14 June 2020

Dr. Craig Scratchely,

School of Engineering Science

Simon Fraser University

8888 University Dr.

Burnaby, B.C., V5A 1S6

RE: ENSC 405W/440 Requirements Specification for CANnect

Dear Dr. Scratchley,

This requirements specification document was prepared by CANtech (Group 8) for ENSC 405W/440 as a part of our engineering design process. It outlines and defines the necessary requirements for our product, CANnect. CANnect is an open source hardware and software package, providing the basis for communication between the Controller Area Network of your car and your smartphone.

From automotive industry professionals, to the everyday car enthusiast, CANnect can be utilized to read and interpret various raw data from a vehicle through a standardized protocol and transmit them to a smartphone app via a Bluetooth connection. CANnect applications include car diagnostic capabilities, such as relaying vital information related to vehicle systems, transmission, battery, engine control, etc. Allowing our product to be open sourced promotes further customization of our design to cater to the needs of our customers.

This report examines the essential hardware and software requirements supporting each phase of our design, including the proof of concept prototype, engineering prototype and applicable production phase. Furthermore, it analyzes the sustainability and safety aspects, and engineering standards incorporated in our design.

Our team at CANtech would like to thank you for taking the time to review our requirement specification document. Any questions related to our report can be directed to ranjoatc@sfu.ca.

Sincerely,

Ranjoat Chana

Chief Communication Officer

CANtech.

Requirements Specification: CANnect

Partners:
Enes Yazici
Choong Jin Ng
Nicholas Lau
Ranjoat Chana
Takehiro Tanaka
Win Aung

Contact:
Ranjoat Chana
ranjoatc@sfu.ca

Submitted To:
Dr. Craig Scratchely, ENSC 405W
Dr. Andrew Rawicz, ENSC 440
School of Engineering Science
Simon Fraser University

Date Issued:
June 14, 2020

Version:
1.0.0

Abstract

The global aftermarket is a rapidly growing industry that is expected to reach about $722 billion by the year 2020 [1]. Today's consumers are taking an interest in the maintenance of vehicles to ensure they maximize its lifetime. Our team at CANtech is introducing an open source hardware and software package called CANnect that allows consumers to perform car diagnostics and read real-time telemetrics of their vehicle. CANnect is a combination of a hardware module that can be connected to the CANbus system of the vehicle, and a software application that receives this vehicle data through a Bluetooth connection. Our product is open source which would allow consumers to customize our base features to meet their needs and introduce new solutions to our platform.

The development of CANnect is split into three subsystems, namely the hardware component, software system and software application. The requirements for each subsystem are categorized according to the various phases of our design and outlined with supporting research.

The main categorization or our requirements:

- General Requirements: General specifications relating to the entire system that cannot be broken down into hardware or software
- Hardware Requirements: Specifications of hardware on the microcontroller and mobile application
- Software Application Requirements: Specifications of software on the microcontroller mobile application

The test strategy for the proof of concept prototype is included as a means to ensure that our product meets a level of quality as outlined in the applicable engineering standards and protocols implemented in our product design. Safety and sustainability measures are put in place to ensure that our product achieves a level of quality and security, that our consumers can be confident in investing.

Changelog

Date	Version	Notes
2020 June 14	1.0.0	Published

Table 0-1: Changelog

List of Tables

List of Figures

Table of Contents

1 Introduction

In many modern cars, the different electrical subsystems and engines talk to each other using CAN Bus. CAN Bus was deliberately designed to facilitate high-speed communication between the devices while minimizing collisions and allowing crucial systems to communicate without being interrupted. CAN Bus was made mandatory in the United States as part of OBD-II in 2008 and has birthed a growth of adoption across manufacturers beyond essential diagnostics.

