Engineering Standards and Future Proofing:

Partners:
Enes Yazici, App Design Engineer, eyazici@sfu.ca
Choong Jin Ng, Project Lead, jinn@sfu.ca
Nicholas Lau, Software Engineer, nla52@sfu.ca
Ranjoat Chana, *Systems Engineer*, ranjoatc@sfu.ca
Takehiro Tanaka, App Design Engineer, tta46@sfu.ca
Win Aung, Hardware Engineer, raung@sfu.ca

Contact:
Ranjoat Chana
ranjoatc@sfu.ca

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

Date Issued:
October 6, 2020

Version:
0.0.1

Contents

1.0 Introduction

CANnect is an open source hardware and software platform, consisting of a CAN bus reader and supporting software application for android devices. The CANnect reader can be plugged into the OBD-II port of a vehicle to access information about the state of a vehicle through the CAN bus system. This information is transmitted via a Bluetooth connection to our smartphone, where the data is interpreted for users to easily learn about the health of their vehicle systems. CANnect also offers modules that communicate additional information to the reader via a stateless WiFi connection. The GPS module provides the user with its location coordinates and other GPS related data which is passed to the user either through Bluetooth or cellular networks. The Sensor module provides the user with data related to the acceleration or orientation and angular velocity on a particular point on the vehicle.

CANnect aims to provide consumers with a customizable solution for their automotive needs, designed to monitor the health of their vehicle and track their vehicle's performance. Our product being open source further allows for a community of engineers and car enthusiasts to improve on our design and collaborate on new solutions for the automotive industry.

2.0 Engineering Standards

The standards outlined in this document will ensure the quality, reliability, and safety of the design solution. They are the technical criteria and methods used to ensure good engineering practices are maintained throughout the design process, and life of the product [1]. The engineering standards are outlined below for the hardware and software components of our CANnect reader and software application design solution, as well as the additional GPS and Sensor modules supported by the reader. It also deals with our approach to licensing our open source platform.

2.1 Software Application

CANnect will be designed to meet various engineering standards published by the IEEE, and the ISO to ensure performance and safety. The following engineering standards will be considered for the design of CANnect and will be used during the software development cycle.

Standard ID	**Description of Engineering Standard**
IEEE 1074	Standard for developing a software project life cycle process
ISO/IEC/IEEE 29119	Standard for developing a software testing
IEEE 1063	Standard for software user documentation

Table 2.1-1 Software Application Engineering Standards

The software application is developed using the Spiral method [2] to fit our incremental, risk-oriented cycle module and are evolving requirements and prototypes. Consistently, we will be testing as defined in ISO/IEC/IEEE 29119 with the focus on functional stability, reliability, security, maintainability, and portability [3]. This will be done through automated UI tests, functional testing and on physical realisations as much as possible (i.e. multiple devices). The user manual of the software application will be structured according to IEEE 1063.

There is ongoing research for Google Firebase [4] which will act as a server to store information received from the various hardware modules, ensuring the latest information is readily available on the app. Due to how new this feature is at time of writing, it is not included in Table 2.1-1.

2.2 CANnect Reader

The CANnect reader will connect to the OBD-II port of the vehicle and communicate with the smartphone device via a Bluetooth connection. To ensure the reader safely operates in relation to the vehicle connections and user handling, CANtech will incorporate the standards listed in Table 2.2-1.

Standard ID	Description of Engineering Standard
Iso15031-5:2015	Specifies the OBD-II connector and five supported protocols [5].
ISO 15765-4:2016	Road vehicles - Diagnostic communication over Controller Area Network (DoCAN) - Part 4: Requirements for emissions-related systems [6]
IEEE 802.15.1-2002	IEEE Standard for Telecommunications and Information Exchange Between Systems [7]
CAN/CSA-C22.2 No 0-10 (R2018)	General Requirements - Canadian Electrical Code, Part I [8]
CAN/CSA-C22.2 No 60950-1-07 (R2015)	Information Technology Equipment - Safety - Part 1: General Requirements [9]

