Oregon State University
College of Engineering

CS Capstone Technology Review

November 21, 2017

Axolotl

Prepared for

Kevin McGrath ______________________ Signature ________ Date

Prepared by

Group 15
Team Wombat

Victor Li ______________________ Signature ________ Date
Ryan Crane ______________________ Signature ________ Date
Nicholas Wong ______________________ Signature ________ Date

Abstract

This document is a technology review that will describe a subset of project components and look into the technologies that can be used to implement them. Three technologies will be identified for each component described. Each identified technology will be examined individually before the three technologies are discussed in comparison to one another. Conclusions will be drawn from the merit of each technology and the most ideal technology will be chosen as the technology used for implementation. The components examined in this technology review include: navigation software, vehicle data connectors, vehicle data connector software, and FM receiver modules.

Contents

1 Navigation Software

Our Axolotl infotainment system must offer a navigation suite capable of routing and mapping with full offline capability. In order to implement this functionality in a reasonable timeframe, we need to utilize an open-source navigation solution. Navigation software options must be compatible with the NVIDIA Jetson TX2's Linux operating system and offer routing capabilities at a minimum in order to be considered viable for our project. The ideal navigation software for our project will be selected based on versatility, offline capability, and availability of documentation.

1.1 Open Source Routing Machine

The Open Source Routing Machine, or OSRM, is a routing system developed by Dennis Luxen and written in C++. The Open Source Routing Machine uses map data from the OpenStreetMap project to provide routing capabilities. The system can generate a routing itinerary of turn-by-turn instructions, find the fastest or shortest route to a destination, and also consider turn lanes and turn restrictions when generating a route. OSRM uses a simple map data format that allows for the use of other street map data sets other than that provided by the OpenStreetMap project [1]. The source code for the OSRM can be obtained from the OSRM's github repository [1] and is licensed under a 2-clause open-source BSD license [2]. The 2-clause BSD license is a very permissive license that enables the software to be freely redistributed and modified on the condition that any and all redistributions/modifications credit the original developers of the OSRM software [10]. Additionally, the OSRM offers a comprehensive set of documentation on its website for developers. The documentation features well-detailed explanations of OSRM code and functionality [3].

1.2 NavIt

NavIt is a navigation system and routing engine project written in C and C++ [4]. The NavIt system features turn-by-turn navigation that supports full offline capability and the ability to use map data from numerous sources. By default, NavIt uses its own map data, but it can utilize map data from the OpenStreetMap project [4]. NavIt can convey its turn-by-turn directions by voice in 49 languages as of its current release [6], and features a 2.5D map view, customizable routing profiles, as well as address and point of interest search [4]. NavIt is designed to support GPS sensors and can read a vehicle's current location directly through a GPS receiver or indirectly through posted GPS event data [7]. The source code for the NavIt navigation system is available from the project's github repository and is licensed under the GPLv2 license [5], which states that the software can be used free-of-charge on the condition that the source code of any redistributions or modifications be available for others to utilize, redistribute, and modify as well [11]. The NavIt project provides minimal documentation for developers via its wiki, however there are numerous third-party guides written for the NavIt software [4].

1.3 libosmscout

libosmscout is an offline mapping and routing library developed by Tim Teulings [8], implementing all of the important features of a navigation application in a package that can be run by low-end hardware such as handheld devices [9]. The libosmscout library is written in C++ [8] and is designed to utilize map data from the OpenStreetMap project in order to provide offline mapping, routing, and destination lookup through location or point-of-interest search [9]. libosmscout offers a variety of APIs or access point interfaces for developers to interact with the software. These include high-level APIs for greater abstraction of the inner workings of the libosmscout library, as well as lower-level APIs for greater control over the the library itself. libosmscout is designed to minimize its use of disk storage space for its maps, allowing even continental map data to be stored on the storage medium it utilizes [9]. libosmscout is available from the libosmscout SourceForge page and utilizes the LGPL license [9], meaning that licensees are obligated to share the source code of any redistributions/modifications they make to the libosmscout library, but are not obligated to do the same with any software that uses the library [12]. A considerable amount of documentation for the libosmscout library is available on its SourceForge page [9].

