paho Android Service [37], is implemented in the app to communicate with the Miovision MQTT server.

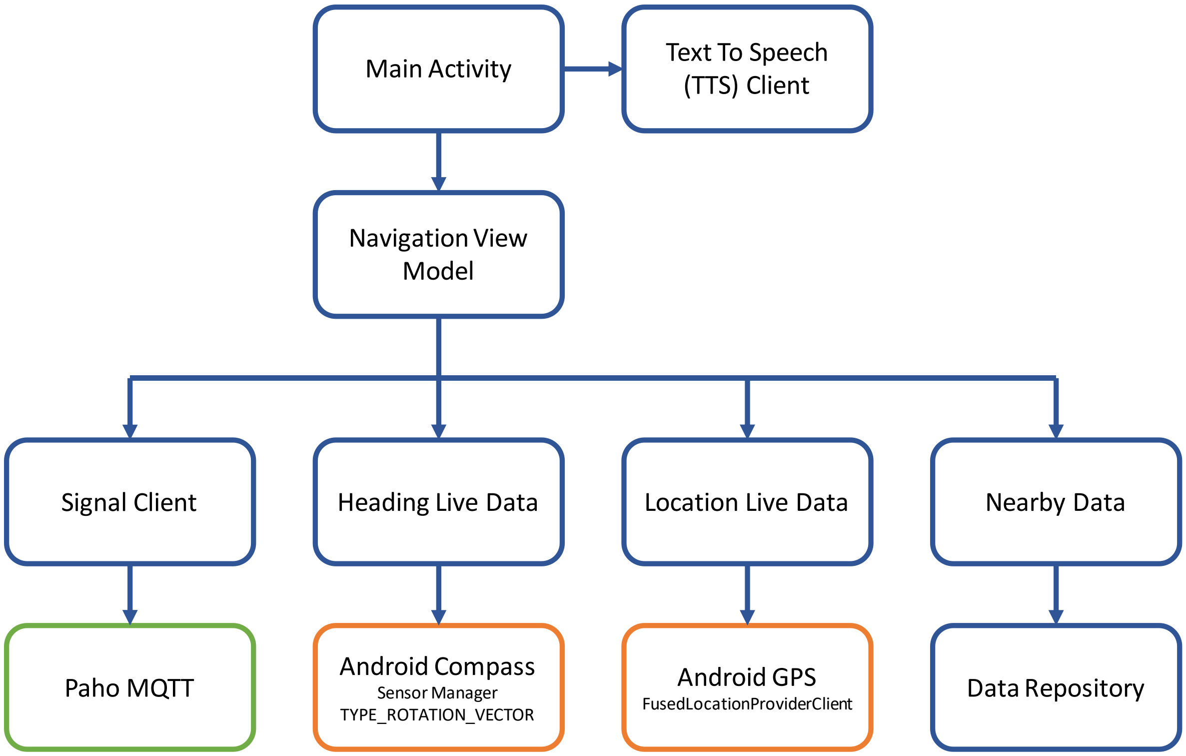

Figure 3.6 App Design Architecture.

CHAPTER 4: FIELD EXPERIMENTS AND SYSTEM VALIDATION

We tested the system at each corner of the intersections with an Android smartphone (Samsung Galaxy S9 running on Android 10) to collection location and Bluetooth data. At an intersection, we pointed the smartphone to directions where a crosswalk may or may not exist using then perform a single or double tap on the phone screen to validate the text and audible message presented to pedestrians.

4.1 LOCATION ACCURACY

We walked to the beginning of the ADA ramp at each intersection corner then started data collection and tested the audible messages in at least 3 different directions where a sidewalk may and may not be present. A list of test locations is included in Appendix A, Table A.1. The average distance from the test phone to the corresponding corner of an intersection is 3.8 m with a standard deviation of 1.8 m. The app was initially configured to display information to pedestrians when the phone is located within 10 m of an intersection corner.

The reported position accuracy at each test location ranges from 4 to 8 meters with an average reported accuracy of 5.6 m (see Appendix A, Table A.2). The reported accuracy obtained from the smartphone GPS sensor is a measure to indicate the quality of the GPS signal at each test location. The Bluetooth sensors are used as a supplement tool to identify the pedestrian's location when the GPS signal is poor.