Table 2.2-1 CANnect Reader Engineering Standards

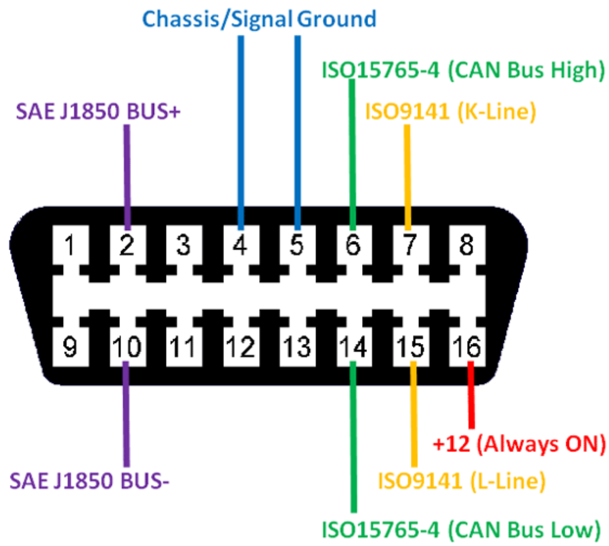

Figure 2.2-1: OBD-II Pin Diagram [10]

As outlined in Figure 2.2-1, the OBD-II specification [10] [11] calls for the above pin diagrams and related OBD-II protocols for communication with the car. We will mainly focus on ISO 15765-4 standard which defines CAN communication link and other network layer protocol applications [6]. IEEE802.15.1 outlines the Link Manager Protocol for Bluetooth technology which specifies the short-range connectivity for portable devices [7], used in our design between the reader and smartphone device. CAN/CSA-C22.2 outlines the electrical safety guidelines electrical equipment for our reader to operate under a vehicle's normal operation [8] [9].

2.2.1 OBD-II

The Controller Area Network or CAN bus system enables the communication between the various electronic control units or ECUs in a vehicle. ECUs control electrical systems in a car, such as the engine, transmission, brake, speed, etc. The Onboard Diagnostic II or OBD-II is a diagnostic system incorporated in vehicles since 1996. The OBD-II connector of all vehicles supports five protocols, including support for the CAN protocol since 2008 [12]. The reader will interface with the CAN bus data lines, CAN High and CAN Low. to transmit and receive CAN frames. This can be either 11-bit (CAN 2.0A) or 29-bit (CAN 2.0B) identifiers [6]. The reader will pass this information to the software app for interpretation.

At time of writing this document, research is ongoing for the incorporation of more OBD-II standards beyond ISO 15765-4 (i.e. CAN Bus). Specifically, we aim to include the OBD-II protocols from 1996 to 2008 before CAN Bus supersedes them. They are outlined in the Appendix B and will follow a similar procedure to CAN Bus.

2.3 GPS Module

The GPS (Global Position System) module is added to the CANnect for anti-theft functionality and will send data to the software application. This can be done via Bluetooth or cellular data (i.e. LTE). The GPS module itself will comply with the standards outlined in Table 2.3-1. At time of writing, this section is subject to ongoing research; future versions will address possible additions of standards such as GNSS (Global Navigation Satellite System).

Standard ID	Description of Engineering Standard
GPS Standard Positioning Service (5th Edition) (GPS SPS)	Standard for civilian GPS service currently in use as of April 2020 [13]
NMEA 0183	Standard for formatting of GPS information from NMEA (National Marine Electronics Association)
GPGGA or GPGLL	Common NMEA sentences transmitted by GPS receivers.

Table 2.3-1 GPS Module Engineering Standards

Users of the GPS module can expect ≤ 15m 95% Horizontal and ≤ 33m 95% vertical error in Worst Site Position. This is acceptable as the purpose of the GPS module is for anti-theft, and most vehicles can be found within a 15m radius. On average, it performs better with accuracy of ≤ 8m Horizontal and ≤13m Vertical with 95% confidence interval [13]. GPS strings will be formatted according to NMEA 0183 (see in Appendix A for a sample data message).