1.4 Discussion

OSRM, NavIt, and libosmscout support similar sets of map data; each can use OpenStreetMap as its map data source. OSRM and NavIt are more versatile than libosmscout as both may be adapted for other sets of street data, whilst libosmscout is designed specifically for the OpenStreetMap data format. NavIt and libosmscout are superior options in comparison to OSRM with regards to offline capability, as both NavIt and libosmscout support offline mapping. OSRM does not directly support offline mapping, though it is possible to develop a workaround by creating a map server on the TX2 and have OSRM query the system itself rather than the OpenStreetMap servers for map data.

OSRM's documentation is very detailed compared to libosmscout and NavIt. The OSRM documentation provides coding and usage examples in a multitude of programming languages including Java, Swift, and Python. libosmscout provides many guides for utilization of the library, however these are relatively simple and not in-depth. NavIt only provides basic installation instructions and limited documentation on its wiki, though the abundance of third-party guides for the software somewhat offset this disadvantage. OSRM also features a BSD license that is less restrictive than the GPL and LGPL licenses used by NavIt and libosmscout.

However, NavIt is superior to OSRM and libosmscout in one key respect: NavIt is a complete turn-by-turn navigation application and can navigate to a destination as well as guide users by voice with turn-by-turn directions. This allows the system to guide the user to their destination one step at a time without requiring them to interpret a large routing itinerary in order to find their next turn instruction. By comparison, OSRM is only routing engine capable of generating routes and route itineraries and is not built to provide turn-by-turn guidance; libosmscout functions in much the same way but is formatted as a routing library instead.

1.5 Conclusion

We chose NavIt as our navigation software solution as it is the most comprehensive navigation solution. NavIt offers native offline capability, the ability to use a variety of map data sets, and true navigate to point capability. Neither OSRM nor libosmscout offer all of these features in one package. NavIt also offers voice guidance that eliminates the need to develop a voice system from scratch to relay directions from the route produced by a routing engine or routing library. The support of GPS sensors makes the system even more appealing as it allows GPS receivers to be easily wired in and used for vehicle location whilst navigating. Overall, NavIt adequately satisfies all selection criteria and is the most capable option out of the three examined. It can compete with the GPS navigation systems offered by car manufacturers and smartphones with its navigate-to-point and voice guidance capabilities, features not offered by OSRM or libosmscout.

2 Vehicle Data Connectors

A critical requirement of the Axolotl infotainment system is the ability to read vehicle data, format it, and save it to a black box log in the manner of an event data recorder. The NVIDIA Jetson TX2 we are using does not have a dedicated method of connecting to automotive computers; thus, some kind of data connector is required in order to gain access to the data. Vehicle data connectors must be compatible with the OBDII vehicle connector standard to be considered viable for our project. The ideal connector for our project is one that is versatile, inexpensive, and offers high performance and reliability.

2.1 BAFX OBDII Diagnostic Interface

The BAFX OBDII Diagnostic Interface is a wireless vehicle diagnostic scanner tool developed and manufactured by BAFX Products. The device costs $21.99 USD retail and can be ordered directly from BAFX Products at bafxpro.com. The OBDII Diagnostic Interface is designed to read vehicle data in near real-time from an automotive OBDII port and make that data readily available over a Bluetooth connection. Statistics that the BAFX system can read include (but are not limited to) engine RPM, vehicle speed, air temperature, fuel metrics, and any OBDII diagnostic trouble codes in order to provide supplementary gauges or display vehicle diagnostic trouble codes that can be used to determine the source of vehicle problems. The BAFX system also supports a variety of CAN bus protocols [13].

2.2 OBDLink MX Bluetooth

The OBDLink MX Bluetooth is a wireless diagnostic scanner tool similar to the BAFX OBDII Diagnostic Interface [14]. The OBDLink MX Bluetooth is designed and marketed by OBD Solutions, and can be purchased from ScanTool.net for $79.95 USD retail [15]. Like the BAFX OBDII Diagnostic Interface, the OBDLink MX Bluetooth supports reading a wide variety of vehicle data from an automotive OBDII port. The OBDLink MX Bluetooth is also backwards-compatible with the ELM327 standard [14], a microcontroller created by ELM Electronics designed specifically for the purpose of interacting with vehicle OBDII ports [16]. Support for the ELM327 standard allows the OBDLink MX Bluetooth to be used with a greater range of OBDII software, as some software only supports ELM-based devices [16]. The OBDLink MX additionally supports all OBDII protocols and is highly secure due to a complex Bluetooth PIN. The OBDLink MX is also capable of automatically sleeping or waking in order to prevent battery drain if left plugged in for an extended period of time [14].