4.2 MESSAGE VALIDATION

We also tested the correctness of text and audible messages displayed on the smartphone when performing a single and double tap. Overall, we performed 137 message correctness tests at 21 intersection corners and 5 randomly selected locations, for example, in the middle of a sidewalk. The system successfully provides 132 (96%) correct feedbacks on intersection geometry and signal status information during our test. We experienced five incorrect information (4%) that were presented to users. It is largely due to incorrect heading measured by the magnetometer sensor on the smartphone. The orientation information from the digital compass on the smartphone could be distorted when the phone is near a large ferrous metal object in the environment.

4.3 PEDESTRIAN SIGNAL LATENCY

Previously, broadcasting the signal phasing and timing (SPaT) information to the smartphone app had a data latency of 3 seconds when using the HTTP protocol. The MQTT protocol has reduced the data transmission delay to about 1 sec. To measure the data latency with the MQTT protocol, we used a camera to record the pedestrian walk phase transition on the pedestrian signal head and the display on the smartphone at different corner of the intersections. Figure 4.1 displays two snapshots of a video we recorded before and after the walk phase transition. Twelve videos were taken at 30 frames per second (FPS) for latency analysis. We later used software to analyze the video frame-by-frame and measure the time difference from the timestamp when the pedestrian walk light is turn on to the timestamp when the pedestrian walk sign icon on the app is displayed. Table 4.1 lists the timestamps of walk phase transitioning from off to on and the calculated SPaT data latency. On average, the MQTT protocol has a data latency of 0.88 sec with a standard deviation of 0.38 sec. The observed maximum latency is 1.3 sec at Main & Myrtle intersection and the minimum latency is 0.33 sec occurred at highway 36 & Greeley intersection.

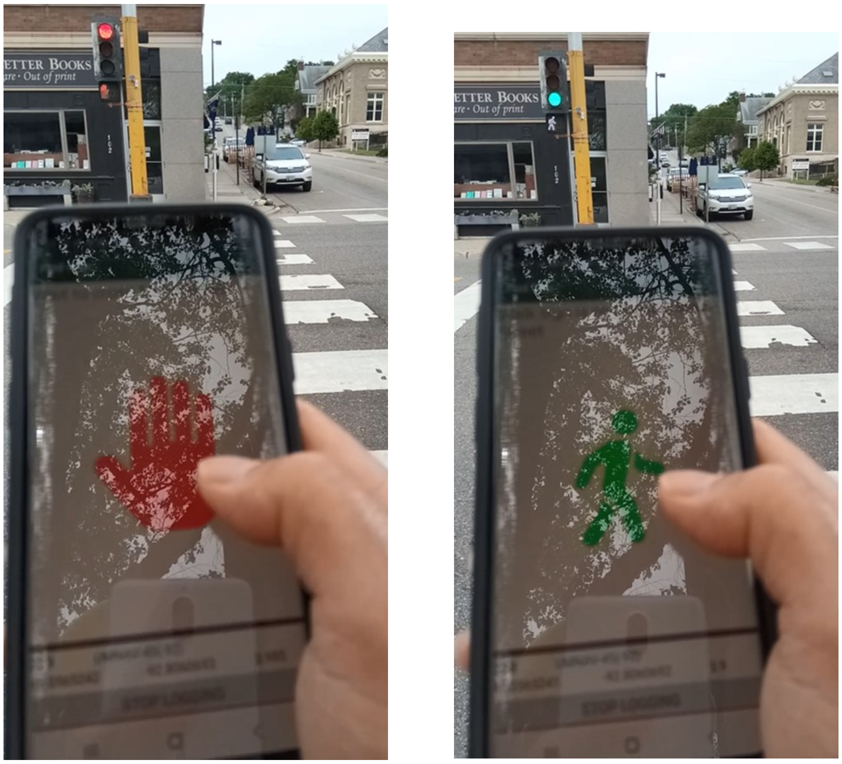

Figure 4.1 Snapshots of a video recording pedestrian walk phase transitions.

Table 4.1 Traffic Signal Data Latency with MQTT.