2.3.1 Cellular Module and Communication to Firebase

The cellular module being implemented will incorporate the use of LTE (Long Term Evolution) that is specifically focused on IoT applications that is deployed in Canada, namely LTE CAT-M1 which falls under the category of eMTC (enhanced Machine Type Communication) and broader category of LTE-M or LTE-MTC (LTE-Machine Type Communication). Listed below are the relevant standards that will be used for the design of the cellular component and any related connected services. The module will work with an ESP32 via UART.

Standard ID	Description of Engineering Standard
3GPP Release 13 (LTE CAT-M1)	Standard containing low bandwidth and power consumption networking applications. This incorporates related IoT features relating to power saving features such as eDRX (Extended Discontinuous Reception) and PSM (Power Saving Mode) [14].
IETF Transmission Control Protocol (TCP)	Standard created by IETF (Internet Engineering Task Force) for maintaining connections and providing reliable data transfer. Used as part of HTTP to send data reliably [15].

IETF Internet Protocol (IPv4 and v6)	Standard created by IETF for addressing the client and server. Both should be supported by the cellular module. Used to support the cellular connection [15].
Hypertext Transfer Protocol (HTTP/2) with HTTPS extension	Web standard for communication to Firebase server. Using HTTPS extension (Secure). Used as part of REST APIs which is used in Firebase [16].
Representational state transfer (REST)	Architecture for APIs and how to interface between a client and a server using HTTP [17].

Table 2.3.1-1 Cellular Module Engineering Standards

2.4 Sensor Module

The sensor module will be a 6DoF device that will connect to the reader via a connectionless wifi protocol, with the aim to monitor the acceleration of the car as well as its roll, pitch and yaw at different points of the car. It also offers possible future implementations for more sensors by the user's discretion by interfacing the ESP32. At time of writing, this section is subject to ongoing research; future versions will address possible additions of standards, specifically in addressing the 6DoF sensor. The standards are listed in Table 2.5-1.

Standard ID	Description of Engineering Standard
I^{2}C Communication Protocol / SPI Protocol	Serial Communication Protocols [1]
CAN/CSA-C22.2 No 0-10 (R2015)	General Requirements - Canadian Electrical Code, Part II [9]

Table 2.4-1 Cellular Module Engineering Standards

The sensor can use the I^{2}C protocol and would feed the ESP32 in-built with raw data information. The ESP32 will then interpret the data and passed the information to the reader via stateless WiFi. In the future, the sensor module will allow the user to fit more additional sensors including an analog signal for the ESP32's in-built Analog-to-Digital Converter (ADC). These will have to comply with the electrical code standards outlined in CAN/CSA-C22.2 [9].

2.5 System Internal Communication Standards

As all modules (i.e. reader, cellular and GPS combined unit and 6DOF and ADC sensors) will have an ESP32 microcontroller, they will all communicate over stateless Wi-Fi (i.e. ESP-NOW) [18]. The reader will communicate to the software application via Classic Bluetooth. Booth communication methods are outlined in Table 2.5-1.

Standard ID	Description of Engineering Standard

Espressif ESP-NOW	Protocol using 2.4 GHz Wi-Fi to provide a stateless connection. This has less overhead and simplifies connections versus regular Wi-Fi (IEEE 802.11 series) [18].
Bluetooth Core v5.2	Bluetooth communication specification to provide 2.4Ghz Wireless communication [19].

Table 2.5-1 General Module Engineering Standards

2.6 Hardware and Software Open Source Licenses

Upon the release of CANnect, we aim to publish the designs, code and documents under Open Source licenses listed in Table 2.6.1. CERN-OHL-P deals with hardware and MIT License deals with software.

Standard ID	Description of Engineering Standard
CERN-OHL-P	Free to use, study, modify, share, and distribute designs under the CERN Open Hardware License Version 2 - Permissive [20].
MIT	Free to use, copy, modify, publish, distribute, sublicense, and sell the copies under the condition that the copyright notice and permission notice is included in the software [21].

Table 2.6-1 Hardware and Software Licenses

3.0 Risk Management

As with any product and venture, there are risks that are identified and possible mitigations. While this list does not cover all possible risks, we aim to address the immediate ones that one may raise.