People wanting to troubleshoot their car or read information about their cars require the use of a OBD-II reader. However, many commercial offerings of OBD-II readers are expensive, and its features are not standardized across the market. Many enthusiasts, engineers and mechanics may have compared different OBD-II readers in order to achieve their goals, ranging from tuning their car to diagnosing engine codes, to engineering a solution that utilises the CAN Bus system. Open-source alternatives are cheaper and more customizable, but many require electronic and software knowledge, assembly of hardware, loading the appropriate firmware and often do not include easy or intuitive user interfaces for the average user.

The inspiration of our product is an open-source package of a CAN Bus Reader that aims to match most of the features and user interface of commercial offerings while allowing easy customization and affordability of open-source alternatives.

The CAN Bus Reader would be attached to the OBD-II port of a car and read the raw data frames from the system. These data frames would be sent to a smartphone app wirelessly where the data frames would be interpreted and displayed to the user. The data frames can range from diagnostic engine codes to engine telemetrics.

1.1 Background

1.1.1 OBDI-II

In 1982, the California Air Resources Board (ARB) developed regulations that required all vehicles sold in California, starting in 1988, to have an onboard diagnostic system (OBD-I) to detect emission standards. It was a good step but was too simple to detect certain kinds of problems and lacked standardization between cars and manufacturers. Thus, the ARB began to develop a better standard for the next generation OBD system for a more complex system. In 1996, OBD-II was made mandatory for cars sold in the United States and became fully compliant on all cars in 1999. The OBD-II connector, protocols and messaging format were also standardized, enabling people to buy one scan tool for all types of cars.

OBD-II provides a standardized series of diagnostic trouble codes (DTCs) and real time data to allow people to identify problems within the vehicle. OBD-II information also allows vehicle telematics, allowing users and businesses to compare data provided by a vehicle to external factors such as fleet emissions, driving behaviour, idle time, and fuel economy.

1.1.2 CAN Bus

CAN Bus (Controller Area Network) is a reliable communication bus that allows components within a system to send data without a centralized host to manage transmissions. In the context of a car, this means the subsystems in a car can share the same communication lines to reduce complexity while still remaining robust. This is achieved by subsystems performing arbitration or negotiating their intent to send data. If another subsystem of higher priority must transmit, it will arbitrate to indicate its priority over another. After arbitration is complete, data is only transferred from one endpoint to another.

Prior to 2008, multiple diagnostic communications standards and even CAN bus standards from both SAE and ISO existed, leading to fragmentation and requiring OBD-II readers to support multiple standards from different car manufacturers. Since 2008, CANBUS has been made standard for vehicles in the United States to support OBD-II, particularly ISO 15765-4 which was made a requirement for standardizing CAN communication with external equipment using interfaces such as OBD-II [2].

CAN Bus is used to provide the communication framework, while OBD2 uses CAN messages to provide diagnostic data through its protocol and physical interface.

Today, manufacturers are utilising the CAN Bus network beyond its intended purpose for OBD-II. Designers and engineers use the CAN Bus network to interface to non-essential systems such as the radio, window motors, interior lights and seats to provide amenities to consumers. Beyond amenities, systems utilise the CAN Bus network to provide features such as adaptive lane assist, cruise control, automatic braking system and real-time suspension adjustment. Software takes advantage of the existing network to determine fuel economy, engine oil health and the tyre pressure of each individual wheel. Finally, vehicle data can be collected in higher fidelity to provide more in-depth tuning of car systems and to replicate car characteristics accurately.

1.1.3 Competitors and Target Audience

The market of ODB-II readers can be categorized into two sections:

- Pricy commercial offerings meant for mechanics and car enthusiasts, with subscription and unlocking fees for more features. Generally has good customer support but lacks customizability [3].
- Cheap open-source electronics meant for hobbyists and electronic enthusiasts, that must be assembled and programmed. Very customizable but very dependent on the skill and knowledge of the user [4].

