SOUTHERN PLAINS
TRANSPORTATION CENTER

Enhancing Driver Safety During Severe Weather Conditions

Mohammed Atiquzzaman, Ph.D.
Ronald Barnes, Ph.D.
Joebob Havlicek, Ph.D.

SPTC14.1-03-F

Southern Plains Transportation Center
201 Stephenson Parkway, Suite 4200
The University of Oklahoma
Norman, Oklahoma 73019

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENTS CATALOG NO.
SPTC14.1-03-F		

4. TITLE AND SUBTITLE Enhancing driver safety during severe weather conditions	5. REPORT DATE November 30, 2019 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) M. Atiquzzaman, R. Barnes and J. Havlicek	8. PERFORMING ORGANIZATION REPORT
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Computer Science The University of Oklahoma Norman, OK 73019	10. WORK UNIT NO. 11. CONTRACT OR GRANT NO. DTRT13-G-UTC36
12. SPONSORING AGENCY NAME AND ADDRESS Southern Plains Transportation Center 201 Stephenson Pkwy, Suite 4200 The University of Oklahoma Norman, OK 73019	13. TYPE OF REPORT AND PERIOD COVERED Final Oct 2014 – Nov 2019 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Matching funds was provided by Oklahoma Department of Transportation

16. ABSTRACT

The objective of this project was to reduce vehicle crashes, fatalities and injuries due to adverse weather conditions, by alerting drivers in real-time of potentially hazardous road conditions in the region, based on information from neighboring vehicles. The concept was demonstrated through a working prototype of a vehicular network using the connected vehicles technology to increase driver safety.

The project consisted of eight tasks which were all completed successfully. We evaluated a number of vendors for purchase of the vehicular networking hardware and finally decided to buy the equipment from Arada Systems. A vehicle interface was programmed with the latest available source-code and pre-built binary to translate known messages on the Controller Area Network (CAN) bus in a vehicle. Data was captured from the two vehicles owned by the Center for Intelligent Transportation; the data were transferred to laptops using Bluetooth. It was found that additional information from the vehicle control system would benefit the situational awareness in our system and we have been in contact with Ford Motor Company to obtain access to proprietary codes needed to obtain traction control and anything else that will enhance the system. We developed algorithms to take into account the hazardous condition encountered by drivers and alerting neighboring drivers of such hazardous conditions.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Connected vehicles, DSRC	No restrictions. This publication is available at www.sptc.org and from the NTIS.

19. SECURITY CLASSIF. (OF THIS REPORT)	20. SECURITY CLASSIF. (OF THIS PAGE)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	31 + cover	

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl	fluid ounces	29.57	milliliters	m
oz	gallons	3.785	liters	L
gal	cubic feet	0.028	cubic meters	L
ft^3	cubic yards	0.765	cubic	m^3
yd^3	meters NOTE: volumes greater than 1000 L shall be			m^3
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 Celsius or (F-32)/1.8		°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals square ins	0.145	poundforce per h	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

ACKNOWLEDGEMENT

The authors would like to express their appreciation to the Southern Plains Transportation Center (SPTC) for their support of this project and Oklahoma Department of Transportation for providing matching funds to carry out the project.

Enhancing driver safety during severe weather conditions

Final report

November 30, 2019

Dr. Mohammed Atiquzzaman
Professor

Dr. Ronald Barnes
Professor

Dr. Joebob Havlicek
Professor

School of Computer Science
School of Electrical and Computer Engineering
The University of Oklahoma
Norman, OK 73019

Contents

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

This document contains details of the work accomplished during the period of the project (Oct 2014 – Jun 2017). The objective of this project was to reduce vehicle crashes, fatalities and injuries due to adverse weather conditions, by alerting drivers in real-time of potentially hazardous road conditions in the region, based on information from neighboring vehicles. The concept was demonstrated through a working prototype of a vehicular network using the connected vehicles technology to increase driver safety.

The project consisted of eight tasks which were all completed successfully. We evaluated a number of vendors for purchase of the vehicular networking hardware and finally decided to buy the equipment from Arada Systems. A vehicle interface was programmed with the latest available source-code and pre-built binary to translate known messages on the Controller Area Network (CAN) bus in a vehicle. Data was captured from the two vehicles owned by the Center for Intelligent Transportation; the data were transferred to laptops using Bluetooth. It was found that additional information from the vehicle control system would benefit the situational awareness in our system and we have been in contact with Ford Motor Company to obtain access to proprietary codes needed to obtain traction control and anything else that will enhance the system. We developed algorithms to take into account the hazardous condition encountered by drivers and alerting neighboring drivers of such hazardous conditions.

1. Introduction

1.1. Background

Modern vehicles are equipped with lots of sensors for measurement of vehicle operating conditions and the surrounding, including weather conditions, and can be a viewed as a web of sensors on wheels. They can sense a range of information about the vehicle, such as location, speed, braking intensity, road traction, etc., some of which can represent road weather conditions. Lots of crashes happen due to the driver being unware of the surrounding road weather conditions, such as icy patches and frozen pavement. By facilitating vehicles within an area to exchange information between themselves in real-time, the drivers can be instantly alerted about road hazards and possibly avoid potential crashes (RITA 2014). The *goal* of this proposal was to increase the safety of drivers and thus reduce crashes resulting from adverse road weather conditions. This was achieved by disseminating, in real-time, the information collected by a vehicle to its surrounding vehicles using state-of-the-art wireless communications (Vorakitolan, et al. 2011) between vehicles. The information was also communicated to road side infrastructure to increase driver safety; for example, the duration of the traffic signals at a junction can be changed dynamically in response to current road weather conditions transmitted by vehicles in the surrounding area.

Vehicles are connected to each other using the Dedicated Short Range Communications (DSRC) protocol (Morgan 2010). The DSRC protocol is standardized in IEEE 1609 and uses the federally allocated 5.9 GHz frequency spectrum. DSRC is similar to WiFi with the added advantage of being fast, secure and reliable, and is specially designed and standardized for vehicular communications. NHTSA will eventually decide whether to bring in regulations requiring all car manufacturers to install DSRC in future cars in the US.

The application of wireless-based communication between vehicles on roads, including cars, trucks and busses, for road safety research is being coordinated in the United Sates by the Research and Innovative Technology Administration (RITA) of the Department of Transportation through their Connected Vehicles program (DOT 2014). RITA and National Highway Traffic Safety Administration (NHTSA) are involved in a joint research initiative, called Safety Pilot (NHTSA 2014), to investigate the connected vehicle technology for real-world application. Connected vehicle technologies has the ability to equip drivers with the tools needed to anticipate potential crashes and significantly reduce the number of lives lost each year.