3.1 OBD-II Protocols

Standard PIDs are the OBD-II PIDs as defined by SAE J1979 which are incorporated into the legislated OBD-II stack as ISO 15031-5 [5]. While all vehicles support the standard PIDs, the reader may not be able to support all PIDs received from the car due to the presence of manufacturer defined custom PIDs. Moreover, cars that are produced before 2008 are not compatible with the current state of the reader (it only reads CAN messages). This is being addressed at time of writing by changing the design of our reader’s hardware to allow reading of all OBD-II protocols. To keep our proposed deadlines, work will be done in parallel between the development of the current state of the reader and the new, proposed reader for production.

However, it is likely that we will not be able to implement all the standard PIDs and most of the custom PIDs to meet the deadline. As the project will be open source, for each type of OBD-II, the code will be easily modifiable for adding new PIDs into the list. To minimize any issues with potentially corrupted data arriving on the reader and causing car malfunctions, a more robust protocol will be implemented which includes the validation of data using a checksum. Manufactured-defined PIDs will either be reverse-engineered or will be acquired through talking with vehicle manufacturers directly. For the user's perspective, additional support of standard PIDs and custom PIDs will be added through software updates in the future.

3.2 Anti-Theft

CANnect will include an anti-theft car tracking module which mostly likely use GPS data. The plan is to forward GPS data from the car to the app via a remote database server. Therefore, a SIM card with a data plan and LTE module will be used to transmit the location information every hour or upon request by the user; A more flexible model may be designed later.

The reliability of this anti-theft feature relies on the carrier being used and their coverage, as well as the ability for the GPS module to find satellites which are factors outside of CANtech's control. Fallback measures for internet access are limited as open Wi-Fi may not always be available. However, given the current spectrum deployment by TELUS [22], the testing carrier, for LTE on which CAT-M1 is deployed, this is not likely to be an issue [23]. Customers should consult with carriers and resources using similar cellular tower databases linked in the references to determine the best option for them.

Since GPS calculates the current position by acquiring signals from several satellites, the module must be installed to have a clear line of sight when receiving data for better accuracy. Urban areas may post challenges to obtain accurate GPS coordinates due to electromagnetic inference. Nevertheless, as discussed in 2.3 GPS Module, our proposed implementation will be sufficient for users to know their car is stolen. A possible alternative, that could also be used in conjunction with GPS, is to use Google Map's Geolocation API through nearby Wi-Fi and cell towers [24].

3.3 Privacy and Security

The integration of cellular communication with a server backend to handle the exchange between the reader-side and app-side means there are more points of failure and risk relating to security. As such, the data sent via cellular communication will be encrypted as part of the 3GPP standard for LTE and Firebase, the backend database server by Google, follows many major ISO standards for privacy and security (e.g. ISO 27001) [25]. Data that is stored on Firebase will only contain data such as user identification and GPS coordinates, but none relating to personal information. Furthermore, data received by the app from the reader will only be stored locally in smartphones.

3.4 Competition

The risk of making our system open source is that competitors and users may opt to copy our designs, repackage them and sell them as their own. This is highly likely as most OBD-II readers in the market are saturated with cheap clones using the pirated design of the ELM [26] [27]. We aim to counter this by offering excellent customer service and support beyond simple product acquisition, and to continuously revise our designs to "stay ahead".

3.5 Interference while Driving

We identified a potential risk of the CANnect system (reader and app especially) may interfere with a typical operation of the car, whether distracting drivers or directly interfering with the car. We will stick closely with the OBD-II standards and carefully ensure PIDs sent and received are compliant with the standards. Furthermore, the reader will only be used to read information rather than writing into the car's CAN bus system which ensures that faults will not cause interference with the car's internal electronics. Lastly, the app will be designed to minimise disruption during normal operation and to only interact with the car via the reader.

