Auto-Logger

Group 13

Zachary Ross – EE

Hassan Siddiqui – EE

Justin Wright – EE

Sponsors

Florida Solar Energy Center

Table of Contents

Executive Summary

The University of Central Florida is now the biggest school in our nation with over 60,000 undergraduate students enrolled. As the population of UCF continues to grow at an outstanding rate, the university must improve upon their parking infrastructure and consider switching from conventionally powered vehicles to renewable resources. In order to better understand the driving habits and parking needs of their students and faculty, UCF, in conjunction with the Florida Solar Energy Center (FSEC), have solicited for a Senior Design group to design and fabricate a device that will collect this information for their use.

Senior Design group #13, consisting of three Electrical Engineering students Hassan Siddiqui, Justin Wright and Zachary Ross, set forth to satisfy the needs of this sponsored solicitation. Their solution, titled The Auto-Logger, will autonomously interface with any internal combustion engine based vehicle as well as a single model of Electric Vehicle, track vehicle localization through GPS and time taken for the vehicle operator to complete various actions as well as some parameters associated with the vehicle ambient temperature and transportation speed. A modified microcontroller will process the information harvested by its constituent components before polling for an internet connection and transmitting the formatted data to a MySQL database hosted and owned by the FSEC in Cocoa Beach, FL. The Auto-Logger will be conservative in design by customizing easily accessible microcontroller technology and utilizing readily available internet accessibility to achieve the lowest power consumption, minimize device package volume and dramatically reduce associated operational costs.

The Auto-Logger can be simplified down to three subsystems: Data Acquisition, Component Integration, and Server Communications. All three previously mentioned subsystems will be seamlessly integrated to collect, manage and transmit characteristic transportation data provided by the UCF community for later analysis by the University of Central Florida and the Florida Solar Energy Center. This team project will be challenging but also rewarding and will allow for each member to make use of the skills attained as an undergraduate while gaining experience on the fundamentals required for a career in engineering.

Project Description

Motivation & Objectives

As the war between groundbreaking electric or hybrid engines and conventional combustion engines wages on, the health of our planet continues to deteriorate at an accelerating rate. In an effort to turn the tables in favor of electric vehicles Group 13 has decided to fabricate a device that will actively monitor specific transportation and characteristic drive information in order to show, plain and simple, the numerous advantages of switching from vehicles with internal combustion engines (ICE) to those with electric or hybrid engines. In addition to providing strong supporting evidence to push the renewable agenda, we can also use such a device to gather data, which can be extrapolated upon by the FSEC and UCF, to conduct research on the efficiency of UCF's parking infrastructure. This could include both standard parking and EV outlet parking for electric vehicles.

Motivation for this senior design project stems directly from our time attending UCF, we experience first-hand how important additional resources and proper lot planning are to the improvement of the UCF community. In an effort to exercise our green thumb as Electrical Engineering undergraduates, we take pride in pushing the political agenda for electric or hybridized vehicles due to their ability to greatly reduce CO2 emissions, improving the air quality of the East Orlando region. Group 13 will be working hand-in-hand with UCF as they, along with the FSEC, will be utilizing the data collected by the Auto-Logger. The final goal will be to create a device which automatically logs, manages and uploads desired information to a MySQL database hosted and managed by UCF's Florida Solar Energy Center in Cocoa Beach, Florida.

The Auto-Logger is an autonomous logging device which will collect a robust amount of information in order to meet the University's and the FSEC's needs. In order to adequately portray the advantages of electric vehicles, data on fuel combustion rates, engine stress, and oil levels could be collected on ICE vehicles. Additionally, the Auto-Logger has the capability to collect information on Battery health, charge capacity, atmospheric temperature and many others when interfacing with electric vehicles. With this information the FSEC can make a strong comparison between combustion engines and electric or even hybrid engines and their impact on both your wallet and the environment. To address parking infrastructure on the UCF campus the Auto-Logger will log information pertaining to the vehicle's location with an integrated GPS module, overall time spent on campus including time parked using the onboard clock associated with our chosen microcontroller, and even duration of time spent accessing one of the EV charge stations if integrated onto electric vehicles.