CANtech aims to offer a middle-of-the-road offering to bridge the two sections to tap a potential market segment. The open-source nature of CANnect aims to make it potentially customizable and fits according to a person’s niche, while CANtech manufactures the plug-and-play product and provides technical and customer support.

Our target audience can be described with at least two of the following statements:

- They self-identify to have an interest in cars and related technology
- Their knowledge of cars and related technology, especially in maintenance and performance, is higher than the average person.
- Their interest and knowledge in electronics and/or software is higher than the average person
- They enjoy working with and driving their cars

Through market research and interviews, most people that identify with at least the two statements above are intrigued by the prospect of CANnect. Potential customers especially take an interest in the distinctive nature of open-source, which allows them to potentially customize their user experience, and plug-and-play, removing the need to assemble or program their hardware. They also like to see the same product have the customer service and technical support, should troubles arise with and without modifications to the original design.

1.2 Scope

This requirement specifications document outlines the deliverables for each phase of the design process from the proof of concept prototype to the engineering prototype and finally the production phase. The document further details the functional requirements of CANnect's hardware and software components. Safety and sustainability considerations as well as the applicable Engineering Standards and acceptance testing plans are also included in the document.

1.3 Audience of this Document

This document is intended for use by CANtech and its partners. Reference to this document is being provided to Dr. Craig Scratchely, Dr. Andrew Rawicz and teaching assistants for review. It will serve as a resource for the team at CANtech through the various stages of our product design. The requirements, safety and sustainability considerations, engineering standards and supporting test plans outlined in this document will be used to ensure the quality and functionality of our product, CANnect, is achieved. Future revisions may be made to this document and will be noted in the changelog.

1.4 Requirement Classification

The requirement specifications are classified for various stages of the design process using the following convention:

[Section].[Subsection].[Requirement Number].[Development Stage]

Classification	**Classification Shorthand**	**Development Stage**	**Release**
Alpha	a	Proof of concept prototype	August 2020
Beta	b	Engineering prototype	December 2020
Prod	p	Production	April 2021

The alpha phase outlines requirements to be completed for the proof of concept prototype as a part of the 405W Demo. The beta phase outlines the engineering prototype requirements for the end of 440. The prod phase represents the requirements for the potential production and manufacturing of our design.

The words "shall" and "must" are binding provisions used in this document that express constraint and certain quality. They are used for functional requirements and non-functional requirements respectively. Non-binding provisions are indicated by the words "should" and "may". A declaration of purpose is indicated by the word "will". Only one provision or declaration of purpose for each requirement shall be used if present.

2 System Overview

CANnect is an open-source hardware and software platform that allows you to access data pertaining to the various systems in your vehicle.

The hardware components on the car-side (the reader) are made up of a microcontroller, Bluetooth adapter and other parts that facilitate the connection to the CANbus system in your vehicle. The raw data from the CANbus system and/or the diagnostic data from OBD2 is processed by said microcontroller in the reader to optimize for transmission, and then sends this to a smartphone device through a secured Bluetooth connection. Once sent over, it can be decoded and visualized on the software application. The overall system overview is displayed in Figure 2-1.