4.0 Human Factors, Usability and Design

Because of the open-source nature of CANnect, future revisions of designs are expected to incorporate fixes and adding in new features. The reader will be designed with all OBD-II protocols to ensure that only software updates are needed to unlock the full potential of the hardware. Similarly, the modules can be purchased separately and are designed such that they can be swapped with better versions in the future. The app itself will be the primarily brains of the entire system through deciphering the information fed from the reader and modules. As such, we expect the app itself to be updated more easily and more frequently when including new features and bug fixes in the future, limiting the need to acquire new hardware for new features.

4.1 Reader & Modules

The reader and its relevant modules will be encased in shells to protect the internal components from the elements and the user. The reader will be installed in the car with a provided OBD-II cable to the OBD-II port while the other modules will be installed as specified by the installation protocol. While the reader falls under normal maintenance of the car, the other modules will require professional installation as they must tap to the car battery's 12V line and ground. This is intentional as it offers a possible more flexibility on where the modules can be installed. A comprehensive manual for installation and use will be provided to the users.

Because the designs of the reader and modules are available to the users, users may introduce their own additions to customise future functionality or to "pick and choose" which features they want. However, CANtech will only provide technical support for user modifications.

4.2 App

The app is designed with the intention that it can be used during a car's normal operation. Therefore, aspects of the user interface is designed to be limit interference of the driver's operation of the car such as a more simplified interface, more reliance on symbols and haptic feedback rather than text, and constant optimisation of the user interface after repeated field tests. Because it is easier and more frequent for the app to be updated, we can continue to refine the app and features even after the launch of CANnect.

4.2.1 Map

The map activity inside the app show the car's current location via Google Maps SDK. This would be taken from both the phone and the GPS module. These GPS coordinates may be read either directly from the devices or from a Google Firebase server, whichever one is more recent. This feature allows users to know the location of the car, serving the purpose as an anti-theft device. This feature was introduced after the alpha-phase prototype beyond simply reading the GPS coordinates from the phone and displaying it to the user.

4.2.2 Data Communication and Serviceability

Ongoing development is focused on improving Bluetooth communication. For the alpha phase, the app reads the data in approximate 60 bytes with optimized system sleep second with 2 seconds. This is unacceptable as this will limit the real-time application of the CANnect reader, especially with the inclusion of new features that will send more information through this pipe.

The design of how we deal with Bluetooth is changed for the app to read the data per single byte of stream, and identifies data frames with end-of-message delimiters. A Bluetooth protocol between the app and reader is also being developed to provide more robust communication. Further research is also done on utilisation of pipelines to feed data directly to activities rather than the current implementation of passing data through multiple activities.

4.2.2 Cellular

For cellular communication, the planned design is to use an IoT oriented module supporting LTE CAT-M1 for use in Canada which allows for small data transfers with lower power consumption. Power saving features relating to CAT-M1 such as sleep modes and disconnection capabilities will also be considered and implemented if feasible. Since it will be controlled via an ESP32 over UART, it will allow for easier development and allow for future upgradability as well as serviceability should changes be required in the future. Given that cellular modules also support standardized commands (AT commands), this also facilitates serviceability and troubleshooting. Communication to Firebase will be performed over REST APIs as it is simple and robust.

5.0 References

[1] Werner Sölken, "What are Engineering Standards?," 2020. [Online]. Available: http://www.wermac.org/documents/engineering_standards_what_are.html. [Accessed 04 October 2020].

[2] IEEE, "IEEE Standard for Developing a Software Project Life Cycle Process," *IEEE Std 1074-2006,* pp. 1-110, 28 July 2006.

[3] ISO/IEC JTC 1/SC 7 Software and systems engineering, "Software testing — Part 1: Concepts and definitions," September 2013. [Online]. Available: https://www.iso.org/standard/45142.html. [Accessed 30 September 2020].

[4] Google, Google, [Online]. Available: https://firebase.google.com/. [Accessed 05 October 2020].

[5] International Organization for Standardization, "ISO 15031-5:2015(en) - Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 5: Emissions-related diagnostic services," International Organization for Standardization, 2015. [Online]. [Accessed 06 October 2020].

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

[7] IEEE, "IEEE 802.15.1-2002 - IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Network," IEEE, 15 April 2002. [Online]. Available: https://standards.ieee.org/standard/802_15_1-2002.html. [Accessed 06 October 2020].