1.2. Problem Statement

According to the National Highway Traffic Safety Administration (NHTSA), 5.4 million crashes in 2010 resulted in 2.24 million injuries and 32,885 deaths. During the ten year period (2002-2012), weather accounted for 23% of the vehicles crashes, 20% of crash injuries and 17% of the crash fatalities. Road weather variables include air temperature, humidity, wind speed, precipitation, fog, pavement temperature, pavement conditions, and water level. Among the weather-related crashes, 74% are on wet pavements, 46% during rainfall, 17% during snow or sleet, 12% on icy pavement, and 14% of crashes take place on snowy or slushy pavement. Data from Oklahoma crash reports during the period Jan 1998 – Oct 2009 shows that more than 52% of total crashes and 52.5% of the fatal crashes occur during inclement weather.

Quoting data from Federal Highway Administration, the per-person cost of traffic fatalities is $3.2M and $68,170 for injuries, with the cost of traffic crashes to be $166.7B per year (RMIIA 2008). Another report from auto club AAA states that the toll for Americans in a motor vehicle crash amounts to an average of $6M per fatal accident. The figure is based on property damage, lost earnings, loss of household activities, medical costs, emergency services, travel delays, vocational rehabilitation, lost time at work, administrative costs, legal costs, and pain and lost quality of life.

The impact of adverse weather conditions on loss of life and property is significant. Previous efforts on increasing road safety has primarily focused on better weather prediction and better pavements. U.S. Department of Transportation (DOT), in particular, has focused on helping people survive crashes rather than help drivers avoid crashes (RITA 2014). There hasn't been much effort in increasing safety by alerting drivers, with a view to avoiding crashes, of dangerous weather condition in real-time, based on exchange of in-situ measurements carried out by vehicles. In contrast to sharing weather road condition information collected by a vehicle, previous efforts have concentrated on off-line coarse level predictions and information on the general area of the vehicle location.

1.3. Previous Effort

For over a decade, the OU Intelligent Transportation Systems Laboratory has endeavored to apply new technology to deliver low-cost solutions to a variety of transportation related problems. By working closely with federal, state and local transportation stakeholders, the OU ITS Lab has delivered practice-ready solutions that have made an enormous impact on surface transportation in the state. The included letter of support from the Oklahoma Department of Transportation is evidence of our continued close collaborative relationship.

Employing a broad set of intelligent transportation system, computer engineering, software, and information technology capabilities to the transportation domain have resulted in numerous practical OU ITS Solutions. Deployed solutions include:

- Statewide Intelligent Transportation System and Virtual Traffic Management Center (vTMC) – Design, testing and participation in the deployment of the Oklahoma Intelligent Transportation System, a private, dual-ring fiber optic network that links hundreds of devices including cameras, dynamic message signs, and remote traffic microwave sensors (RTMS). The OU ITS-developed vTMC software consoles have been deployed to almost 100 transportation agents across the state. Robust peer-based communication protocols designed by the OU ITS Lab support the dissemination of a growing set of data from third party sources including weather information and third-party-collected traffic speed/volume data even in the presence of network disconnectedness. As part of the Oklahoma ITS, the OU ITS Lab hosts the Oklahoma's Advanced Traveler Information System (http://www.oktraffic.org), the public portal for monitoring traffic conditions. ATIS is used daily by thousands of travelers and several television and radio stations across the state. It provides users with live data and video of traffic conditions for key metropolitan roadways.

- Roadway Weather Information System (RWIS) (Vorakitolan, et al. 2011) and Inclement Weather vehicle AVL – The OU ITS Lab has provided support for the deployment of several RWIS stations and have implemented both fiber-optic and wireless communication to bring RWIS data into the statewide ITS system. In order to improve the effectiveness of the response to inclement weather, the lab implemented an automatic vehicle location system to wirelessly track the location of salt trucks and other vehicles used in response to inclement weather.

- Police Automated Records Information Systems (PARIS) – a mobile traffic records and electronic citation system deployed to Oklahoma Highway Patrol and Oklahoma City Police Department. Exploits GPS to enhance location information in traffic records and utilizes wireless communication for submission of traffic record from field. Chosen by Oklahoma Traffic Records Council for deployment to statewide municipal agencies as part of the state's Traffic Records Plan.

- Statewide Analysis for Engineering & Technology (SAFE-T) – traffic data analysis system used by Oklahoma Department of Transportation, Federal Highway Administration, US Indian Health Service, Oklahoma Highway Safety Office, Oklahoma Highway Patrol Troops, municipal police agencies, city, town and county governments and municipal government associations to analyze and map collision data. SAFE-T has been used by numerous agencies to demonstrate the efficacy of transportation improvement projects and to provide statistical support for grant applications and bond initiatives.

- Oklahoma Drivers License Testing System – Networked driver testing software system used by every Driver's License Testing Facility across the state.

1.4. Objectives

The objective of the project was to reduce vehicle crashes, fatalities and injuries due to adverse weather conditions, by alerting drivers in real-time of potentially hazardous road conditions in the immediate region, based on information from neighboring vehicles. The dissemination of vehicle and road condition information was carried out by an Ad Hoc network (Levin, Efrat and Segal 2014) connecting the vehicles in the neighborhood. The data measured by vehicle sensors were transmitted to road side units for transmission to servers for road condition prediction and control of road side infrastructure, such as traffic lights, work zones, dynamic message signs, pedestrian signals, and curve speed warning. The connected vehicles concept that is promoted by RITA is primarily to increase driver safety by enabling collision prevention applications during "normal driving" such as forward collision warning, blind spot warning, lane change warning, emergency brake lights, etc.; this proposal aimed at reducing crashes during "severe weather driving" by alerting drivers of potentially hazardous road conditions during inclement weather conditions.

The following tasks that were proposed were successfully completed by the research team:

- Survey DSRC-based equipment, both on-board equipment (OBE) and road-side equipment (RSE), available in the market with respect to standards-compliance, available interfaces, interoperability with other vendor equipment, performance, etc. and place equipment order.
- Install the OBE equipment in cars and make them communicate among themselves and the RSE.
- Install OBE equipment in ODOT vehicles used in inclement weather response.
- Install the RSE in the trailer.
- The equipment will be tested for interoperability with other vendor's equipment, especially those mounted on the trailers owned by OU ITS lab. They will also be tested for adherence to

the manufacturer's specifications; any issue arising will be communicated with vendors to come up with a solution.

- Program all the equipment for allowing data transfer between vehicles
- Build the vehicular ad hoc network based on Geographical routing to enable routing of data between vehicles and the RSE

2. Accomplishment by Tasks

In this section, we describe in detail the accomplishments of the various tasks outlined in Section 1.4.

2.1. Task 1: Evaluation and purchase of equipment

Work Done: Selecting OBU and RSU Equipment

The project required building a DSRC-based vehicular networking testbed for testing the concepts to be used in alerting drivers. In order to select DSRC equipment for our testbed, we made a short list of vendors which provide the DSRC devices and were recommended by the U.S. Department of Transportation (USDOT). Then, we compared them by considering the primary requirements of the project.

Project Requirements: In this project, we made connection among vehicles to exchange data called vehicle-to-vehicle communication (V2V) is based on DSRC protocol. So, the equipment had to support DSRC and the 802.11p standard which are mandatory for vehicle-to-X (V2X) communications.