2.3 Vgate iCar 2 WiFi

The Vgate iCar 2 WiFi is a wireless OBDII vehicle diagnostic tool designed for automotive professionals and enthusiasts. The Vgate iCar 2 WiFi can be purchased from OBD Innovations at a retail price of $39.99 USD. Like the other OBDII devices, the Vgate device supports the ability to read OBDII data, and also supports manufacturer-specific protocols (specifically Ford and GM protocols) that most inexpensive OBDII connectors don't offer. The iCar 2 is capable of auto-sleep after 30 minutes in order to preserve a vehicle's battery life if left un-driven for an extended period of time with the connector still plugged in. The iCar 2 transmits OBDII diagnostic data over a WiFi network, requiring other devices to join the WiFi network broadcast by the Vgate device in order to access OBDII data. The advantage of utilizing a WiFi network to transmit OBDII data is greatly increased range over that of a Bluetooth connection [17].

2.4 Discussion

The diagnostic interfaces are well-matched in terms of capability. All support OBDII protocols and communicate vehicle data wirelessly, although by different means. Both the BAFX diagnostic tool and the OBDLink MX transmit vehicle data via Bluetooth in comparison to the Vgate connector, which utilizes a WiFi network. The Vgate's use of WiFi to transmit OBDII data conflicts with our system as the Axolotl already utilizes the WiFi radios to connect to a mobile hotspot in order to provide internet-connected services (only one WiFi connection can be active at any time). The BAFX's and OBDLink's use of Bluetooth connectivity does not encounter this issue as the NVIDIA Jetson TX2's Bluetooth 4.0 can handle multiple Bluetooth devices at once, so the Bluetooth OBDII connectors will not interfere with other devices connected to the Axolotl via Bluetooth [18].

2.5 Conclusion

We chose the OBDLink MX Bluetooth OBDII connector as our vehicle data connector because of its versatility, low power consumption, and support for the ELM327 standard. Though the OBDLink MX Bluetooth is noticeably more expensive than the BAFX and Vgate connectors, it is more capable. The BAFX connector does not feature auto-sleep, nor does it explicitly support the ELM327 microcontroller. The Vgate connector does have auto-sleep but does not support the ELM327 microcontroller and does not synergize well with our Axolotl system due to its use of a WiFi network to transmit OBDII data. Overall, the OBDLink MX is the most capable and the least compromised, as it supports ELM327, consumes very little power, and is highly secure. Our client also has personal experience with the OBDLink MX Bluetooth and personally recommends the use of the connector due to proven capability and reliability.

3 Vehicle Connector Software

Both a physical vehicle data connector and data connector software is required to read vehicle data. Connectors only provide the physical data link; supplemental software is necessary in order to interpret the data signals from the data connector and parse it into readable and usable data. Project-viable vehicle data connector software must be open-source and compatible with OBDII diagnostic tools; data connector software will be selected for use based on versatility with respect to project goals.

3.1 Perl OBD-II Logger

The Perl OBD-II Logger project is an OBDII data reader tool developed to capture and log OBDII data from OBDII automotive scan tools. As the name suggests, the Perl OBD-II Logger is written in the Perl programming language and can operate on any system that has a Perl interpreter installed. Users of the software can specify a sampling rate that the logger will adhere to, for example, taking one reading every two seconds. The Perl OBD-II Logger supports two standards of automotive scan tool implementations; the ELM327 provided by ELM Electronics or the STN1110 standard provided by OBD Solutions. The software does not provide any support for managing vehicle diagnostic trouble codes. Extensive testing has been done with the Perl OBD-II Logger utilizing a variety of OBDII scan tools, including USB interfaces and Bluetooth interfaces [19]. The source code for the Perl OBD-II Logger can be found on the Perl OBD-II Logger SourceForge page, and the software is licensed using the Artistic License 2.0 [19]. The Artistic License 2.0 provides freedom of use and modification of the software, allowing for redistribution of the software on the condition that the original copyrights are indicated and any modifications of the software is indicated [22].

3.2 OBD GPS Logger

The OBD GPS Logger is a toolkit that is comprised of a collection of small Unix programs designed to each provide simple OBDII functionality. The OBD GPS Logger software is designed to log data from a OBDII port and as well as a GPS receiver; any logged data can then be written out as an output that can be easily read or forwarded as useful input to another subsystem. The OBD GPS Logger only supports ELM327-compatible OBDII devices and requires any connected GPS receivers to be compatible with gpsd [20], a GPS service daemon that makes GPS receiver data available via a TCP connection [23]. The source code for the OBD GPS Logger is available via downloadable packages on its website and is licensed with the GPLv2 license [20].