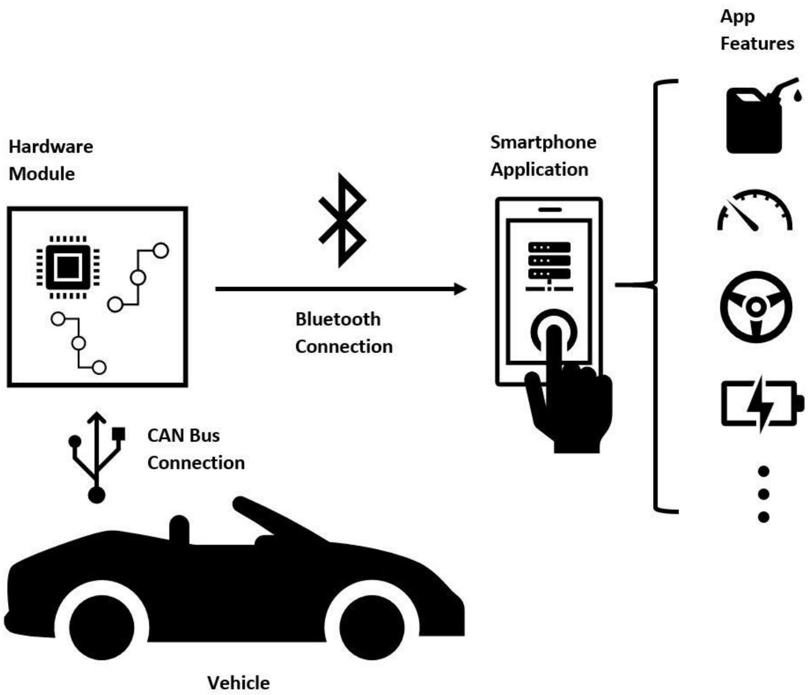

Figure 2-1: System Overview of CANnect

The software application converts and categorizes the data into a user-friendly format for the purposes of vehicle diagnostics and more. Our open-source platform allows for further customization of our base design and to our application features.

Figure 2-2: CANnect Reader Concept Design

Referencing the concept design in Figure 2-2, CANnect has a male type 16-bit OBD-II port connector to connect to car's communications busses. With this connector CANnect can fit the OBD-II diagnostics port in most cars, which are usually located around the numbered areas as shown in Figure 2-3

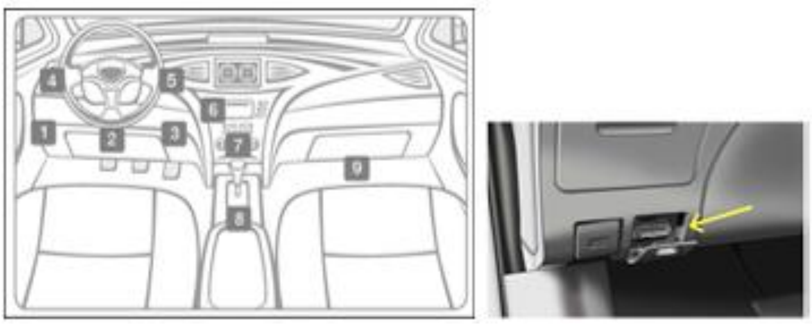

Figure 2-3: OBD-II Port Location Inside a Vehicle

For simplicity, the hardware module as outlined in Figure 2-2 will be referred to as 'reader' and the software application will be referred to as 'software application'.

1.1.1.a	The entire system shall consist of the following sub-systems: • CAN-Bus Reader • Smartphone
1.1.2.a	The entire system shall be open-source in accordance to the licenses detailed in
1.1.3.b	The smartphone shall have a maximum operating distance of 10 metres from the reader.
1.1.3.p	The entire system shall support the following manufacturers:

	• Toyota

Table 2-1: System Requirements

3 Hardware Requirements

The main hardware component of the product is a microcontroller with CANBUS and Bluetooth interfaces. Requirements will be met at different increments at the development stages specified.

Table 3-2 outlines the high-level hardware requirements of the reader. Note that these requirements can be expanded heavily during the development cycle.

2.1.1.b	The reader shall support a data speed of at least 1 Mb/s (megabits/s)
2.1.2.a	The reader shall have a visual indicator to indicate that it is powered.
2.1.3.a	The reader shall have Bluetooth capability.
2.1.4.b	The reader shall use the car's power supply.
2.1.5.a	The reader shall interface with the car's OBD-II port.

Table 3-1: CAN Bus Reader Hardware Requirements

Since smartphones vary across the years and among users, the main hardware requirement at time of writing is to support wireless capability (i.e Bluetooth).

2.2.1.a	The smartphone shall have Bluetooth capability.