The Auto-Logger will incorporate customized microcontroller design and fabrication in an attempt to minimize housing space required within the vehicle. We expect the Auto-Logger to reside in a small and discreet package, enough to fit within the vehicle without imposing on the driver's legroom. This microcontroller will interface directly with the vehicle through the standardized OBD-II interface port, which resides in the cabin of the vehicle. To drastically reduce the cost of the Auto-Logger, Groups 13 plans to make use of the UCF Wi-Fi network to upload all the data that is collected to MySQL database hosted by the FSEC in Cocoa Beach as opposed to charging the drivers additional fees, or including them in our own budget, for cellular data we might use.

Requirements and Specifications

Through the development of this project, numerous requirements and associated specifications have been imposed on the Auto-Logger design. Standards from vehicle manufacturers, such as SAE J1962, mandate how our device will communicate and receive power from various makes/models of vehicles. Our device is limited to 12 volts set forth by the SAE J1962 port standard, mandating that pin 16 supplies 12 volts from the vehicle. Designing around certain requirements impose additional specification to the device such as vehicle age restrictions, device size and vehicle operation frequency. Requirements and Specifications for the Auto-Logger are defined below in Table 1 - System Specifications and Table 2 - Data Channel Specifications:

Table 1 - System Specifications

System	Constraint
Available Power	12V DC provided by OBD-II pin 16
Vehicle Compatibility	Internal Combustion Engines: - Post-1996 makes/models - CAN and SAE J1962 compliant makes/models Electric Vehicles: - FSEC provided Nissan LEAF
Incidental User Error	Protective, beveled Casing Non-invasive vehicle mounting No stray wiring
Device Size	Length: 20cm Width: 10cm Height: 10cm *Excluding antenna length
Signal Strength Threshold	-80 dBm *Average signal strength measured over 30 second interval
Vehicle Operation	Operational at least 4 days per week Provides transportation to UCF at least twice per week
Temperature Durability	Withstand fluctuations up to 50° Celsius over 1 hour
RF Interference	GPS and Wi-Fi shouldn't operate at the same time if possible

Table 2 - Data Channel Specifications

Data Channel	Data Acquired
OBD-II Interface (ICE)	Velocity Engine Load Ambient Temperature
OBD-II Interface (EV)	Velocity Battery Condition Ambient Temperature Battery Charge Level EV Charge Duration
GPS	Location Distance Traveled
Temperature Sensor	Cabin Temperature
Wi-Fi Module	mySQL Database communication

Related Standards

IPC/JEDEC J-STD-020

During the solder reflow process a device is subject to the extreme temperatures associated with this process, as a result the vapor pressure of moisture inside a non-hermetic package increases greatly. Under certain conditions, this pressure can lead to device damages listed below:

- Internal delamination of the packaging materials from the die and/or lead frame/substrate.
- Internal cracks that do not extend to the outside of the package, bond damage, wire necking, bond lifting, die lifting, thin film cracking, or cratering beneath the bonds.

In the most severe cases, the stress can result in external package cracks. The purpose of this standard is to identify the classification level of non-hermetic solid state Surface Mount Devices (SMDs) that are sensitive to moisture-induced stress so that they can be properly packaged, stored, and handled to avoid subsequent thermal/mechanical damage during the assembly solder reflow attachment process and/or repair operation.

RS-232 Transfer Protocol

The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pinout of connectors. It formally defines the signals traveling between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data communication equipment), such as a Wi-Fi modem. The current version of the standard was issued in 1997 and is depicted below.

Table 3 - RS-232 Signals with Associated Pin Assignments on DB-25 Connectors

Signal			**Origin**		DB-25**pin**
Name	**Typical purpose**	**Abbreviation**	**DTE**	**DCE**	
Data Terminal Ready	DTE is ready to receive, initiate, or continue a call.	DTR	●		20
Data Carrier Detect	DCE is receiving a carrier from a remote DCE.	DCD		●	8
Data Set Ready	DCE is ready to receive commands or data.	DSR		●	6
Ring Indicator	DCE has detected an incoming ring signal on the telephone line.	RI		●	22
Request To Send	DTE requests the DCE prepare to transmit data.	RTS	●		4
Ready To Receive	DTE is ready to receive data from DCE. If in use, RTS is assumed to be always asserted.	RTR	●		4
Clear To Send	DCE is ready to accept data from the DTE.	CTS		●	5
Transmitted Data	Carries data from DTE to DCE.	TxD	●		2
Received Data	Carries data from DCE to DTE.	RxD		●	3
Common Ground	Zero voltage reference for all of the above.	GND	common		7
Protective Ground	Connected to chassis ground.	PG	common		1