3.3 pyOBD

pyOBD is an open-source OBDII car diagnostic tool written in Python and developed by Donour Sizemore. The software was primarily developed to interface with low-cost OBDII diagnostic interfaces, most prominently the ELM-compatible diagnostic connectors. pyOBD can read diagnostic trouble codes, read measured vehicle metrics, as well as read vehicle status tests. On top of that, a python module called obd_io is provided for developers to interact with sensor data and diagnostic trouble codes, enabling this information to be used by another subsystem for logging or management. pyOBD source code is available for download from its supporting website in the form of debian or tar archives; the source is licensed using the GPL license [21].

3.4 Discussion

These three software options are quite varied in their capabilities. Both the Perl OBD-II Logger and the OBD GPS Logger support native logging functionality; by comparison, pyOBD does not. pyOBD instead provides an interface that allows users to access the OBDII data read by the software, requiring users to use this interface and develop their own logging solution if logging is desired. The OBD GPS Logger is more versatile than the Perl OBD-II Logger and pyOBD as it is capable of managing OBDII and GPS data, whereas Perl OBD-II Logger and pyOBD are limited to only OBDII. However, a key advantage that pyOBD has over the OBD GPS Logger and Perl OBD-II Logger is the ability to read and manage vehicle diagnostic trouble codes, which is an highly desirable optional feature of the Axolotl infotainment system.

3.5 Conclusion

We chose pyOBD as our project's OBDII software as it can fulfill all base project requirements regarding the OBDII port and can also be used to implement optional OBDII features. One of these optional features enables users to read, understand, and manage vehicle diagnostic trouble codes through the Axolotl's graphical interface; this may provide insight on vehicle issues and breakdowns. This feature requires our OBD software suite to be able to read and manage vehicle diagnostic trouble codes, thus the ability of pyOBD to interact with diagnostic trouble codes is highly desirable. The lack of logging capabilities with the pyOBD software does mean that we need to write our own data logging software utilizing the IO interface provided by pyOBD. Should issues arise during development or if we choose not to implement diagnostic trouble code functionality, Perl OBD-II Logger or OBD GPS Logger will be used as an alternative OBDII software solution.

4 FM Receivers

Our system's media suite must support FM radio playback and tuning. Additionally, our system must be able to read the RDS or Radio Data System information from the currently playing FM radio broadcast and display the encoded RDS text on the infotainment screen. In order to do this, we will require some type of FM receiver. Project-viable FM receivers must be able to support U.S. FM station bands of 87.5 to 108 MHz [29] and must be able to handle RDS data. FM receivers will be selected based on price and connection method.

4.1 SparkFun Si4703

The SparkFun Si4703 is a FM receiver module developed by SparkFun based on the Si4703 FM receiver chip designed by Silicon Labs. The SparkFun receiver is advertised as a tuner evaluation board for the Si4703 chip and can be purchased at a retail price of $19.95 USD from SparkFun.com [24]. The receiver supports connection via the I2C bus, offering 8 pin holes for soldered connections (the module does not ship with pins included). The receiver also FM radio bands 76 to 108 MHz, RDS data, volume control, automatic frequency/gain control, and 2.7 volts to 5.0 volts of supply voltage. A headphone jack is provided for audio out or external antenna connection [25].

4.2 Velleman MM100

The Velleman MM100 is a FM radio receiver chip manufactured by Velleman and distributed by Jameco Electronics. The MM100 can be purchased from jameco.com for $9.95 USD retail. The Velleman MM100 claims worldwide FM support and has a small form factor for easy integration into computer modules. The MM100 chip is designed to support the I2C bus connection method at 3.3 volts and does not offer pins for jumper wire connection. The receiver features support for RDS system data, volume control, automatic gain control, automatic frequency control, and analog output. Additionally, Velleman does not currently offer a data sheet for the MM100 receiver module [26].

4.3 Grove I2C FM Receiver

The Grove I2C FM receiver module is developed by Seeed Studio based on the RDA5807M FM receiver chip, and can be purchased from seeedstudio.com for $8.80 USD retail [27]. The Grove FM receiver module, as per its namesake, supports the I2C connection method at either 3.3 volts or 5.0 volts. A dedicated 4-pin I2C bus connector is present on the receiver board, allowing it to be connect to other systems via jumper wires. The module is capable of supporting FM frequencies of 50 to 115 MHz which includes all U.S. FM station bands, as well as RDS data. The Grove receiver features digital auto gain control, lower power consumption, and a headphone jack for antennas or audio out. Seeed Studio provides documentation for the Grove I2C through its wiki [28].