We read information about vehicle operating conditions and surrounding weather conditions using sensors in the vehicle. The information is accessible via CAN bus which is connected to On-Board Diagnostic II (OBD-II) interface. An OBD-reader was used to read data which support Bluetooth. Extracted data from CAN were transmitted to an On-Board Unit (OBU). If DSRC device support Bluetooth, it will be helpful in making connection between OBU and OBD reader. Also, if we develop an application on a smart phone, OBU and the smart phone can communicate with each other via Bluetooth.

Furthermore, we developed a program to receive, analyze and transmit data. As this is a research project and our aim was to learn via experimenting, applications were developed by ourselves and it was vital for us to receive Software Development Kit (SDK) in addition to the equipment.

In all safety applications, it is required to know where exactly an event happens. It will help to provide drivers with the exact location of an incident thus they can assess their distance from that location and make a wise decision. So, it is obvious that we need Global Positioning System (GPS) information and equipment had to include GPS.

We used an OBD-reader from OpenXC to read data from CAN. The requirements that were considered to procure the equipment as shown in Table 1.

Table 1 Project Requirements in selecting DSRC Equipment

Requirement	Mandatory /Optional
DSRC	M
802.11 P	M
Bluetooth interface	M
Software Development Kit (SDK)	M
GPS	M
CAN interface	O

Comparison among DSRC equipment: As already explained to procure equipment for testbed, we compared the existing DSRC equipment recommended by USDOT to understand which one covered all mandatory requirements in this project. The result is shown in a Table 2.

Table 2 Comparison among DSRC-based OBU equipment.

Vendor	ARADA	ARADA	KAPSCH	COHDA
OBU Model	Classic LocoMate	Mini2 LocoMate	TS3306	MK5
Protocols	**Protocols**	**Protocols**	**Protocols**	**Protocols**
DSRC	Y	Y	Y	Y
IEEE 802.11 P	Y	Y	Y	Y
IEEE 1609	2, 3, 4	2, 3, 4	2, 3, 4	Complete stack
SAE J2735	Y	Y	Y	Y
ETSI ES 202 663	N	N	Y	Y
Interface	**Interface**	**Interface**	**Interface**	**Interface**
Ethernet	Y	Y	Y	Y
Serial Console	Y	N	Not mentioned	Y
CAN	Optional	Optional	Optional	Y

Vendor	ARADA	ARADA	KAPSCH	COHDA
USB	Y	N	Y	Y
Micro-USB	N	Y	Y	N
Bluetooth	Y	Y	Y	Y
Other Specifications	**Other Specifications**	**Other Specifications**	**Other Specifications**	**Other Specifications**
Power	12 DC	12 DC	8-36 DC	12 DC
Platform	Linux/ Unix Compatible	Linux/ Unix Compatible	N	Linux
Flash memory	Y	Y	Not mentioned	Not mentioned
microSD	N	N	Y	Y
RAM(Mbit)	512	512	N	N
Frequency	5.700 to 5925	5.700 to 5925	5.9 GHz	5 GHz
Bandwidth	10 and 20 MHz	10 and 20 MHz	10 MHz	10 MHz
GPS	Y	Y	Y	Y
Security	Y	Y	On request	Y
Log	Y	Y	Not mentioned	Not mentioned
SDK	Y	Y	N	Optional

Purchase Decision: Although equipment of all three vendors were mostly similar, there were few important criteria which played a substantial role in selecting the vendor. The most important factor was SDK. Since we wanted to develop applications on devices by ourselves, availability and delivery of SDK was vital for us. One of the well-known suppliers of DSRC equipment is Savari. But at first step of negotiation, they cited that they will not provide SDK and for developing any application, we have to ask them and there will be a charge without any doubt. Therefore, Savari was not good choice for us and we removed it from our short list. Since Capsha reveals no information about SDK in its datasheet, our choices were limited to Arada and Cohda. Despite the fact that both vendors provided SDK, there was a considerable difference between the costs. SDK was included in the Arada package and there was no need to pay for it separately, while it is not included in Cohda OBU package and we were required to buy it separately. Regarding the cost, Cohda sold OBU for $2080 and SDK for $4300 while ARADA sold OBU with SDK package for $1500. Therefore, we selected ARADA and we purchased two Classic LocoMate OBUs and three RSUs.

Work Done: Selecting OpenXC and CrossChasm

OpenXC is a platform developed by Ford Motor Company with intended use by third party manufacturers and researchers to obtain data from a vehicle. JavaScript Object Notation (JSON) libraries are available on most modern platforms and OpenXC libraries are also available for Python and Android applications. OpenXC provides some data from the OBD-II standard as well as the proprietary CAN bus data defined by the vehicle manufacturer in an open and easily used standard format based on the JSON markup language. This data was used in our system for analysis of driving conditions.

The Crosschasm 5 (C5) device was chosen due to its compatibility with OpenXC, support for custom firmware, low cost, and ODB-II hardware connection that is available on all modern vehicles. The C5 firmware can be customized to allow additional CAN bus signaling within the vehicle that is not part of the standard set of data provided by the default implementation of OpenXC. This combination of hardware and software components was used to obtain weather based hazard information such as windshield wiper status, traction control status, headlights, brake status, etc. The obtained data were used to identify hazardous conditions and send status messages to other nearby vehicles.

2.2. Task 2: Testing of equipment for interfacing with other equipment

Work Done: Configuring Crosschasm

The OpenXC platform is available as open-source code as well as pre-built specialized binaries from Ford Motor Company. The OpenXC firmware core for the C5 device is built within a Linux-based build environment. We programmed the C5 with the latest available source-code (v7.0) built on a Linux laptop and pre-built binary. The C5 translates known messages on the CAN bus in a vehicle. The data was transmitted through the Bluetooth or USB to a receiver such as the OpenXC mobile application, OpenXC Enabler. The enabler has a basic Graphic User Interface which displays vehicle data obtained from the car in a clearly interpretable form. Figure 1 summarizes the architecture of OpenXC.

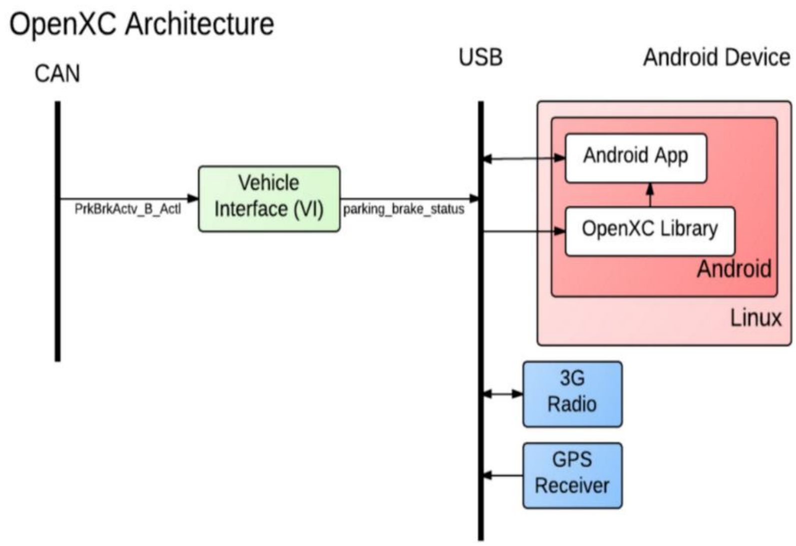


Figure 1 Open XC Architecture block diagram. Source: http://openxcplatform.com/vehicle-interface/concepts.html

Initially, the C5 is programmed and plugged into the car to capture the near real time vehicle data available on CAN bus. The data is transmitted to the mobile phone read by the mobile app using Bluetooth. We have programmed all the devices to ensure that all the equipment interoperate.