Table 3-2: Smartphone Hardware Requirements

4 Software Requirements

The CANnect consist of 2 software components; the software on the microcontroller will extract the data from a data frame and then transmit the serial data from the microcontroller to a smartphone app through a Bluetooth module. Within the app, the data will be processed and displayed to the user, providing meaningful information to the user. With a user-friendly interface, the users will be able to easily interpret the data for their own needs.

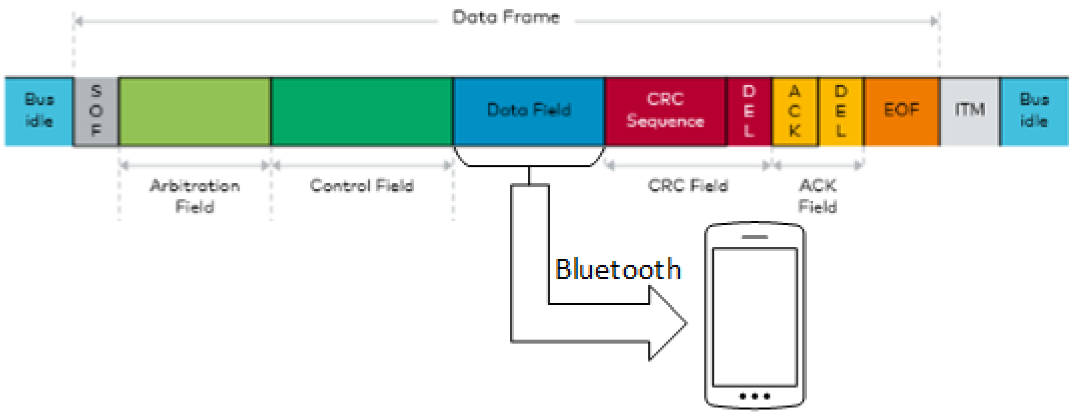

Figure 4-1: Data Transfer Overview from Data Frame of CAN Bus to Smartphone

Because the reader will most likely be a microcontroller, the software requirements for the reader will be small in size and will be mostly describing the functionality.

3.1.1.b	The reader shall read the data frames received from the car.
3.1.2.a	The reader shall not perform any data processing in the software.
3.1.3.b	The reader shall have offloaded the data frames read to the software application.
3.1.4.a	The reader shall offload the data frames only if the data frames are valid.

Table 4-1: CAN Bus Reader Software Requirements

The software application low-level requirements will grow larger during the development cycle as the software application is the main user interface most users will operate on. Therefore, the scope of these software requirements in Table 4-2 are high-level and are meant to convey an overview operation of the software application.

3.2.1.b	The software application size, excluding user generated data, shall be, at most, 150 MB.
3.2.2.a	The software application shall be available for Android.
3.2.3.b	The software application shall ensure and verify the data is received from the reader.
3.2.4b	The software application shall interpret the data received from the reader.
3.2.5.b	The software application shall connect to only one reader at a time.
3.2.6.b	The software application shall connect to the reader with Bluetooth.
3.2.7.b	The software application shall display the status of the connectivity to the reader.

3.2.8.a	The software application shall show a list of parameters for the user to choose and display.
3.2.9.a	The software application shall show the reading values corresponding to the parameter the user chooses to view. parameter
3.2.10.a	The software application shall present the values in a standardised, user-accessible format.
3.2.11.b	The software application shall display, but not limited to, the following parameters: • Engine RPM • Engine Speed • Engine Oil Temperature • VIN Number
3.2.12.a	The software application shall update its readings at least once per five seconds.

Table 4-2: Software Application Requirements

5 Safety Requirements

Getting the appropriate product certifications will ensure that our product is safe and a certain standard of quality has been met. It indicates to our consumers that our product meets acceptable performance and quality assurance tests have passed. CANnect interacts with the systems in user vehicles, so the requirements of our product will comply with vehicle safety measures as well. The requirements are split between ensuring hardware safety and ensuring safe, willful interaction with the software application. The latter is important to vehicle safety and is our purpose as outlined in the first entry of Table 5-1.