Video #	Video Frame Timestamp (sec)		MQTT Latency, $t_2 - t_1$ (sec)
	Ped Head Displays Walk Sign (t_1)	Smartphone App Displays Walk Sign (t_2)	
1	1.77	2.10	0.33
2	2.87	4.00	1.13
3	1.53	1.93	0.40
4	25.60	26.23	0.63
5	3.33	3.97	0.64
6	38.87	39.63	0.76
7	59.17	59.63	0.46
8	16.23	17.37	1.14
9	1.67	2.83	1.16
10	30.80	32.07	1.27
11	15.63	16.93	1.30
12	27.47	28.77	1.30

4.4 SYSTEM LIMITATION AND RECOMMENDATION

Our system is intended to use the smartphone as a personal assistive device to provide transportation information to people with vision impairment. However, it does not account for the differences in personal understanding and perception of an environment. Possible system limitations include: (1) when using the smartphone as a pointer to survey the environment for geometry information, users may not point the phone in the same direction as they are facing; (2) the user's cognitive understanding of the environment and provided messages may vary; and (3) the orientation reading from the digital compass on the smartphone may be influenced by a large ferrous metal object in the environment.

The PedNav app relies on the smartphone sensors to determine a user's location and orientation. In our field tests, we noticed 5 instances that the app was not able to correctly provide the geometry and signal timing information when the magnetic sensor on the phone is temporarily affected by the interference.

Modern smartphones use magnetic sensor, called magnetometer, to measure the strength and direction of magnetic fields surrounding the phone. By analyzing Earth's magnetic field, the sensor allows the phone to determine its orientation. However, the magnetometer in the smartphone must contend with multiple radios within centimeters emitting electromagnetic radiation and metallic objects in the environment which can cause reflections. In general, there are hard and soft iron errors that can really affect compass readings from the magnetometer sensors. Hard iron effects are a constant, static interference applied to the compass. This constant bias is usually due to how the magnetometer is installed in the phone. It's also easy to fix by a simple realignment, and only needs to be done once typically. Soft iron errors are dynamic effects, such as walking past a large metal or something that affects the local magnetic field. Calibrations can be done on-demand if there is some interference. But, soft iron can happen in many places, and can significantly affect the heading readings.

With this known system limitations, we recommend: (1) avoid placing the phone too close to the signal pole while requesting intersection geometry and signal timing information; (2) periodically perform a sensor calibration in an open space as recommended by Google [38].

CHAPTER 5: SUMMARY AND DISCUSSION

This chapter summarizes the research scope and methodology of this project. Additional discussion on potential safety and mobility application of our system with the Connected Vehicle (CV) initiative is also included.

5.1 SUMMARY

The research team worked with MnDOT staff to identify 4 signalized and 2 un-signalized interactions in Stillwater, Minnesota, for installing the smartphone-based traffic information systems. The SmartLink controller from Miovision is installed in each controller cabinet. The SmartLink system has the capability of monitoring and managing traffic signal control remotely. It enables us to collect and broadcast SPaT data for our application to provide pedestrian signal information for people with vision impairment.

A HTTP-based interface was initially developed to transmit live signal and timing information from the 4 traffic controllers through the Miovision SmartLink devices. However, the HTTP protocol has a data latency over 3 seconds due to the data overhead of the protocol. The data transmission latency varied depending on the Internet network traffic. To address the latency issue, a Message Queuing Telemetry Transport (MQTT) interface was implemented to reduce data transmission latency.

An Android app called *PedNav* was revised to provide wayfinding guidance to visually impaired pedestrians. It uses smartphone sensors to determine the user's position and heading. An Internet connection is used to communicate with an MQTT server that provides live signal phase information that is used to notify the user when the walk or don't walk sign is displayed. A digital map was also generated for the app to determine which crosswalk's SPaT information to display to the user.

To validate the system performance in supporting wayfinding and navigation for the visually impaired, the team tested the system at each of the corners of the intersections with an Android smartphone to collect location and Bluetooth data. For example, after arriving at an intersection, the user points the smartphone in directions where a crosswalk may or may not exist then performs a single or double tap on the phone screen to respectively request intersection geometry or signal information. Experiments were conducted to validate the text and audible message presented to pedestrians.

The reported position accuracy at each test location ranges from 4 to 8 meters with an average reported accuracy of 5.6 meters. The reported accuracy obtained from the smartphone GPS sensor is a measure to indicate the quality of the GPS signal at each test location. The team also tested the correctness of text and audible messages displayed on the smartphone when performing a single or double tap. Out of 137 message correctness tests, the system successfully provided 132 (96%) correct feedbacks on intersection geometry and signal status information. Incorrect information was provided five times (4%) to users due to incorrect headings measured by the magnetometer sensor (i.e., digital compass) on the smartphone. The orientation information from the digital compass on the smartphone could be distorted when the phone is near a large ferrous metal object in the environment.