2.3. Task 3: Interface equipment with cars, trailer, salt trucks and computer system

The project involves interfacing of the OBD reader to car engine, interfacing the OBD reader to laptop and OBU, and interfacing between OBU to Road Side Unit (RSU). We decided to go with the OpenXC open source hardware and software platform for obtaining car engine parameters.

Work Done: Interfacing Crosschasm and Cars

OU Center for Intelligent Transportation System owns two vehicles: Ford F-250 and a Ford Expedition. The open-source Crosschasm C5 firmware v7 was used to capture the information shown in the figure below as it was most accurate and compatible with Ford F-250. Figure 2 below gives a snapshot of the data obtained using OpenXC Enabler from the Crosschasm C5.

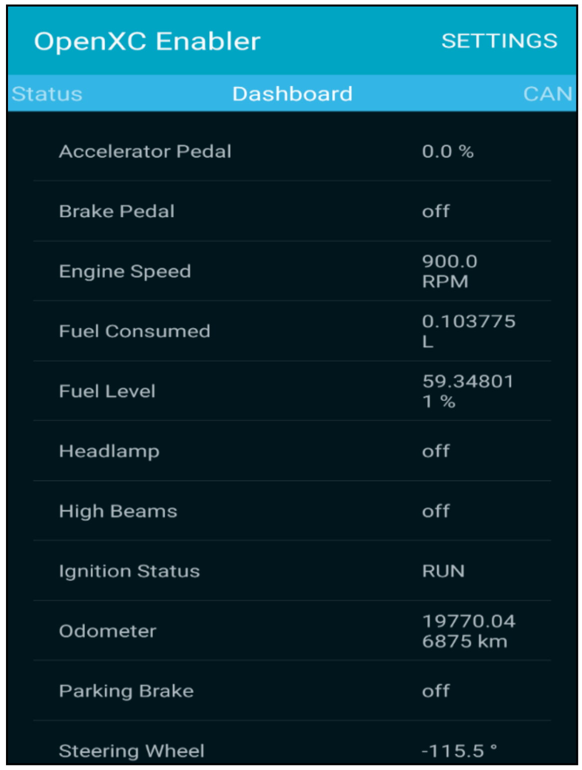


Figure 2 Screenshot of data received from 2013 Ford F-250 using the OpenXC Enabler app for Android.

Similarly, Figure 3 represents the data obtained from our 2013 Ford Expedition.

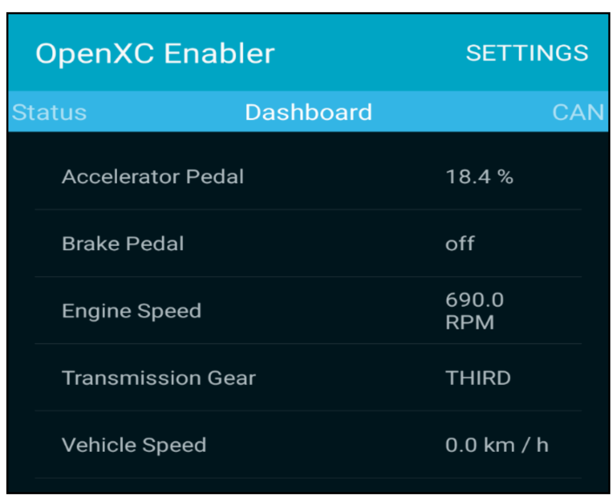


Figure 3 Screenshot of data received from 2013 Ford Expedition using the OpenXC Enabler app for Android.

We captured data from the two vehicles using the OpenXC enabler app and directly into a laptop connected to the C5 device as shown in Figure 4. We found that the two devices cannot capture the same information. Moreover, the parameters captured from the two vehicles are also different. Figure 5 also shows the parameters that can be captured from the two vehicles using the two different methods. As can be seen, we obtained more information from Ford F-250 than the Ford Expedition. This could be due to the OpenXC firmware being more compatible with the F-250 than the Ford Expedition.

Parameters	Ford F-250		Ford Expedition	
	App	Laptop	App	Laptop
Steering Wheel Angle	✓	✓	✓	✓
Torque at transmission	☒	☒	☒	☒
Engine speed	✓	✓	✓	✓
Vehicle Speed	✓	✓	✓	✓
Accelerator Pedal position	✓	✓	✓	✓
Brake Pedal position	✓	✓	✓	✓
Parking Brake Status	✓	✓	☒	☒
Transmission Gear Position	✓	✓	✓	✓
Odometer	✓	✓	☒	☒
Ignition Status	✓	✓	☒	☒
Fuel Level	✓	✓	☒	☒
Fuel Consumed since restart	✓	✓	☒	☒
Headlamp Status	✓	✓	☒	☒
High Beam Status	✓	✓	☒	
Windshield Wiper Status	✓	✓	☒	☒

INDEX	
✓	Obtained
☒	Not obtained

Figure 4 Parameters obtained using the OpenXC platform.

The following has been done with respect to interfacing CrossChasm C5 and ODB-II/CAN Interface Device:

- Firmware reprogramming for use of pre-built binaries from Ford Motor Corporations (FMC).
- In-vivo testing of the C5 in our Ford Expedition and F-250.
- Logging of data that is available in the defaults of the pre-built firmware.
- Development environment setup for customization of firmware.
- Contacted FMC for additional codes and values not currently available, such as traction and control status.

We have obtained a CrossChasm C5 ODB-II/CAN device for the direct interfacing of our V2V system with the on-board vehicular control system. Successful reprogramming of the device with pre-build firmware from FMC and subsequent in-vivo testing has allowed for the examination of the OpenXC data format as well as the supporting firmware for determining the applicability of the C5 in our system design. It was found that additional information from the vehicle control system would benefit the situational awareness in our system and we have been in contact with FMC to obtain access to proprietary codes needed for traction control and anything else we find will enhance the system.

A Linux based build environment has been set up in our lab for further customization and re-programming of the C5 in our subsequent development and integration work.