5.1.1.a	The software application will minimise interference and distraction with the user.
5.1.2.p	The reader should display appropriate warnings and product labelling for electronic devices.
5.1.3.b	The reader shall be fully enclosed in a protective casing
5.1.4.a	The reader shall not cause irreparable damage to the car
5.1.5.a	The reader shall not overheat, or pose as a fire hazard
5.1.6.b	The reader shall provide minimum user interference.

5.1.7.p	The reader shall be compliant with CSA standards
5.1.8.p	The software application shall not interfere with existing smartphone operations unless intended by the user.
5.1.9.a	The software application shall

Table 5-1: System Safety Requirements

6 Sustainability Requirements

CANtech aims to incorporate environmental sustainability into our designs as much as possible [5]. For CANnect specifically, we aim to make use of secondary materials such as recycled plastics in the outer-casing of our hardware component. Considerations for vehicle safety, electrical components and circuitry, and product sustainability are outlined below in Table 6-1. Further progress into development may produce more high-level requirements.

5.2.1.p	The reader protective casing should be formed from recycled plastics
5.2.2.b	The reader shall be composed of non-toxic materials and shall not pose health risks to the user (electromagnetic radiation)
5.2.3.p	The reader should incorporate sustainable packaging.

Table 6-1:System Sustainability Requirements

7 Engineering Standards

Engineering standards are set in place as a guideline for emerging technologies. They outline the important features of products and are used to ensure the compatibility, safety and quality of products and services [6] [7]. The engineering standards applicable to the hardware and software components of our product design are referenced below. While there will not be any specific requirements pertaining to adhering exactly to the engineering standards, the product will be designed to ensure compliance.

Standard	**Description**
ICES-003	Information Technology Equipment (Including Digital Apparatus) - Limits and Methods of Measurement [8]
CAN/CSA-C22.2 No. 0-19 (R2018)	General Requirements - Canadian Electrical Code, Part II [9]
CAN/ CSA-C22.2 No. 61508-1:17	Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements [10]

ISO 15765-4:2016	Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 4: Requirements for emissions-related systems [10]

Table 7-1: Hardware Engineering Standards

Standard	**Description**
CAN/CSA-ISO/IEC 9126-1:02 (R2007)	Software Engineering - Product Quality - Part1: Quality Model [14]
CAN/ CSA-C22.2 No. 61508-3:17	Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements [10]

Table 7-2: Software Engineering Standards

Standard	**Description**
IEC GUIDE 114: 2005	Environmentally conscious design - Integrating environmental aspects into design and development of electrotechnical products [11]
RoHS (Directive 2002/95/EC)	Restriction of Hazardous Substances - Environmental policy to restrict the use of specific hazardous materials found in electrical and electronic products [12]

Table 7-3: Environment Standards

8 Licenses

Our project will be following the following licensing standards:

- CERN Open Hardware License Version 2 - Permissive (referred to as CERN-OHL-P) [13]
- MIT License [14]

CERN-OHL-P allows people to take our work, relicense it and use it without any obligation to distribute the sources when they ship a product. Note that CERN-OHL-P, for our product, only applies to hardware and its associated documentation.

The MIT license allows people to take our software, make copies, distribute, and/or sell copies of it. Note that MIT license, for our product, only applies to software and its associated documentation.

The team will maintain an open-source repository of documentation, designs and software, and the repository will be opened to the public. The support of any particular version and build will be supported for at least 2 years upon release.

9 Acceptance Test

To ensure the above requirements are met, CANtech has devised a series of acceptance tests that fully cover all the requirements listed in this requirement specification document. These acceptance tests are designed to be non-technical and do not require prior skill nor knowledge beforehand, and can be performed by any user to demonstrate compliance to the requirements as outlined. Because each acceptance test is expected to cover different types of requirements, it is imperative that the tester follows each step one-by-one as each step pertains to one or more requirements.