To evaluate the MQTT data latency, the research team used a video camera to record the pedestrian walk phase transition on the pedestrian signal head and the display on the smartphone at different corners of the intersections. The MQTT protocol had a data latency of 0.88 sec with a standard deviation of 0.38 sec. The observed maximum latency was 1.3 sec at the Main and Myrtle intersection and the minimum latency was 0.33 sec occurred at the Highway 36 and Greeley intersection.

The mobile traffic information system is intended to use the smartphone as a personal assistive device to provide transportation information to people with vision impairment. Possible system limitations include: (1) when using the smartphone as a pointer to survey the environment for geometry information, users may not point the phone in the same direction as they are facing; and (2) the orientation reading from the digital compass on the smartphone may be influenced by large ferrous metal objects in the environment.

5.2 DISCUSSION

Connected Vehicle (CV) technology has enabled instantaneous and dedicated communications among the vehicles, infrastructures, and pedestrians for a plethora of safety and mobility applications. In a CV environment, a vehicle can wirelessly exchange critical safety information with other vehicles, nearby infrastructure (traffic intersections, signs, service facilities, etc.), and non-motorists such as pedestrians and cyclists. According to USDOT, the CV environment currently includes three major approaches to communication, i.e., Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), and Vehicle to Pedestrian (V2P) [39]. Our system demonstrates an application that establishes Infrastructure to Pedestrian (I2P) communication and supports the mobility and safety for people with disability at traffic intersections.

One V2P application could potentially use our system to provide safety-related vehicle information to pedestrians who are waiting at a corner of an intersection. For example, a local dedicated communication system at an intersection can rebroadcast a CV Basic Safety Message (BSM) [40 & 41] received from vehicles in the MQTT protocol that our system can subscribe to and then alert pedestrians of any left- or right-turning vehicles approaching the intersection corner (See Figure 5.1).

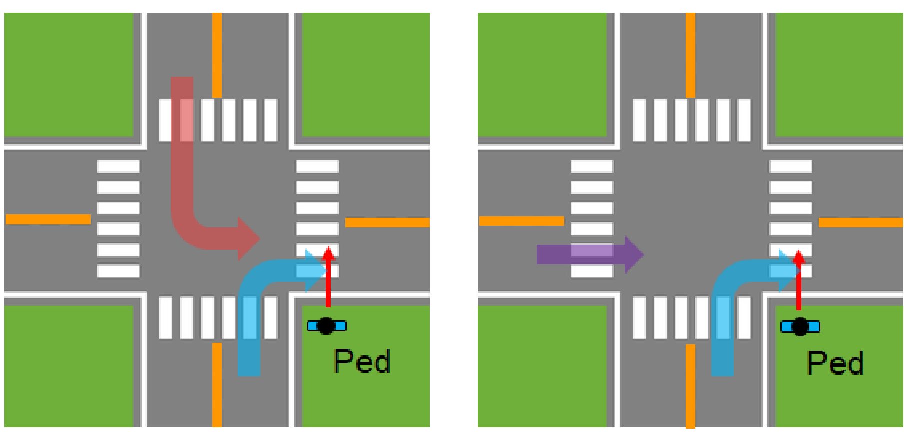

Figure 5.1 Examples of potential vehicles and pedestrians conflict at an intersection.

Similarly, the location of a pedestrian using our system (waiting at a specific corner of an intersection) can be sent to the local communication server that transmits pedestrian related BSM messages through the Infrastructure to Vehicle (I2P) communication protocol to inform approaching drivers about the presence of a pedestrian who is intending to enter the crosswalk.

This proposed approach can enable state, county, or local agencies to provide a more complete and accessible solution at both signalized and un-signalized intersections to the visually impaired, thus improving their mobility and independence in using the transportation system. In addition, the mobile APS approach can also be applied to provide the visually impaired with bus arrival information at bus stops and light rail stations or detour information for a work zone on the sidewalk. The FHWA is conducting a Work Zone Data Initiative (WZDI) to facilitate the work zone data exchange [42] and management for application related to safety and mobility. In the future, our system could be expanded to interface with the WZDI and other ITS data standards.