Discussion with Ford:

We had telephone discussion with Dr. Venkatesh Prasad at Ford Motors to obtain the proprietary firmware for our two vehicles, F-250 and Ford Expedition, to obtain all the necessary parameters that will be required to improve the accuracy of warning messages generated for vehicle to vehicle communication during unfavorable weather conditions. The parameters that we would like to obtain from Ford are: (i) traction control status, (i) anti-lock braking system status, and (iii) airbag status, (iv) Traction main current, (v) ambient temperature, (vi) brake pressure, (vii) brake pedal acceleration, (viii) automatic braking status, and (ix) windshield wiper park to park time. Dr. Prasad mentioned the requirement of legal documents between Ford and OU to obtain those parameters. It was felt that the additional information that would be obtained from the engine can in fact be simulated by other parameters, such as wiper on signal, that can be obtained and used within the limited scope of this project. The information from Ford would help for real implementation and testing on the road under a real inclement weather situation in the future.

2.4. Task 4: Programming of equipment to allow data exchange

We have successfully programmed the OBU and RSUs to communicate over DSRC, the RSU to communicate with the DMS over Sprint network and the OBU to receive data from OBD-II over Bluetooth.

Work Done: Interfacing between OBU and RSU

We have developed programs which run on the OBU and RSU and allows data exchange between the two devices using the DSRC protocol. The messages are transmitted using the Basic Safety Messages of the DSRC protocol. These messages are transmitted every 100 msec and contain various parameters such as

- Time (UTC time)
- Message Count (random starting time)
- Temporary ID (randomized every 5 min)
- Position Data Elements (Latitude, Longitude, Elevation)
- Positional Accuracy (Semi Major Axis, Semi Minor Axis, Semi Major Axis Orientation)
- Transmission State
- Speed
- Heading
- Steering Wheel Angle
- Acceleration (Longitudinal, Lateral, Vertical, Yaw Rate)
- Brake System Status (for each wheel [traction, abs, scs, brakeBoost, and auxBrakes])
- Vehicle Size (Width, Length)

Additionally, we developed an app that can read the real time emergency warning message coming from the vehicle and display it on a laptop screen so as to alert the driver about a potential hazard that she/he should be aware of. The app was designed in C# using .NET framework 4.5.

2.5. Task 5: Build ad hoc network and implement Geographical routing

The test bed will require necessary services to be in place to fulfill the test requirements. At this stage, we choose streets in front University of Oklahoma's Five Partners Place Building in Norman to do the test.

Work Done: Building the testbed network

Figure 5 shows how the physical components of the test bed that are connected together using wireless communication.

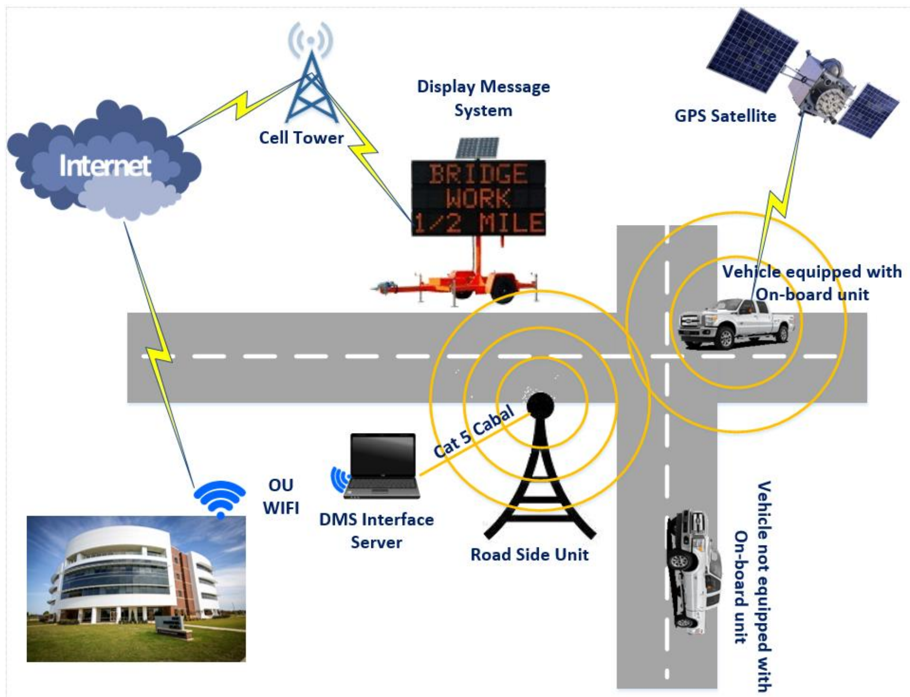


Figure 5 Physical components of the test bed.

A Ford F-250 was equipped with an ARADA OBU which has a GPS Antenna to provide the vehicle's location information. Also, an ARADA RSU was placed beside the street. The OBU and RSU communicate with each other through DSRC protocol whenever they are in the transmission range of each other. The RSU and DMS Interface Server (DMSIS) are connected with Cat5 cable. The DMSIS acts as an interface between RSU and Display Sign which gets data from RSU and transmits it to the sign. The DMSIS needs Internet connection to communicate with the display sign located beside the street. Therefore, the server was connected to OU's WIFI network which provided the Internet access.

2.6. Task 6: Develop algorithms to alert drivers of hazardous road conditions

The OBU in vehicles broadcast engine parameters and road conditions over DSRC. We developed algorithms to combine these parameters to warn drivers of hazardous road conditions.

We transmitted the car engine parameters from the OBU to the RSU, which in turn sent the message to the DMS via the DMS Interface server running on a laptop connected to the RSU. If the engine speed was over a certain threshold, the RSU sent the message “OVER” to the DMS via the Sprint Internet; the message was successfully displayed on the DMS.

Work Done: Software Modules

The main functions of each application are described. The flow chart of the applications based on each device is depicted in Figure 6.

<u>On-Board Unit</u>

As depicted in Figure 4, first the Bluetooth connection should be set up between OBD-II and OBU as explained in Section 2.3. Then, the “DSRC Transmission App” will run on the OBU. This program will read data from OBD-II via Bluetooth connection, receive location data from GPS and build the WSMP to transmit to other OBUs and RSUs via DSRC. The main tasks of this program are listed below:

DSRC Transmission App: (developed in C)

1. Make socket connection with OBD-II
2. Read data from OBD-II via BT after Bluetooth connection sets up between OBD-II and OBU.
3. Make socket connection with GPS. This works using the library gpsc_probe.h and you connect using function gps_connect with the IP address of local host as a parameter.
4. Read location data from GPS.
5. Append data read from OBD-II with GPS information.
6. Prepare DSRC requirements like invoking WAVE driver, registering as a provider, etc.
7. Build Wave Short Message Packet (WSMP).
8. Transmit WSMP to “DSRC Receiver App.” On RSU using DSRC.

We had one OBU and one RSU in our test bed, so we use OBU as a transmitter and RSU as a receiver in our scenario.