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

[9] CSA Group, Canadian Electrical Code, part II, CSA Group, 2015.

[10] Components101, "ODB-II Connector," 2020. [Online]. Available: https://components101.com/connectors/obd2. [Accessed 02 June 2020].

[11] "OBD-II Protocols," 2020. [Online]. Available: http://www.obdtester.com/obd2_protocols.. [Accessed 06 October 2020].

[12] 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].

[13] Department of Defense, "GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE SIGNAL SPECIFICATION 5th Edition," National Coordination Office for Space-Based Positioning, Navigation, and Timing, April 2020. [Online]. Available:

https://www.gps.gov/technical/ps/2020-SPS-performance-standard.pdf. [Accessed 05 October 2020].

[14] GSMA, "LTE-M Deployment Guide to Basic Featyre Set Requirements Version 3.0," GSM Association, June 2019. [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf. [Accessed 03 October 2020].

[15] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," December 1998. [Online]. Available: https://tools.ietf.org/html/rfc2460. [Accessed 05 October 2020].

[16] M. Belshe, R. Peon and M. E. Thomson, "Hypertext Transfer Protocol Version 2 (HTTP/2)," May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7540. [Accessed 05 October 2020].

[17] R. T. Fielding, "CHAPTER 5 - Representational State Transfer (REST)," 2000. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. [Accessed 02 October 2020].

[18] Espressif Inc, "ESP-NOW User Guide Version 1.0," Espressif Systems, 2016. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp-now_user_guide_en.pdf. [Accessed 29 September 2020].

[19] Bluetooth SIG, Inc. , "Bluetooth Core Specifications v5.2," Bluetooth SIG, Inc. , 31 December 2019. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=478726. [Accessed 03 October 2020].

[20] 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].

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

[22] TELUS, "TELUS Coverage Map," TELUS, [Online]. Available: https://www.telus.com/en/bc/mobility/network/coverage-map. [Accessed 06 October 2020].

[23] Government of Canada, "700 MHz Spectrum Auction-Process and Results," Government of Canada, 19 February 2014. [Online]. Available: https://www.canada.ca/en/news/archive/2014/02/700-mhz-spectrum-auction-process-results.html. [Accessed 06 October 2020].

[24] Google, "Geolocation API Overview," Google, [Online]. Available: https://developers.google.com/maps/documentation/geolocation/overview#introduction. [Accessed 06 October 2020].

[25] Google, "Privacy and Security in Firebase," Google, 05 October 2020. [Online]. Available: https://firebase.google.com/support/privacy. [Accessed 06 October 2020].

[26] "ELM327 USB Vehicle OBD-2 Scanner Tool," DigitalPoint, [Online]. Available: https://priuschat.com/threads/elm327-usb-vehicle-obd-2-scanner-tool.83427/. [Accessed 06 October 2020].

[27] "Why you shouldn't buy ELM327 clones," [Online]. Available: https://www.scantool.net/forum/index.php?topic=2379.0. [Accessed 06 October 2020].

[28] J. Stoltzfus, "Your Car, Your Computer: ECUs and the Controller Area Network," Techopedia Inc, 07 August 2020. [Online]. Available: https://www.techopedia.com/your-car-your-computer-ecus-and-the-controller-area-network/2/32218. [Accessed 01 October 2020].

APPENDIX A: GPS Message Format

The idea of NMEA (National Marine Electronics Association) strings is that its information is self-containing and do not require further interpretation. For GPS NMEA strings, this takes on the form of GPGGA and GPLLL formats.

In Figure A.1, the standard message format of Global Positioning System Fix Data (GPGGA) which contains position, time and related data.