REFERENCES

[1] World Health Organization (WHO). (2019, October 8). Blindness and vision impairment. Retrieved from https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] National Health Interview Survey (NHIS). (2017). Data release. Retrieved from https://www.cdc.gov/nchs/nhis/nhis_2017_data_release.htm

[3] American Foundation for the Blind (AFB). (2020, August). Statistical snapshots from the American Foundation for the Blind. Retrieved from https://www.afb.org/research-and-initiatives/statistics#:~:text=For%20example%2C%20findings%20from%20the,or%20unable%20to%20see%20at

[4] National Federation of the Blind (NFB). (2020, August). Blindness statistics. Retrieved from https://www.nfb.org/resources/blindness-statistics

[5] Gaunet, F., & Briffault, X. (2005). Exploring the functional specifications of a localized wayfinding verbal aid for blind pedestrians. Simple and structured urban areas. *Hum. Comput. Interact., 20*, 267–314

[6] Blades, M. (1991). Wayfinding theory and research: The need for a new approach. In Mark, D.M., Frank, A.U. (eds.) *Cognitive and Linguistic Aspects of Geographic Space* (pp. 137–165). Heidelberg, Netherlands: Springer.

[7] Golledge, R. G. (1999). *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. Baltimore: Johns Hopkins University Press.

[8] Golledge, R. G., & Gärling, T. (2004). Cognitive maps and urban travel. In D. A. Hensher, K. J. Button, K. E. Haynes & P. R. Stopher (Eds), *Handbook of Transport Geography and Spatial Systems*. Amsterdam: Elsevier.

[9] Giudice, N. A. & Legge, G. E. (2008). Blind navigation and the role of technology. In A. Helal, M. Mokhtari, & B. Abdulrazak (Eds), *The Engineering Handbook of Smart Technology for Aging, Disability, and Independence.* Hoboken, NJ: John Wiley & Sons, Inc.

[10] Ponchillia, P., Rak, E., Freeland, A., & LaGrow, S. (2007). Accessible GPS: Reorientation and target location among users with visual impairments. *Journal of Visual Impairment & Blindness, 101*(7), 389–401.

[11] Barlow, J. M., Bentzen, B. L., & Bond, T. (2005). Blind pedestrians and the changing technology and geometry of signalized intersections: Safety, orientation, and independence. *J. of Visual Impairment & Blindness, 99*(10), 587–598.

[12] Guth, D. (2007). Why does training reduce blind pedestrians' veering? In J. J. Rieser, D. H. Ashmead, F. Ebner & A. L. Corn (Eds.), *Blindness and Brain Plasticity in Navigation and Object Perception* (pp. 353-365). New York: Taylors & Francis Group.

[13] Bourquin, E., Wall Emerson, R., Sauerburger, D., & Barlow, J. (2018). Conditions that influence drivers' behavior at a roundabout: Increasing yielding for pedestrians who are visually impaired. *Journal of Visual Impairment & Blindness, 112*(1), 61–71.

[14] Bourquin, E., Wall Emerson, R., Sauerburger, D., & Barlow, J. (2017). The effect of the color of a long cane used by individuals who are visually impaired on the yielding behavior of drivers. *Journal of Visual Impairment & Blindness, 111*(5), 401–410.

[15] Bourquin, E., Wall Emerson, R., Sauerburger, D., & Barlow, J. (2016). Conditions that influence drivers' yielding behavior: Effects of pedestrian gaze and head movements. *International Journal or Orientation and Mobility, 8*(1), 13–26.

[16] Bourquin, E., Wall Emerson, R., Sauerburger, D., & Barlow, J. (2014). Conditions that influence drivers' yielding behavior in turning vehicles at intersections with traffic signal controls. *Journal of Visual Impairment & Blindness*. Volume: 108 issue: 3, 173-186 https://journals.sagepub.com/doi/10.1177/0145482X1410800302

[17] Rose, C., Pierce, D., & Gao, S. (2016). Blind Pedestrian Body-Worn Navigational Aid Based on Pedometry and Smart Intersection Connectivity, *Compendium of papers, Transportation Research Board 95th Annual Meeting*, Washington, D.C.

[18] Tian, Y. (2014). RGB-D sensor-based computer vision assistive technology for visually impaired persons. In L. Shao, J. Han, P. Kohli, Z. Zhang (Eds), *Computer Vision and Machine Learning with RGB-D Sensors* (pp.173-191). Springer, Cham. https://doi.org/10.1007/978-3-319-08651-4_9

[19] Yi, C., Tian, Y., & Arditi, A. (2014). Portable camera-based assistive text and product label reading from hand-held objects for blind persons. *IEEE/ASME Transactions on Mechatronics, 19*(3) 808-817.