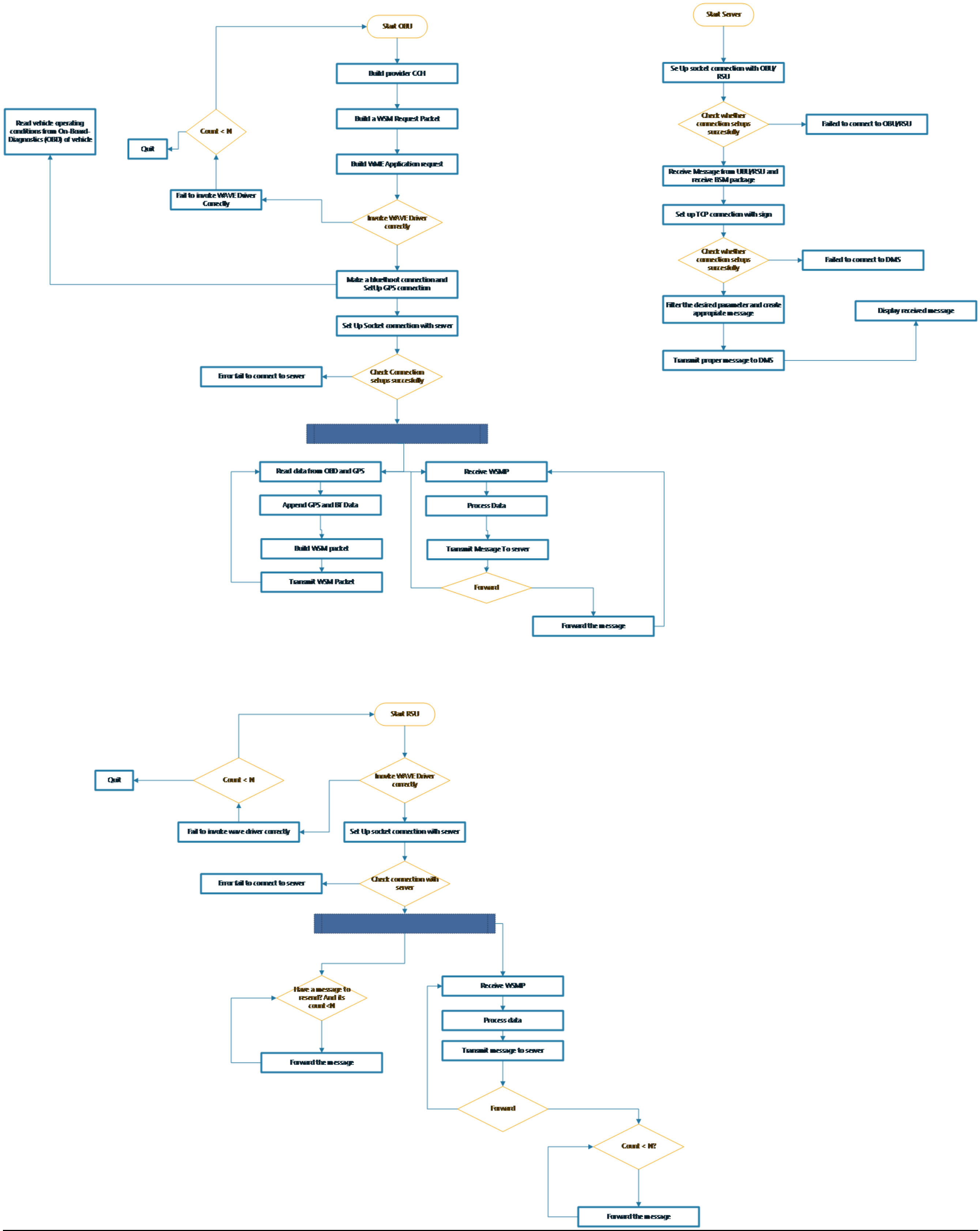


Figure 6 Flow chart of applications on each device.

Data Communication Format

Main functions in DSRC Transmission App:

1. BUILDPSTENTRY () // Filling Provider Service Table Entry
2. BUILDWSMREQUESTPACKET() // Building a WSM Request Packet
3. BUILDWMEAPPLICATIONREQUEST() // Building a WME Application Request
4. INVOKEWAVEDRIVER() // open a connection to a wave device either on the local machine or on a remote machine
5. SetOBDConnection(), GetOBDData() //Functions that open the Bluetooth socket to get data from OBD and get the data from it
6. SetGPSConnection(), GetCPSConnection() //Functions that open socket connection for the GPS and get data from it
7. REGISTERPROVIDER () //Register OBU as a provider
8. BuildWSMRequestPacket() //This function is used to build the wsm packet that is going to be transmited.
9. TXWSMPPKTS() //Function to send the package after build the packt
10. PROCESS_BT_MESSAGE_WITH_GPS() // Append GPS data with parameters and make WSMP

Roadside unit

One applicaton is running on RSU called "DSRC Receiver App.", Which receives data from OBU and transmit it to DMS Interface App on server.

- ***DSRC Receiver App:*** (developed by C)

1. Receive WSMP
2. Prepare DSRC requirements like invoking WAVE driver, registering as a user, channel allocation, etc.
3. Make TCP connection with "DMS Interface App." on a laptop depicted in Figure 4.
4. Transmit messages to "DMS interface App." via the socket connection.

Main functions in DSRC Receiver App:

1. INVOKEWAVEDEVICE ()
2. REGISTERUSER() //Register RSU as a user
3. RXWSMMESSAGE () // Function to receive the Data from TX
4. RXWSMIDENTITY () //Identify the type of received Wave Short

DMS Interface Server

The laptop plays relay node role. In order to communicate with Display sign we need to use a library which developed in .Net. Therefer, the DMS Interface App is developed in C# that can use this library and communicate with sign.

- ***DMS Interface App:*** (developed in C#)

1. Make TCP connection with RSU
2. Receive data from RSU
3. Display the messages on a GUI
4. Make a connection with controller on a sign
5. Filter parameters
6. Transmit message to the sign

Main Functions:

1. SETUPTCPSERVER // create socket connection with RSU
2. UPDATEGUI
3. SENDMESSAGETOSIGN // Use a library to send message to the sign
4. DISPLAYMESSAGETODMS // send proper message to sign based on specific parameter

The software diagram in Figure 7 shows the applications running on each device and their functionality.

Figure 7 Flowchart of implementation.

3. Test Results

3.1. OBU

On the OBU, we can see the status of transmitting data packets to RSU. The numbers of transmitted and dropped packets can be seen as marked by red in Figure 8.