4.4 Discussion

In terms of FM support, all three receivers meet requirements. The Grove I2C receiver supports the widest range of FM bands of 50 to 115 MHz in comparison to the SparkFun receiver, which only supports 76 to 108 MHz (the Velleman does not indicate its FM band range, only claiming ”worldwide FM support”). The three receivers also use the I2C bus as a connection method, though each differs slightly. The SparkFun and Velleman receivers have empty pin holes for each of the I2C connector pins, requiring us to solder wires onto the receiver module in order to connect the receiver to our Axolotl system. On the other hand, the Grove I2C receiver already has pins built into those slots, allowing for the use of a 4-pin I2C connector to connect the module to the Axolotl without extra soldering. The supply voltage each receiver supports is different. The SparkFun receiver is the most flexible as it accepts a supply voltage of anything between 2.7 and 5.0 volts; the Grove receiver can only accept a supply voltage of 3.3 or 5.0 volts and the Velleman can only accept a supply voltage of 3.3 volts. The Grove I2C is also the least expensive option of the three, priced at $8.80 versus $9.95 for the Velleman receiver and $19.95 for the SparkFun receiver.

4.5 Conclusion

We chose the Grove I2C FM receiver module for our project as it is the least expensive and can be installed via an I2C connector without soldering. Both the Velleman and SparkFun receivers require soldering wires into the pin holes on the receiver module and then connecting those to our NVIDIA Jetson TX2 module via jumpers; hand soldering wires onto the receiver in this manner may affect its reliability and/or damage the hardware if not done correctly. With the Grove receiver, we can use a ready-made 4-pin jumper wire that can be plugged into the Grove receiver's I2C acceptor and then attached to the correct I2C expansion header pins on our TX2 [30]. As we are utilizing a software-defined radio to provide FM radio functionality, the additional features afforded by the SparkFun or Velleman receivers are unnecessary and do not warrant the extra cost.

REFERENCES

[1] ”Open Source Routing Machine”, Open Source Routing Machine - OpenStreetMap Wiki. [Online]. Available: http://wiki.openstreetmap.org/wiki/Open_Source_Routing_Machine. [Accessed: 14-Nov-2017].

[2] ”Project OSRM”, Project-osrm.org, 2017. [Online]. Available: http://project-osrm.org/. [Accessed: 14-Nov-2017].

[3] ”OSRM API Documentation”, Project-osrm.org, 2017. [Online]. Available: http://project-osrm.org/docs/v5.10.0/api/#general-options. [Accessed: 14-Nov-2017].

[4] ”Navit’s Wiki”, Wiki.navit-project.org, 2017. [Online]. Available: https://wiki.navit-project.org/index.php/Main_Page. [Accessed: 14-Nov-2017].

[5] ”Navit - OpenStreetMap Wiki”, Wiki.openstreetmap.org, 2017. [Online]. Available: http://wiki.openstreetmap.org/wiki/Navit. [Accessed: 14-Nov-2017].

[6] ”Navit - Car navigation system”, Navit-project.org, 2017. [Online]. Available: http://www.navit-project.org/. [Accessed: 14-Nov-2017].

[7] ”navit-gps/navit”, GitHub, 2017. [Online]. Available: https://github.com/navit-gps/navit. [Accessed: 14-Nov-2017].

[8] ”libosmscout - OpenStreetMap Wiki”, Wiki.openstreetmap.org, 2017. [Online]. Available: http://wiki.openstreetmap.org/wiki/Libosmscout. [Accessed: 14-Nov-2017].

[9] T. Teulings, ”libosmscout”, Libosmscout.sourceforge.net, 2017. [Online]. Available: http://libosmscout.sourceforge.net/. [Accessed: 14-Nov-2017].

[10] ”The 2-Clause BSD License — Open Source Initiative”, Opensource.org, 2017. [Online]. Available: https://opensource.org/licenses/BSD-2-Clause. [Accessed: 14-Nov-2017].

[11] ”GNU General Public License v2.0 - GNU Project - Free Software Foundation”, Gnu.org, 2017. [Online]. Available: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html. [Accessed: 14-Nov-2017].