[20] Tian, L., Tian, Y. and Yi, C. (2013). "Detecting good quality frames in videos captured by a wearable camera for blind navigation," 2013 IEEE International Conference on Bioinformatics and Biomedicine, Shanghai, pp. 334-337, doi: 10.1109/BIBM.2013.6732513.

[21] Real, S., & Araujo, A. (2019). Navigation systems for the blind and visually impaired: Past work, challenges, and open problems. *Sensors, 19*(15), 3404. https://doi.org/10.3390/s19153404

[22] Coughlan, J. M., & Shen, H. (2013). Crosswatch: A system for providing guidance to visually impaired travelers at traffic intersections. *Journal of assistive technologies, Vol. 7 No. 2, pp. 131-142. https://doi.org/10.1108/17549451311328808*

[23] Fusco, G., Shen, H., Murali, V., & Coughlan, J. M. (2014). Determining a blind pedestrian's location and orientation at traffic intersections. *Computers helping people with special needs: International Conference on Computers Helping People with Special Needs, 8547*, 427–432.

[24] Liao, C.-F. (2013). Using smartphone app to support visually impaired pedestrians at signalized intersection crossings. *Transportation Research Record, 2393,* 12-20.

[25] Liao, C.-F. (2014). *Development of a Navigation System Using Smartphone and Bluetooth Technologies to Help the Visually Impaired Navigate Work Zones Safely* (Report #2014-12). St. Paul, MN: MnDOT.

[26] Liao, C.-F. & Donath, M., (2019). A Self-Monitoring Network to Support Situation Awareness and Navigation for the Visually Impaired in GNSS Unfriendly Environments. Paper presented at the 26th ITS World Congress, October 21-25, 2019, Singapore.

[27] Miovision. (2020, August). SmartLink hardware. Retrieved from https://miovision.com/trafficlink/hardware/

[28] Message Queue Telemetry Transport (MQTT) (2020, August). Retrieved from http://mqtt.org/

[29] Internet of Things (IoT). (2020, August). Retrieved from https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/

[30] Miovision. (2020, August). TrafficLink MQTT API documentation. Retrieved from https://help.miovision.com/s/article/Miovision-TrafficLink-APIs?language=en_US#MQTT

[31] Adafruit. (2020, August). USB/DC/Solar Lithium Ion/Polymer charger. Retrieved from https://www.adafruit.com/product/390

[32] Adafruit. (2020, August). Lithium Ion Battery Pack - 3.7V 4400mAh. Retrieved from https://www.adafruit.com/product/354

[33] NORDIC. (2020, August). nF52832 product specifications. Retrieved from https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf

[34] Adafruit. (2020, August). Feather nRF52 Bluefruit LE - nRF52832. Retrieved from https://www.adafruit.com/product/3406

[35] PolyCase. (2020, August). WP-21F polycarbonate NEMA enclosure. Retrieved from https://www.polycase.com/wp-21f

[36] JavaScript. (2020, August). Object notation (JSON) is a lightweight data-interchange format. Retrieved from https://www.json.org/json-en.html

[37] Eclipse Paho Android Service. (2020, August). MQTT library. Retrieved from https://www.eclipse.org/paho/clients/android/

[38] Google. (2020, August). Find and improve your location's accuracy. Retrieved from https://support.google.com/maps/answer/2839911?co=GENIE.Platform%3DAndroid&hl=en

[39] USDOT. (2020, August). Connected vehicle: Vehicle-to-Pedestrian (V2P) communications. Retrieved from https://www.its.dot.gov/factsheets/cv_v2pcomms.htm

[40] USDOT (2020, August). Vehicle based data and availability. Retrieved from https://www.its.dot.gov/itspac/october2012/PDF/data_availability.pdf

[41] SAE International. (2020, August). Dedicated short-range communications (DSRC) message set dictionary. Retrieved from https://www.sae.org/standards/content/j2735_200911/

[42] FHWA. (2020, August). Work Zone Data Exchange (WZDx). Retrieved from https://www.transportation.gov/av/data/wzdx

APPENDIX A
PHONE GPS DATA

Table A.1 Phone test location and its distance to the corresponding intersection corner.