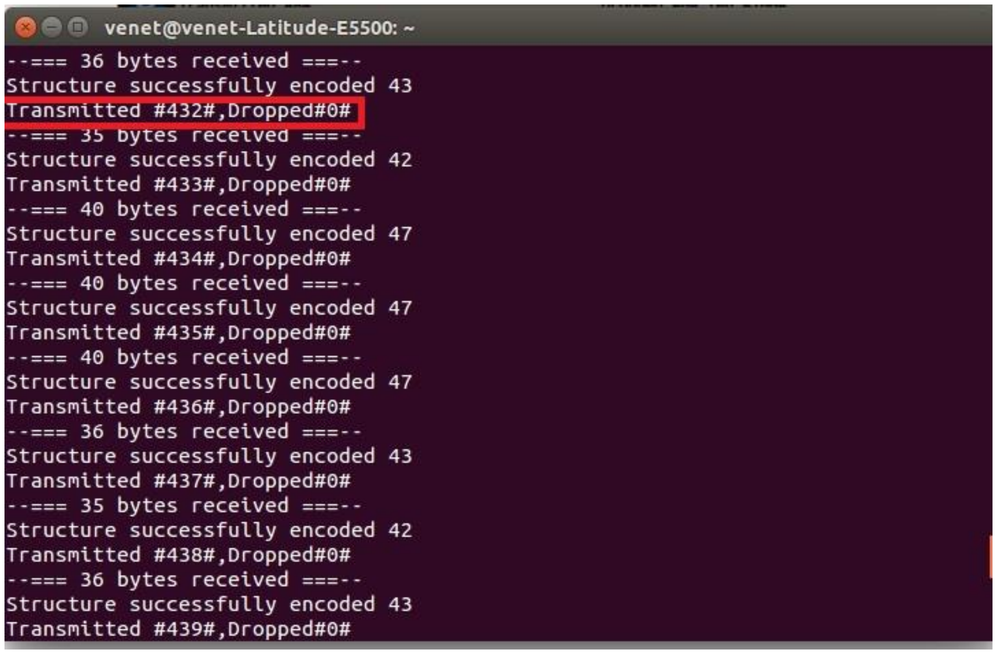
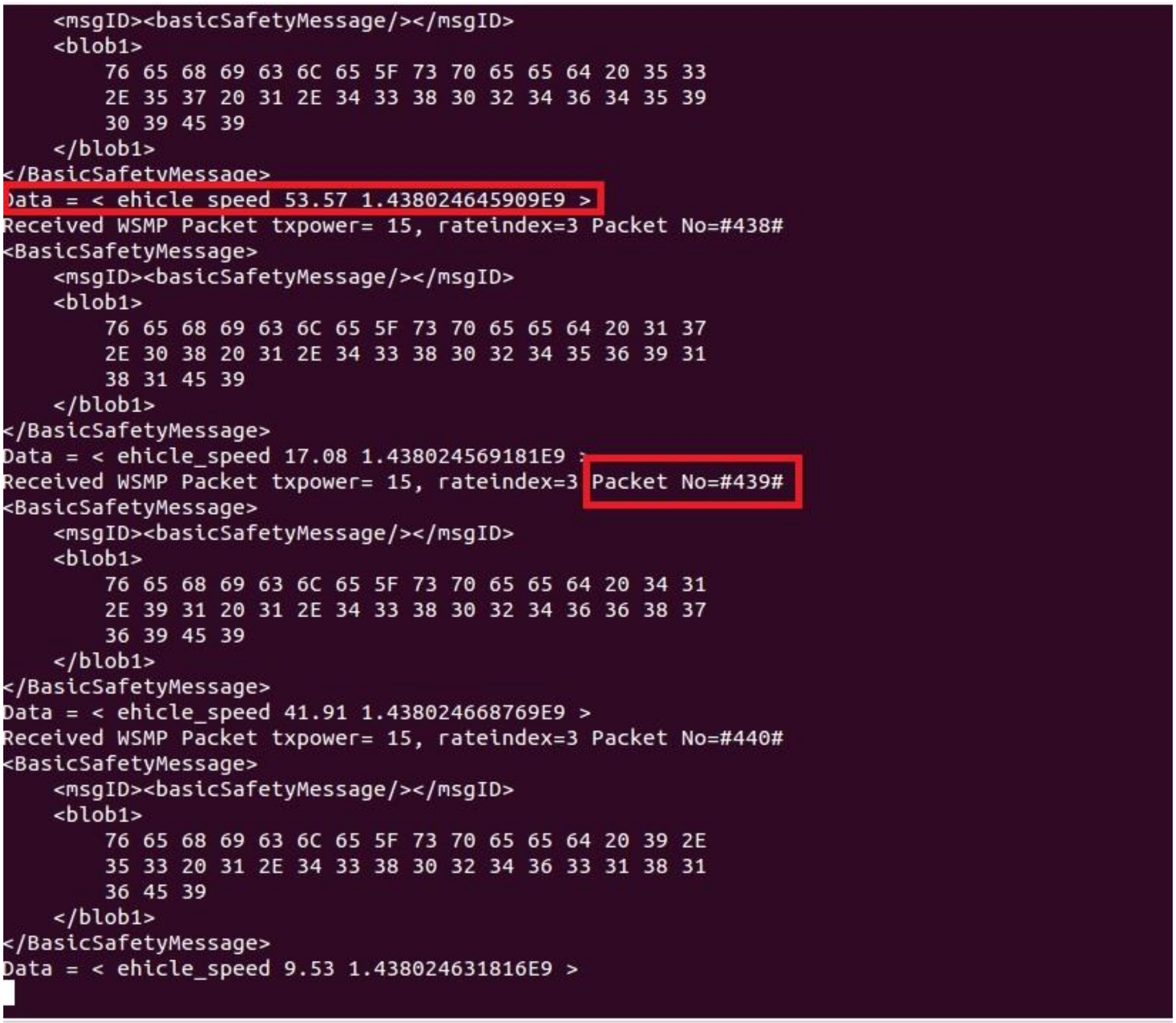


Figure 8 The status of packet transmission on OBU

3.2. RSU

In Figure 9, the result of running “DSRC Receiver program” on RSU is depicted. As marked by red, the number of received packet and filtered data can be seen. In this scenario, vehicle speed values are received.



```
    <msgID><basicSafetyMessage/></msgID>
    <blob1>
        76 65 68 69 63 6C 65 5F 73 70 65 65 64 20 35 33
        2E 35 37 20 31 2E 34 33 38 30 32 34 36 34 35 39
        30 39 45 39
    </blob1>
</BasicSafetyMessage>
)ata = < ehicle speed 53.57 1.438024645909E9 >
Received WSMP Packet txpower= 15, rateindex=3 Packet No=#438#
<BasicSafetyMessage>
    <msgID><basicSafetyMessage/></msgID>
    <blob1>
        76 65 68 69 63 6C 65 5F 73 70 65 65 64 20 31 37
        2E 30 38 20 31 2E 34 33 38 30 32 34 35 36 39 31
        38 31 45 39
    </blob1>
</BasicSafetyMessage>
Data = < ehicle_speed 17.08 1.438024569181E9 >
Received WSMP Packet txpower= 15, rateindex=3 Packet No=#439#
<BasicSafetyMessage>
    <msgID><basicSafetyMessage/></msgID>
    <blob1>
        76 65 68 69 63 6C 65 5F 73 70 65 65 64 20 34 31
        2E 39 31 20 31 2E 34 33 38 30 32 34 36 36 38 37
        36 39 45 39
    </blob1>
</BasicSafetyMessage>
Data = < ehicle_speed 41.91 1.438024668769E9 >
Received WSMP Packet txpower= 15, rateindex=3 Packet No=#440#
<BasicSafetyMessage>
    <msgID><basicSafetyMessage/></msgID>
    <blob1>
        76 65 68 69 63 6C 65 5F 73 70 65 65 64 20 39 2E
        35 33 20 31 2E 34 33 38 30 32 34 36 33 31 38 31
        36 45 39
    </blob1>
</BasicSafetyMessage>
Data = < ehicle_speed 9.53 1.438024631816E9 >
```