TIA/EIA-422

This technical standard, similar to RS232 but with differential signaling, was developed by the Electronic Industries Alliance and specifies electrical characteristics of the balanced voltage digital interface circuit. TIA/EIA-422 supports cases identical to RS232, interchange of serial binary signals between DTEs and DCEs, but extends to incorporate any point-to-point interconnection of serial binary signals between digital equipment. The standard only defines signal levels; other properties of a serial interface, such as electrical connectors and pin wiring, are set by other standards.

Table 4 - RS-232 Standard Specifications

Standard	TIA/EIA-422
Physical Media	Twisted Pair
Network Topology	Point-to-point, Multi-dropped
Maximum Devices	10 (1 driver & 10 receivers)
Maximum Distance	1500 meters (4,900 ft.)
Mode of Operation	Differential
Maximum Binary Rate	100 Kb/s – 10 Mb/s
Voltage Levels	–6V to +6V (maximum differential Voltage)
Available Signals	Tx+, Tx-, Rx+, Rx- (Full Duplex)

RoHS Compliance

RoHS stands for Restriction of Hazardous Substances. RoHS, also known as Directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment, originated in the European Union and restricts the use of six hazardous materials found in electrical and electronic products. All applicable products in the market after July 1, 2006 must pass RoHS compliance. RoHS impacts the entire electronics industry and many electrical products as well.

ASME Y14.5

The ASME Y14.5 standard is intended for design, drafting, mechanical, manufacturing, production, tool/gage, quality, process and project engineers, CAD/CAM/CAE specialists, inspectors and educators and is internationally applied in the manufacturing industry. This standard is considered the guideline for the design and manufacture of geometric dimensioning and tolerencing. This works to establish uniform practices in the industry to stating and interpreting requirements for use in on engineering drawing and in all related documents.

Wi-Fi Communications:

Must comply with FCC and IEEE wireless transmission standards so the device doesn't cause interference with other devices or cause a safety hazard to people and animals near the transmission device.

802.11 – Pertains to wireless Local Area Networks (LANs) and provides 1 or 2 Mb/s transmission in the 2.4GHz band through implementation of either frequency-hopping spread spectrum (FHSS) or direct-sequence spread spectrum (DSSS).

802.11a – An extension to the original 802.11 standard that pertains to wireless LANs and goes as fast as 54 Mb/s in the 5GHz band. 802.11a employs the orthogonal frequency division multiplexing (OFDM) encoding scheme as opposed to FHSS and DSSS.

802.11b – The 802.11b High Rate Wi-Fi is an extension to the original 802.11 and secondary 802.11a, that also pertains to wireless LANs but yields a connection as fast as 11 Mbps transmission in the 2.4GHz band. The 802.11b specification uses only DSSS. Note that 802.11b was actually an amendment to the original 802.11 standard added in 1999 to permit wireless functionality to be analogous to hard-wired Ethernet connections.

802.11g – Pertains to wireless LANs and provides 20+ Mbps in the 2.4-GHz band.

Table 5 - Technical Illustration of Wi-Fi Standards

Feature	WiFi (802.11b)	WiFi (802.11a/g)
Primary Application	Wireless LAN	Wireless LAN
Frequency Band	2.4 GHz ISM	2.4 GHz ISM (g) 5 GHz U-NII (a)
Channel Bandwidth	25 MHz	20 MHz
Half/Full Duplex	Half	Half
Radio Technology	Direct Sequence Spread Spectrum	OFDM (64-channels)
Bandwidth	≤ 0.44 bps/Hz	≤ 2.7 bps/Hz
Modulation	QPSK	BPSK, QPSK, 16-, 64-QAM
FEC	None	Convolutional Code
Encryption	Optional- RC4m (AES in 802.11i)	Optional- RC4 (AES in 802.11i)
Mobility	In development	In development
Mesh	Vendor Proprietary	Vendor Proprietary
Access Protocol	CSMA/CA	CSMA/CA

Global Positioning System (GPS)

GPS module interfacing and communication structures must comply with National Marine Electronics Association (NMEA) standards and regulations. NMEA 2000 Edition 3.101 standards define the interface between various pieces of marine electronic equipment. The standard permits marine electronics to send information to computers and to other marine equipment.