The acceptance test plan for the alpha build is outlined in 12.2 below.

10 Conclusion

CANnect enables car owners to evaluate car diagnostic information using a plug-and-play product according to their needs. By leveraging the CAN-BUS system within vehicles as well as OBD2, data can be extracted and interpreted by the accompanying phone application in a user-friendly manner. The open-source nature of the product is intended to aid car owners and enthusiasts in insight of their vehicles with an end-to-end solution, able to adapt to the users' needs without the complexity of managing hardware.

The requirements outlined in this specification document details the two sub-systems, the CAN Bus reader and the software application, and the intended functionality. The acceptance test plan for the alpha build is outlined in Table 12-2: Acceptance Test Plan – Alpha and will cover the acceptance test for the proof of concept build. The alpha build is expected to be completed by August 2020 while the beta build is expected to be finalised by December 2020.

Beyond this document, future requirements will be updated and added on the website link: https://csil-git1.cs.surrey.sfu.ca/capstone-1194-group-8/ensc-405w/-/wikis/requirements.

11 References

[1] R. "A Look at Trends and Statistics in the Automotive Aftermarket Industry 2017 | V12," 27 May 2018. [Online]. Available: https://v12data.com/blog/a-look-at-trends-and-statistics-in-the-automotive-aftermarket-industry-2017/. [Accessed 11 June 2020].

[2] CSS Electronics, "OBD2 Explained - A Simple Intro (2020)," CSS Electronics, 2020. [Online]. Available: https://www.csselectronics.com/screen/page/simple-intro-obd2-explained. [Accessed 10 June 2020].

[3] OBD Solutions, "OBDLink® MX+," 2020. [Online]. Available: https://www.obdlink.com/mxp/. [Accessed 08 June 2020].

[4] "The OpenXC Platform," [Online]. Available: http://openxcplatform.com/. [Accessed 01 June 2020].

[5] G. Cook and E. Jardim, "Greenpeace Report: Guide to Greener Electronics 2017 - Greenpeace USA," October 2017. [Online]. Available: https://www.greenpeace.org/usa/reports/greener-electronics-2017/. [Accessed 13 June 2020].

[6] CSA Group, "Product Certification & Standards Development," [Online]. Available: https://www.csagroup.org. [Accessed 14 June 2020].

[7] "Why Product Certification Matters," 14 August 2019. [Online]. [Accessed 12 June 2020].

[8] Government of Canada, "ICES-003 — Information Technology Equipment (Including Digital Apparatus) — Limits and Methods of Measurement," 18 April 2019. [Online]. Available: https://www.ic.gc.ca/eic/site/smt-gst.nsf/eng/sf00020.html. [Accessed 12 June 2020].

[9] CSA Group, Canadian Electrical Code, part I, 24th ed., Toronto, Ontario: CSA Group, 2018.

[10] "ISO 15765-4:2016(en) - Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 4: Requirements for emissions-related systems," International Standards Organisation, 2016. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:15765:-4:ed-3:v1:en. [Accessed 14 June 2020].

[11] "IEC GUIDE 114:2005 - Environmentally conscious design - Integrating environmental aspects into design and development of electrotechnical products," Standards Council of Canada, 27 May 2005. [Online]. Available: https://www.scc.ca/en/standardsdb/iec/10461. [Accessed 13 June 2020].

[12] "RoHS Guide," 14 June 2020. [Online]. Available: https://www.rohsguide.com/. [Accessed 14 June 2020].

[13] J. Serrano, M. Ayass and A. Katz, "CERN OHL Version 2," CERN, 20 May 2020. [Online]. Available: https://www.ohwr.org/project/cernohl/wikis/Documents/CERN-OHL-version-2. [Accessed 11 June 2020].