Figure 9 Received data on RSU

3.3. DMS Interface Server

As already explained, DMS Interface plays relay role to transmit message to the sign and also shows the parameters which were received. The sample of what we received from vehicle operational parameters is shown in
Figure 10. This helps us to compare received values for various parameters with real values in the vehicle. Moreover, we can check whether the received data is real-time or we receive it with delay. The result of this comparison is shown in Table 3.

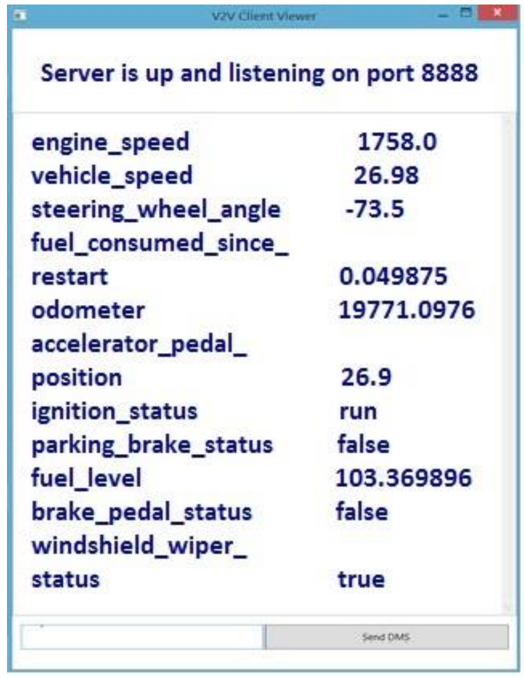


Figure 10 Vehicle operational status are shown on DMS Interface App

Table 3 Comparison of values of read parameters with real values.

	DMS App	DMS App	OpenXC App.	OpenXC App.
Parameter	Correctness	Real-time / delay	Correctness	Real-time / delay
Engine Speed	yes	Real-time	Yes	Real-time
Vehicle Speed	Not always	Delay	Yes	Real-time
Steering Wheel Angle	Yes	Real-time	Yes	Real-time
Odometer	Yes	Real-time	Yes	Real-time
Head lamp status	Not always	Delay	Yes	Real-time
High beam status	Not always	Delay	Yes	Real-time
Brake pedal status	Not always	Delay	Yes	Real-time
Windshield wiper status	Not always	Delay	Somehow	Delay
Parking brake status	Not always	Delay	Yes	Real-time

3.4. Display Sign

At the end, a proper message should be displayed on the DMS. In our scenario, we sent a message based on wiper status to the sign that is when the wiper was on, the related parameters changed from false to true. In the DMS Interface App. The status of this parameter will be checked and whenever it changes to true, a message will be displayed on the sign as depicted in
Figure 11.

Figure 11. Display the message on DMS.

4. Conclusions

All the tasks of this project has been successfully completed. The PIs and graduate students learnt a lot from this new and emerging area of research in the area of Transportation. Exposure to the new protocols and new hardware for vehicular communications will help the PIs to incorporate new material in the classroom and will improve the chances of securing future funding.

It was difficult to hire graduate students with the right expertise. It would be good to have a good pool of students at OU with the right expertise from which the PIs could hire research assistants for future projects.

The following agencies fund research in the area of vehicular communications. We should seek future funding from the agencies.

- RITA of U.S. DOT has a very large program called Connected Vehicles to improve the safety of drivers using vehicular communications for exchange of information between drivers. RITA and NHTSA are involved in a joint research initiative, called Safety Pilot (NHTSA 2014) to investigate the connected vehicle technology for real-world application. These objectives of this proposal is very much aligned to these very large federally funded programs; Preliminary results from this project will thus position OU at a very competitive edge to obtain funding from these agencies.
- Eight major automotive manufacturers are working collaboratively with U.S. DOT and University of Michigan to test the effectiveness of vehicular communications. The manufacturers include Ford, General Motors, Honda R&D Americas, Hyundai-Kia America Technical Center, Mercedes-Benz Research and Development North America, Nissan Technical Center North America, and Toyota Motor. Results and expertise gained from this project will help OU to compete for industry funding from automotive companies such as those mentioned above, in addition to possible Federal funding through RITA an NHSTA.

- Several NSF programs welcome proposals in the area of vehicular communications. Example of such programs are Cyber-Physical Systems (NSF 14-542) and the Computer and Network Systems (CNS): Core Programs (NSF 13-581).
- The OU ITS Lab has a long standing working relationship with ODOT. Results obtained from this project should increase our chances of obtaining funding in this new area of weather-related vehicular safety.

Bibliography

DOT. 2014. *Connected Vehicles.* Mar. http://www.its.dot.gov/safety_pilot/index.htm.

FHWA. 2014. *How do weather events impact roads.* Mar. http://www.ops.fhwa.dot.gov/weather/q1_roadimpact.htm.

Khabbaz, M. J., C. M. Assi, and W. F. Fawaz. 2012. "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges." *IEEE Communications Surveys and Tutorials* 14 (2): 607-640.

Levin, L., A. Efrat, and M. Segal. 2014. "Collecting data in ad-hoc networks with reduced uncertainty." *Ad Hoc Networks* 17: 71-81.

Morgan, Y.L. 2010. "Notes on DSRC & WAVE Standards Suite: Its Architecture, Design, and Characteristics." *IEEE Communications Surveys & Tutorials* 12 (4): 504 - 518.

NHTSA. 2014. *Connected Vehicles.* Mar. http://www.safercar.gov/ConnectedVehicles/pages/v2v.html.

RITA. 2014. *RITA - Intelligent Transporation Systems.* http://www.its.dot.gov/connected_vehicle/connected_vehicle_research.htm.

—. 2013. *Road Weather Connected Vehicle Applications.* http://www.its.dot.gov/connected_vehicle/road_weather.htm.

RMIIA. 2008. *Cost of auto crashes and statistics.* www.rmiia.org/auto/traffic_safety/Cost_of_crashes.asp.

Vorakitolan, E., R. Barnes, J. Havlicek, and M. Atiquzzaman. 2011. "Exploiting Trunked Radio to Support ITS Network Expansion and Redundancy." *PIMRC'11: 22nd IEEE Personal Indoor Mobile Radio Communications.* Toronto, Canada. 761-766.