NMEA - The protocol has changed and the number and types of sentences may be different depending on the revision. Most GPS receivers understand the NMEA standard 0183 version 2. This standard dictates a transfer rate of 4800 b/s as well as a standard set of output sentences which are depicted in Table 29 - NMEA Standardized GPS Output Sentences.

NMEA OneNet - Network interface standard based on the IEEE 802.3 Ethernet Standard and provides transport of standard NMEA 2000 messages using Standard Ethernet protocol in a common non-proprietary format.

Universal Asynchronous Receiver/Transmitter (UART)
UART is a microchip device or component that translates data between parallel and serial interfaces. UARTs are commonly implemented in conjunction with communication standards such as RS-232, RS-422 or RS-485. This will be used as a reference when translating data between the OBD-II interface protocols depicted below in Table 6 - OBD-II Associated Standards and the microcontroller for efficient communication.

Table 6 - OBD-II Associated Standards

Standard Number	Scope	Title
SAE J1699-3-2015	OBD-II	Vehicle OBD II Compliance Test Cases: The main purpose of this Recommended Practice is to verify that vehicles are capable of communicating a minimum subset of information, in accordance with the diagnostic test services specified in SAE J1979: E/E Diagnostic Test Modes, or the equivalent document ISO 15031-5: Communication Between Vehicle and External Equipment for Emissions-Related Diagnostics Part 5: Emissions-related diagnostic services.
ISO 11898 - Road Vehicles Controller Area Network (CAN) Package	OBD-II	ISO 11898 - Road Vehicles Controller Area Network (CAN) Package addresses the data link layer, signaling, high speed access unit, time triggered communication and low speed, fault tolerant, medium dependent interface of the control area network within road vehicles.
ISO 9141-2:1994	OBD-II	Is limited to vehicles with nominal 12 V supply voltage. Describes a subset of ISO 9141:1989. Specifies the requirements for setting-up the interchange of digital information between on-board emission-related electronic control units of road vehicles and the SAE OBD II scan tool as specified in SAE J1978.
SAE J2178-1-2011	OBD-II	This SAE Recommended Practice recommends test methods, test procedures, and specific test parameters to help verify that vehicles and test tools can communicate using the SAE J1850. This document only verifies the portion of SAE J1850 that is used for OBD-II communications.
SAE J1962-2015	OBD-II	This document is intended to satisfy the requirements of an OBD connector as required by U.S. On-Board Diagnostic (OBD) regulations. The diagnostic connection specified in this document consists of two mating connectors, the vehicle connector and the external test equipment.

For fire safety in the PCB the board has to use standard UL 796 which requires the use of a UL 94 flammability test. This test will be performed by the manufacturer prior to shipment.

Design Constraints

Economic Constraints

The economic constraints, defined both by the wants and needs presented by the FSEC and the miniscule net worth of the design team, are to reduce the fabrication and operational costs associated with the Auto-Logger design and implementation. The FSEC strongly expressed their desire to reduce or ultimately eliminate costs incurred through data transmission. Economic constraints imposed on the Auto-Logger include, but are not limited to:

- Avoid charges from data transfer through mobile carrier networks such as AT&T and T-Mobile.
- Hardware costs including PCB Fabrication, hardware acquisition as well as testing tools and materials
- Avoid costs incurred through vehicle testing (renting electric car)
- Protect against driver interference cause by auto-logger malfunction

Environmental Constraints

The environmental constraints for the Auto-Logger arise not from the device itself but through its intended application and utilization of the resources it creates. The FSEC plans to use the Auto-Logger to study the parking habits and future requirements of the UCF community as well as investigate operational differences between electric and internal combustion engine vehicles. The study could lead to environmental effects through the depiction that it is actually more cost effective for individual drivers to switch from a gas powered car to electric. Each person who could switch from a gas powered car to electric would have great environmental ramifications that could decrease emissions into the air.

Similarly, parking lots require a large and ultimately level plot of land, but actually only experience usage about 50% of the time. If the FSEC were able to discern a more intelligent manner of creating parking infrastructure for the UCF community, the environmental impacts would be noticeable as we move forward into the future. Especially with the rate at which the UCF community continues to grow.