Location ID	Intx Description	Intx Corner	Ped Facing	Phone Heading (Deg)	Intersection Corners (MN S-2203)		Phone Location (MN S-2203)		Distance (m)
					X	Y	X	Y	
UMNAV-42	Main & Nelson	NE	S	172	894122.055	328954.121	894121.361	328953.076	1.25
UMNAV-42	Main & Nelson	NE	SW	261	894122.055	328954.121	894119.359	328956.083	3.33
UMNAV-47	Main & Nelson	SE	NW	324	894128.189	328938.651	894126.045	328937.838	2.29
UMNAV-47	Main & Nelson	SE	SW	258	894128.189	328938.651	894126.083	328940.632	2.89
UMNAV-44	Main & Nelson	SW	NW	338	894115.266	328928.126	894114.466	328930.276	2.29
UMNAV-44	Main & Nelson	SW	NE	48	894115.266	328928.126	894117.412	328928.229	2.15
UMNAV-43	Main & Nelson	NW	NE	30	894109.476	328941.600	894107.444	328946.442	5.25
UMNAV-43	Main & Nelson	NW	SE	158	894109.476	328941.600	894109.195	328937.913	3.70
UMNAV-29	Main & Chestnut	SE	N	340	894072.211	329108.879	894074.537	329108.722	2.33
UMNAV-29	Main & Chestnut	SE	W	260	894072.211	329108.879	894070.069	329106.117	3.50
UMNAV-30	Main & Chestnut	NE	W	265	894066.950	329123.917	894065.640	329123.927	1.31
UMNAV-30	Main & Chestnut	NE	S	149	894066.950	329123.917	894070.707	329121.568	4.43
UMNAV-41	Main & Chestnut	NW	S	158	894051.501	329119.135	894048.991	329118.299	2.65
UMNAV-41	Main & Chestnut	NW	E	74	894051.501	329119.135	894053.439	329120.613	2.44
UMNAV-28	Main & Chestnut	SW	NW	338	894056.291	329103.978	894058.131	329109.350	5.68
UMNAV-28	Main & Chestnut	SW	E	71	894056.291	329103.978	894057.626	329105.125	1.76
UMNAV-49	Main & Myrtle	SE	N	338	894042.937	329198.476	894043.675	329200.780	2.42
UMNAV-49	Main & Myrtle	SE	W	243	894042.937	329198.476	894039.042	329194.514	5.56
UMNAV-48	Main & Myrtle	NE	W	220	894039.589	329211.986	894036.465	329209.272	4.14
UMNAV-48	Main & Myrtle	NE	S	209	894039.589	329211.986	894036.630	329209.979	3.58
UMNAV-46	Main & Myrtle	NW	SE	141	894023.436	329206.860	894026.234	329202.055	5.56
UMNAV-46	Main & Myrtle	NW	NE	69	894023.436	329206.860	894025.953	329206.281	2.58
UMNAV-45	Main & Myrtle	SW	E	62	894027.183	329193.022	894026.724	329193.209	0.50
UMNAV-45	Main & Myrtle	SW	NW	318	894027.183	329193.022	894023.710	329196.143	4.67
UMNAV-50	Main & Olive	NE	SW	240	894097.336	329028.335	894092.019	329021.895	8.35
UMNAV-50	Main & Olive	NE	SW	231	894097.336	329028.335	894097.776	329029.373	1.13
UMNAV-91	Main & Olive	NW	SE	129	894082.961	329020.123	894087.434	329017.972	4.96
UMNAV-91	Main & Olive	NW	NE	73	894082.961	329020.123	894091.860	329018.197	9.11
UMNAV-92	Main & Olive	SW	N	356	894086.694	329007.285	894089.198	329007.412	2.51
UMNAV-93	Main & Nelson Aly.	NW	S	150	894096.425	328977.530	894099.391	328975.682	3.49
UMNAV-94	Main & Nelson Aly.	SW	N	360	894098.837	328968.785	894100.057	328969.777	1.57
UMNAV-10	Hwy 36 & Greeley	NE	W	277	892931.725	326917.405	892930.984	326915.961	1.62
UMNAV-10	Hwy 36 & Greeley	NE	S	187	892931.725	326917.405	892933.448	326915.515	2.56
UMNAV-21	Hwy 36 & Greeley	NE	S	153	892883.327	326918.263	892883.369	326916.420	1.84
UMNAV-21	Hwy 36 & Greeley	NE	E	87	892883.327	326918.263	892882.815	326920.257	2.06
UMNAV-24	Hwy 36 & Greeley	SW	E	85	892887.284	326873.419	892889.016	326871.288	2.75
UMNAV-24	Hwy 36 & Greeley	SW	N	361	892887.284	326873.419	892883.496	326874.962	4.09
UMNAV-27	Hwy 36 & Greeley	SE	N	358	892933.192	326870.302	892935.381	326872.338	2.99
UMNAV-27	Hwy 36 & Greeley	SE	W	279	892933.192	326870.302	892931.555	326869.081	2.04