[12] ”GNU Lesser General Public License v3.0- GNU Project - Free Software Foundation”, Gnu.org, 2017. [Online]. Available: https://www.gnu.org/licenses/lgpl-3.0.en.html. [Accessed: 14-Nov-2017].

[13] ”Android Bluetooth Wireless OBDII Reader & Scan Tool - For Android Devices Only”, BAFX Products, 2017. [Online]. Available: https://bafxpro.com/products/obdreader. [Accessed: 13-Nov-2017].

[14] ”OBDLink MX Bluetooth”, OBDLink — OBD Solutions, 2017. [Online]. Available: http://www.obdlink.com/mxbt/. [Accessed: 20-Nov-2017].

[15] ”OBDLink MX Bluetooth Scan Tool”, ScanTool.net LLC, 2017. [Online]. Available: https://www.scantool.net/obdlink-mxbt/. [Accessed: 20-Nov-2017].

[16] ”OBD - Elm Electronics”, Elmelectronics.com, 2017. [Online]. Available: https://www.elmelectronics.com/products/ics/obd/?v=7516fd43adaa. [Accessed: 20-Nov-2017].

[17] ”OBDLink MX WiFi”, OBDLink — OBD Solutions, 2017. [Online]. Available: http://www.obdlink.com/mxwf/. [Accessed: 20-Nov-2017].

[18] ”Bluetooth Basics - learn.sparkfun.com”, Learn.sparkfun.com, 2017. [Online]. Available: https://learn.sparkfun.com/tutorials/bluetooth-basics/how-bluetooth-works. [Accessed: 13-Nov-2017].

[19] ”Perl OBD-II Logger”, SourceForge, 2017. [Online]. Available: https://sourceforge.net/projects/pobd-logger/. [Accessed: 14-Nov-2017].

[20] ”OBD GPS Logger for Linux and OSX”, Icculus.org, 2017. [Online]. Available: http://icculus.org/obdgpslogger/. [Accessed: 14-Nov-2017].

[21] ”pyOBD - Open-source OBD-II diagnostics”, Obdtester.com, 2017. [Online]. Available: http://www.obdtester.com/pyobd. [Accessed: 14-Nov-2017].

[22] ”Artistic License 2.0 — Open Source Initiative”, Opensource.org, 2017. [Online]. Available: https://opensource.org/licenses/Artistic-2.0. [Accessed: 14-Nov-2017].

[23] ”GPSd - Put your GPS on the net!”, Catb.org, 2017. [Online]. Available: http://www.catb.org/gpsd/. [Accessed: 14-Nov-2017].

[24] ”SparkFun FM Tuner Evaluation Board - Si4703 - WRL-12938 - SparkFun Electronics”, Sparkfun.com, 2017. [Online]. Available: https://www.sparkfun.com/products/12938. [Accessed: 21-Nov-2017].

[25] ”Si4702/03-C19: BROADCAST FM RADIO TUNER FOR PORTABLE APPLICATIONS”, Cdn.sparkfun.com, 2017. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/2/7/4/Si4703_datasheet.pdf. [Accessed: 21-Nov-2017].

[26] ”MM100: Velleman: FM Radio and RDS Mini Receiver Module : Electronic Kits & Projects”, Jameco.com, 2017. [Online]. Available: https://www.jameco.com/z/MM100-Velleman-FM-Radio-and-RDS-Mini-Receiver-Module_2263359.html. [Accessed: 19-Nov-2017].

[27] ”Grove - I2C FM Receiver”, Seeedstudio.com, 2017. [Online]. Available: https://www.seeedstudio.com/Grove-I2C-FM-Receiver-p-1953.html. [Accessed: 21-Nov-2017].

[28] ”Grove - I2C FM Receiver - Seeed Wiki”, Wiki.seeed.cc, 2017. [Online]. Available: http://wiki.seeed.cc/Grove-I2C_FM_Receiver/. [Accessed: 21-Nov-2017].

[29] ”Map of radio stations in North America ? World Radio Map”, Worldradiomap.com, 2017. [Online]. Available: http://worldradiomap.com/map/north-america. [Accessed: 21-Nov-2017].

[30] ”NVIDIA Jetson TX2 J21 Header Pinout - JetsonHacks”, JetsonHacks, 2017. [Online]. Available: http://www.jetsonhacks.com/nvidia-jetson-tx2-j21-header-pinout/. [Accessed: 21-Nov-2017].