Social Constraints

The social impact of the Auto-Logger is to encourage daily drivers to switch from gas cars to an electric cars by showing those drivers data of their daily driving habits. Through those driving habits FSEC can assess how much money a driver can save by switching to an electric car. Showing drivers how much money could be saved by switching to electric vehicles could ignite social change if enough drivers switch.

Political Constraints

The FSEC is a state government affiliate program which has solicited this request for an automotive transportation data logger, because the FSEC is a government associated entity there are additional constraints which the design team must adhere to. The FSEC will provide the design team with access to a test vehicle, the LEAF is an electric vehicle manufactured by Nissan. The design team is not allowed to operate this vehicle, under any circumstance, not only because it is the property of the FSEC (a government affiliated center) but also due to liability issues. If one of the members of the design team were a UCF employee or a contractor for the state, the FSEC would have the ability to obtain clearance for one of the members to operate the Nissan LEAF provided without the supervision of an FSEC employee. However, since all members of the team are undergraduate students, the FSEC requires one of their own employees oversee all work performed take responsibility for all operation of the vehicle.

There are possible political ramifications of the Auto-Logger because the results obtained by the FSEC study could show that, without question, electric vehicles save money on gas and reduce environmental pollution. As a byproduct of the research conducted by FSEC through this study, the state government could take action to encourage people to switch to electric vehicles to save on gas and help the environment, thus lowering CO2 emissions into the air, reducing pollutants that are sent into the atmosphere all while saving money.

The Auto-Logger will never attempt to breach personal security, as the auto-logger will notify the user before any action it takes via the LCD screen. The Fifth Amendment states “nor shall private property be taken for public use, without just compensation” protects the privacy of personal information.

Ethical Constraints

The Auto-Logger introduces numerous ethical constraints due to the type of information it will be collecting and the inherent lack of input required by the user for proper operation. The device will characterize and log personalized transportation data as well as active localization of participant’s vehicle. It would be unethical if the volunteers set to participate in such a study were not informed that their location and driving habits would be actively monitored and stored through the duration of the study. The participants need to give their permission in writing when accepting the terms and conditions required to participate in the study and place the Auto-Logger in their vehicle. The terms and conditions which apply to this project include, but are not limited to:

- Permission to log and study characteristic drive habits

- Access to vehicular transportation data
- Permission to share data with 3rd party affiliates

Health and Safety Constraints

The spatial needs of the volunteers participating in this study have to be taken into account as not to hinder the driver's ability to access or operate the vehicle. The device needs to be small enough such that the driver can, in no way, make contact with the device while entering, exiting or operating the vehicle. OBD-II ports are located in different areas of the vehicle, but are generally placed around the steering wheel. This positioning introduces a constraint dependent upon the participants' vehicle, as the placement of this interface port varies between vehicle manufacturers.

The OBD-II port is located in various locations depending on the make and model of the vehicle. OBD-II ports can be found to the left or right of the steering wheel or even centered beneath it. The port can face towards the driver or down towards the floor of the vehicle. Downward facing ports introduce the most restriction into the project design because the device cannot be any longer than two inches or risk physical interference with/from the driver's legs. Ports that are located on the left side of the steering column introduce an obstacle while entering and exiting the vehicle. When either entering or exiting the vehicle, with the OBD-II port in this orientation, the device could make full contact with the driver. Even if contact is minimized, the location could eventually lead to problems for device functionality through repeated contact.

On various makes and models the OBD-II port is located beneath the dash, which is lower than most locations. Since the placement of the interface port so low, once again, the size of the device must be restricted as to prevent physical contact with the driver's legs. The device is fragile, even with a protective enclosure, so if the device were to make continuous contact with the driver's legs it would lead to damage either to the device, the vehicle operator or to the OBD-II port itself. The driver should not be aware of the device while operating the vehicle, which increases the risk associated with continuous contact.