[14] Open Source Initiative, "The MIT License," Open Source Initiative, [Online]. Available: https://opensource.org/licenses/MIT. [Accessed 11 June 2020].

12 Appendix

12.1 Glossary

Terms used in this document are further explained in this glossary, organised by alphabetical order.

Term	**Definition**
CERN	European Organization for Nuclear Research
Controlled Area Network (CAN bus)	In automotive applications it is referred to as the nervous system, that enables communication between the various ECUs in a vehicle.
Diagnostic Trouble Codes (DTC)	Used to communicate and help diagnose automotive problems in a vehicle
Electronic Control Unit (ECU)	Devices that control the various electrical systems in automobiles and interact via the CAN bus
International Standards Organization (ISO)	Create international standards based on expert research, as a guide for new products and ideas
On-board Diagnostics (OBD)	A computer system designed to track the state and performance of various components inside a vehicle.
Society of Automotive Engineers (SAE)	Develop standards for engineering professionals
VIN	Vehicle Identification Number

Table 12-1: Glossary

12.2 Acceptance Test Plan – Alpha

This acceptance test plan is meant to demonstrate compliance of the alpha build with the requirements as outlined in this document. Future acceptance test plans of other builds may vary in detail and specifics but will consist of the core concepts of this test plan.

Due to the ongoing development during the alpha acceptance test demonstration, the equipment and steps chosen are used to reflect the current state of development and may deviate in design and appearance to the final product

The equipment used for the Acceptance Test Plan are as follows:

- CAN Bus System Simulator (referred to as “simulator”)
- CANtech Reader (referred to as “reader”)
- Smartphone with an alpha build of the app

The simulator for the CAN Bus system will be constructed and is meant to replicate how a car performs. The simulator will output a repeatable list of data frames that mimic an actual car’s telemetrics. At the time of writing, it is probable that the simulator will most likely consist of an Arduino and a CAN Bus Shield. The code inside the Arduino will simply iterate through a list of data frames, waiting to be read by the reader.

If there is any deviation when performing the test, leave a note and move on as best as possible. It is recommended to execute each and every acceptance test plan as ordered.

Please contact CANtech if compliance is not met or results deviate from the expected.

Acceptance Test Plan - Alpha			
Steps	**Expected Results**	**Observations**	**Comments**
1. Connect the reader onto the simulator. Ensure that both the reader and simulator is turned off.	The reader connects with minimal difficulty. The reader’s light remains off.		
2. Power the simulator and reader on.	The reader’s light turns on as it receives power.		
3. Open the software application on the smartphone. Ensure that the smartphone is not connected to the software application on the reader.	The main menu of the software application is displayed. The software application indicates that it is not connected to any device.		

4. Connect the smartphone to the reader via Bluetooth. The distance from the reader and the smartphone must be no more than 10m.	The smartphone successfully connects to the reader. The software application indicates that it is connected to the reader.		
5. After 5 minutes, check the status of the smartphone application.	The software application indicates that it is still connected to the reader.		
6. Disconnect the connection between the reader and the smartphone.	The software application indicates it is no longer connected to the reader.		
7. Remove the existing Bluetooth pairing between the reader and the smartphone. Repeat steps 4-6.	The expected results from steps 3-5 results are observed.		
8. Click on 'Settings'. Click on the 'Acceptance Test' option. Wait for 5 minutes then observe the software application	The software application displays the engine RPM, the engine oil temperature and vehicle speed within reasonable bounds. The values are updated at least once per 5 seconds.		
9. Click on 'Logs' and select the newly created log. Compare the values of the logs and the scripted values of the simulator.	The repeating pattern in the newly created log matches the values of the simulator.		
10. Close the software application on the smartphone. Wait for 5 minutes then open the software application	The main menu of the software application is displayed. The software application indicates that it is connected to the reader.		

Table 12-2: Acceptance Test Plan – Alpha