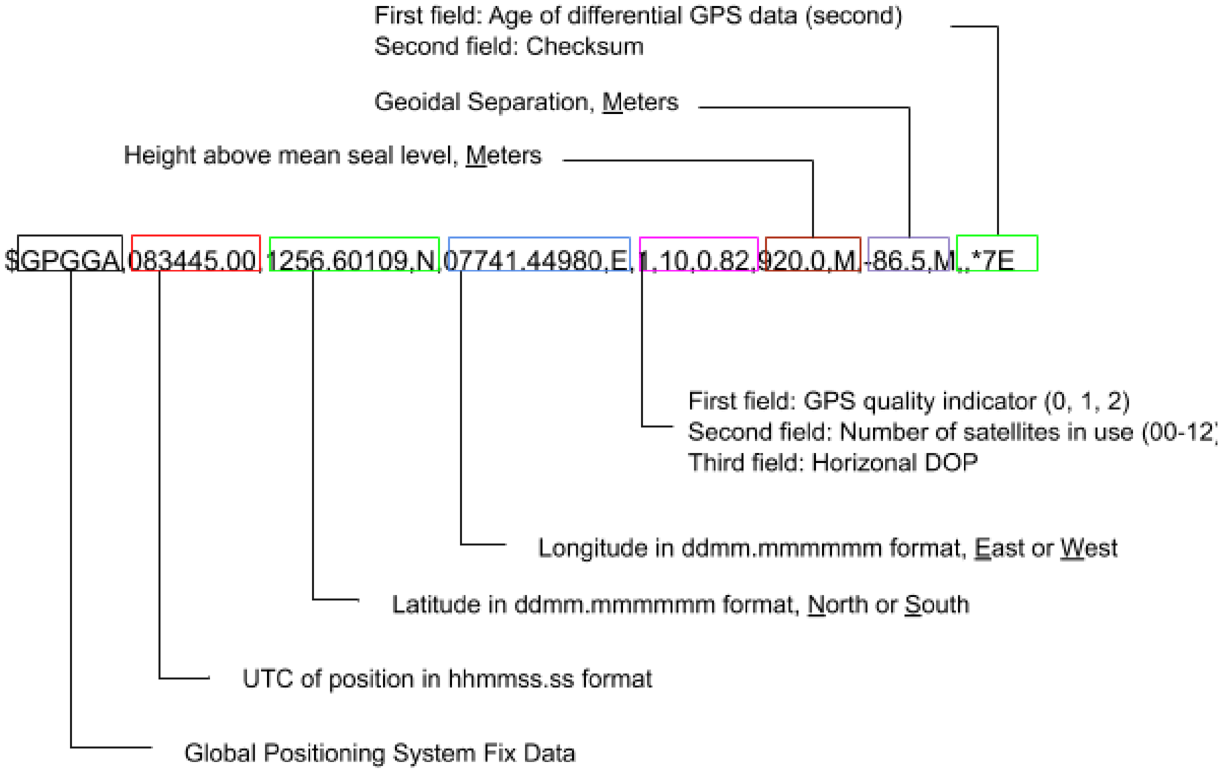

Figure A.1 - GPGGA Message Format

In Figure A.2, the standard message format of Geographic Position, Latitude / Longitude (GPGLL).