To solve these design constraints the design team needed to have something that was universal and could fit in all cars with a low probability of incurred damages. The design team decided that instead of mounting the device directly to the port, a cord could be used to connect the device with the OBD-II port. With the device connected with a cord, the driver is given more freedom to put the device where they feel most comfortable. This will allow the device to be mounted on the dashboard, windshield, or even in the center console (not recommended). The implementation of an extension cord removes the device from the immediate vicinity of the driver, and permits secure placement of the device. This solution prevents hindrance of the driver's ability to operate the pedals because the cord can be relocated or rerouted through the dashboard to appease the driver. The

OBD-II port on the left and right sides of the steering wheel are much more accessible with the addition of a connection extender. If the OBD-II port is on the right hand side and the device is mounted on the left side of the steering wheel (on the dashboard) the device could block gauges that the driver needs to see.

Sustainability Constraints

The sustainability of the Auto-Logger is dependent on the power requirement while in low power mode and the associated Lead Acid battery charge capacity. The sustainability of the Auto-Logger would also depend on how long the components last. Some of the optional Printed Circuit Board (PCB) finishing methods have a low shelf life, which limits the sustainability of any device which utilizes this technique. The device can only function as long as it can connect to the internet via Wi-Fi, to send the data it has collected from the car to the server for collection. Correspondingly, without the regular establishment of a connection to the internet the Auto-Logger is not a functionally sustainable device. Without the Internet connection the device would have to be physically retrieved in order to harvest the data. This is not a desired not is it a sustainability or efficient method for which the FSEC wishes to conduct this study.

Network Accessibility

All data collected from the vehicle by the Auto-Logger must be sent over the internet to a MySQL database hosted by the Florida Solar Energy Center in Cocoa Beach. For the device to send the large repository that has been collected to the FSEC it must be connected to the internet. A Wi-Fi module creates the ability to send data to the database without incurring charges from the use of a participant's cellular network. The drawback is that Wi-Fi signal strength in the parking lots and garages may not always be sufficient enough to support reliable transmission. To counter this it may be required for the participants in the study to drive to certain areas on the UCF campus in order to find an adequate connection to reliably send the required data to the servers.

Bluetooth has not been chosen to be implemented because the FSEC has requested that the user interact with the device as infrequently as possible, besides initially plugging the device into the OBD-II port. Another disadvantage with Bluetooth is that it still requires a network to transfer any data from the participant's phone to the FSEC database. Data charges will incur on the participant, which is not desired. Utilization of the UCF Wi-Fi resolves this issue by alleviating any fees that may incur through data use or overage charges from their associated cellular network provider and also eliminates most, if not all, interaction with the study participant.

Wi-Fi in the parking lots is not reliable in all locations therefore data cannot always be sent due to connectivity issues. To compensate for lack of Wi-Fi connectivity

FSEC can set up Wi-Fi hotspots at the road entrances to the UCF campus to reliably collect their data. FSEC has stated they want to track parking trends so if the participants are restricted to certain lots to get Wi-Fi connectivity will undesirably skew their data.

Overall the network constraint introduced due to the use of UCF Wi-Fi is the limited propagation range to the roads and parking lots where the vehicles will be located. The countermeasures to the limited Wi-Fi range is setting up hotspots that the participants will drive by during their typical commute, introduce a high gain Wi-Fi antenna to boost the Wi-Fi signal received by the Auto-Logger, or even establish a schedule for the participants to successfully transmit the data to the FSEC servers in Cocoa Beach, Florida.

Manufacturer Compatibility

The compatibility of the manufacture to the device varies greatly between makes and models. Prior to 1996 vehicles were not required by law to have an OBD-II port, and because of this, vehicles before 1996 cannot participate in the study. After 2008 all cars sold in the United States have to use Control Area Network (CAN) bus as one of their signal protocols.

The SAE J1962 specification provides two standard hardware interfaces from the manufacturer, a type A female connector or a type B female connector. They both have the same pin setup, but the type B has a groove blocked in the middle. The type A male connector cannot be plugged into a type B female connector. However a type B male connector can fit into both type A and B female connectors.

The OBD-II port has 16 pins and they are not uniformly used across all manufactures prior to 2008. Pin 16 is used as battery voltage, however, type A connectors delivers 12 volts while type B connectors deliver 24 volts. There are 5 different protocols used in OBD-II ports that vary across manufacturers.