Table A.2 Reported GPS position precision at each intersection corner.

Location ID	Intersection Description	Intersection Corner	Pedestrian Facing	Phone Heading (Degree)	Reported GPS Accuracy AVG (m)	Accuracy STDEV (m)	Sample Size (N)
UMNAV-42	Main & Nelson	NE	S	172	4.4	0.6	137
UMNAV-42	Main & Nelson	NE	SW	261	4.0	0.2	23
UMNAV-47	Main & Nelson	SE	NW	324	4.7	1.2	141
UMNAV-47	Main & Nelson	SE	SW	258	4.9	1.6	34
UMNAV-44	Main & Nelson	SW	NW	338	4.0	0.2	211
UMNAV-44	Main & Nelson	SW	NE	48	7.6	3.4	38
UMNAV-43	Main & Nelson	NW	NE	30	4.0	0.1	368
UMNAV-43	Main & Nelson	NW	SE	158	7.8	2.4	25
UMNAV-29	Main & Chestnut	SE	N	340	4.0	0.2	176
UMNAV-29	Main & Chestnut	SE	W	260	7.1	2.9	40
UMNAV-30	Main & Chestnut	NE	W	265	5.3	0.2	70
UMNAV-30	Main & Chestnut	NE	S	149	6.7	2.7	62
UMNAV-41	Main & Chestnut	NW	S	158	3.9	0.1	170
UMNAV-41	Main & Chestnut	NW	E	74	6.4	2.7	37
UMNAV-28	Main & Chestnut	SW	NW	338	4.3	0.4	172
UMNAV-28	Main & Chestnut	SW	E	71	6.3	3.2	41
UMNAV-49	Main & Myrtle	SE	N	338	4.0	0.1	380
UMNAV-49	Main & Myrtle	SE	W	243	6.1	2.3	88
UMNAV-48	Main & Myrtle	NE	W	220	4.9	2.2	372
UMNAV-48	Main & Myrtle	NE	S	209	7.7	3.5	30
UMNAV-46	Main & Myrtle	NW	SE	141	3.9	0.1	480
UMNAV-46	Main & Myrtle	NW	NE	69	7.3	2.8	41
UMNAV-45	Main & Myrtle	SW	E	62	4.6	1.3	370
UMNAV-45	Main & Myrtle	SW	NW	318	7.5	3.1	42
UMNAV-50	Main & Olive	NE	SW	231	6.0	2.5	31
UMNAV-91	Main & Olive	NW	SE	129	6.8	2.6	31
UMNAV-91	Main & Olive	NW	NE	73	4.9	2.0	41
UMNAV-92	Main & Olive	SW	N	356	5.1	2.0	27
UMNAV-93	Main & Nelson Alley	NW	S	150	4.0	0.2	56
UMNAV-94	Main & Nelson Alley	SW	N	360	6.4	2.6	60
UMNAV-10	Hwy 36 & Greeley	NE	W	277	5.0	2.0	124
UMNAV-10	Hwy 36 & Greeley	NE	S	187	5.4	2.5	81
UMNAV-21	Hwy 36 & Greeley	NE	S	153	5.5	2.2	101
UMNAV-21	Hwy 36 & Greeley	NE	E	87	7.3	2.8	72
UMNAV-24	Hwy 36 & Greeley	SW	E	85	4.4	1.5	43
UMNAV-24	Hwy 36 & Greeley	SW	N	361	6.2	2.8	92
UMNAV-27	Hwy 36 & Greeley	SE	N	358	4.3	1.0	88
UMNAV-27	Hwy 36 & Greeley	SE	W	279	7.2	3.1	34