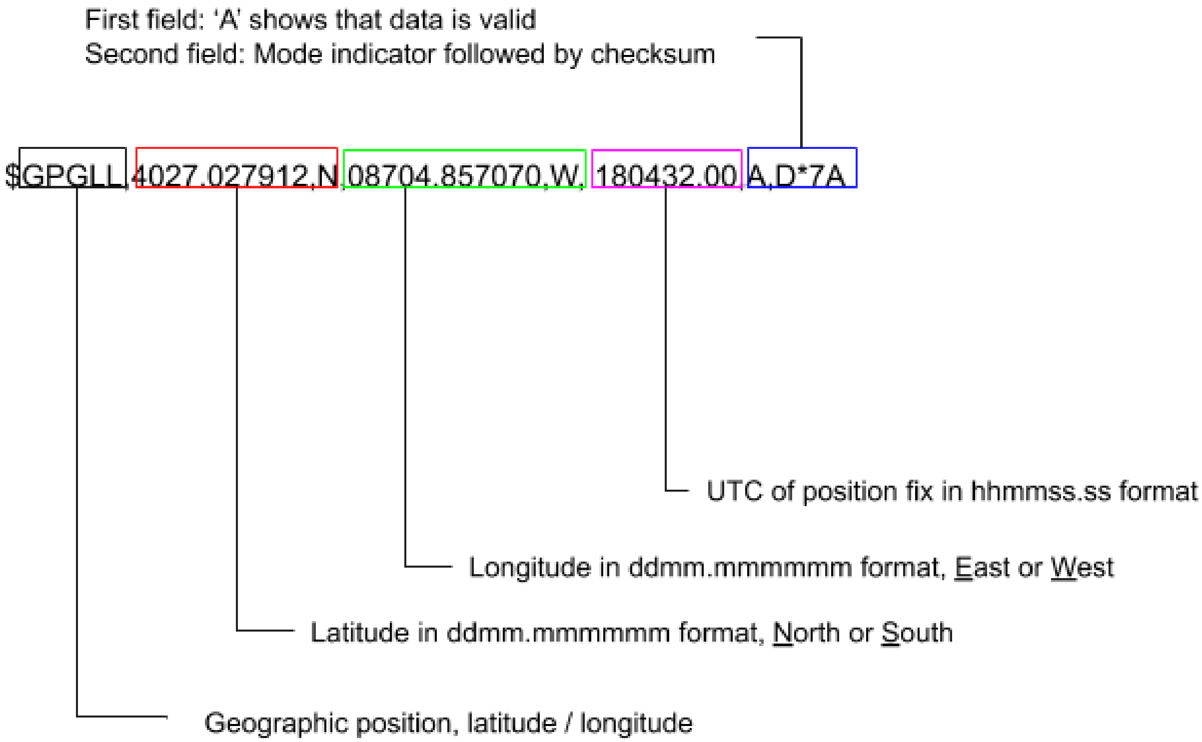

Figure A.2 - GPGLL message format

APPENDIX B - OBD-II Protocols

The OBD-II standard supports five communication protocols [5] which are listed below. As illustrated in Figure 2.2-1 [10] and in ISO 15765 [6], all cars must have the OBD-II port as of 1996 and must support CAN from 2008 onwards. In Table B.1, the protocols are listed directly to the pin diagram in Figure 2.2-1.

The primary reason of choosing CAN Bus over the other OBD-II protocols is for a working Proof-of-Concept and universal protocol standard for most modern cars. Nevertheless, there is significant interest in working on a reader that supports all OBD-II protocols and the related standards for it will be introduced to this document in the future.

Standard	**Pin 2**	**Pin 6**	**Pin 7**	**Pin 10**	**Pin 14**	**Pin 15**
J1850 PWM	Must have	-	-	Must have	-	-
J1850 VPW	Must have	-	-	-	-	-
ISO9141/14230	-	-	Must have	-	-	Optional
ISO15765 (CAN)	-	Must have	-	-	Must have	-

Table B.1 - Supported Protocols Based on OBD-II Connector Pins [11]

The five supported OBD-II communication protocols are:

- SAE J1850 PWM
- SAE J1850 VPW
- ISO 9141-2
- ISO 14230-4 (KWP200)
- ISO 15765/SAE J2480

APPENDIX B.1 SAE J1850 PWM

- Diagnostic bus utilized by Ford vehicles
- Access through pin connectors 1 and 2

APPENDIX B.2 SAE J1850 VPW

- Diagnostic bus utilized by GM vehicles
- Access through pin 1 connector

APPENDIX B.3 ISO 9141-2

- Old protocol supported on European model vehicles manufactured between the years 2000 and 2004
- The OBD-II pin 7 must be used and pin 15 is optional

APPENDIX B.4 ISO 14230-4 (KWP200)

- Protocol supported on vehicles manufactured in 2003 and onwards
- The OBD-II pin 7 connection is used
- Two variants based on how the communication is initialized

APPENDIX B.5 ISO 15765/ SAE J2480 (CAN Bus)

- Support for this protocol was made mandatory for all vehicles manufactured in 2008 and onwards in the US
- The OBD-II pin connections 6 and 14 should be accessible
- Four variants of this protocol based on the identifier length and speed of bus