The SAE J1850 Pulse Width Modulated (PWM) communication protocol is the standard for the Ford Motor Company. For this OBD-II protocol, pin 5 is the ground and pin 2 is the positive bus and pin 10 is the negative bus. The SAE J1850 Variable Pulse Width Modulation (VPW) protocol uses a single wire approach and is the standard of General Motors. The VPW protocol uses pin 2 for the positive bus and the negative bus idles low. Pin 5 is used for the ground when using VPW. ISO 9141-2 is a protocol that is used by Chrysler. ISO 9141-2 protocol uses pin 7 as the K-line data and an optional L-line at pin 15. The ISO 14230/KWP2000 protocol is used by most European and Asian manufacturers. The ISO 14230/KWP2000 protocol is physically the same as ISO 9141-2 except the message can contain 255 bytes in the data field instead of 12 bytes. The ISO 15765 CAN protocol is used in all vehicles after 2008 and uses pin 6 as CAN high and pin 14 as CAN low. In Table 7 - OBD-II Manufacturer Specifications each protocol is shown with their respective pins and the manufacturers that use them.

All protocols use pin 16 as power with pin 4 as the chassis ground and pin 5 as the signal ground. Fiat, Alfa Romeo, and Lancia use a CAN bus at 50K baud, which is incompatible with OBD-II standards.

Table 7 - OBD-II Manufacturer Specifications

Protocol	Manufacturers	Pin Number
SAE J850 PWM	Ford Motor Company	2,10
SAE J1850 VPW	General Motors	2
ISO 9141-2	Chrysler, European, and Asian Vehicles	7, 15(optional)
ISO 14230/KWP 2000	Alfa Romeo, Audi, BMW, Citroen, Fiat, Honda, Hyundai, Jaguar, Jeep, Kia, Land Rover, Mazda, Mercedes, Mitsubishi, Nissan, Peugeot, Renault, Saab, Skoda, Subaru, Toyota, Vauxhall, Volkswagen, Volvo	7, 15(optional)
ISO 15765 CAN	All cars after 2008	6,14

Research Related to Project Definition

Currently on the market there are various technologies that make use of the standard OBD-II port. These technologies have already iterated through design, testing and final implementation phases, and as such could be utilized as a guideline. Group 13 took note of these technologies when developing their design for the Auto-Logger. Products harnessing the OBD-II interface can be used to obtain diagnostic information on a vehicle, this information can be used, for example, to identify problems the vehicle may be experiencing and expedite repairs. Insurance companies also use OBD-II ports to track driving habits to determine insurance rates. OBD-II ports are also used to track vehicle fleet information to predict repairs and restrict employee driving habits.

The OBD-II diagnostic tool is used to interpret the many codes output by the vehicle’s computer when the check engine light has illuminated. Such a diagnostic tool has to input the make, model, and year of the vehicle to properly decipher the associated codes. The difference between these diagnostic tools and the Auto-Logger is that the diagnostic tools are utilized when the vehicle is not in use to read the output codes associated with the check engine light.

Progressive’s Snapshot

Insurance companies have used OBD-II products before, such as Progressive’s Snapshot. Progressive’s Snapshot is a device that plugs into the vehicle's OBD-II port. With the Snapshot, Progressive has the capability to track the time of day, total miles driven, and number of hard brakes of a vehicle equipped with the device. Progressive uses this information to determine a customer’s insurance rate based on their driving habits. Progressive also has a blinking light on the Snapshot to show that the device is working when it is plugged in. The device can also be setup to send emails to the user if it is unplugged from the vehicle. Since the OBD-II port is in a position that is difficult to see, this feature provides the user with a reliable method to check the functionality of the device.

Freematic’s Data logger

A very similar data logger to the Auto-Logger is Freematic’s vehicle data logger, V3, which uses Bluetooth that can connect to other devices such as phones and tablets to see real time data. Freematic’s device uses a MicroSD card to store data. The device uses a MEMS module as an accelerometer, gyroscope, compass, and temperature sensor. Freematic’s device uses an Arduino which is the same as the Auto-Logger. The difference between the Freematic’s device and the Auto-Logger is that the Auto-Logger will automatically upload any data collected to a specified server or computer.

IOSiX’s OBD-II/CAN v4 Datalogger

Another device that can be used for fleet tracking information for businesses is the IOSiX OBD-II/CAN v4 Datalogger. This device is very close to what we are trying to accomplish, however this device can track more variables than the Auto-Logger. The v4 Datalogger can interface wirelessly with a computer, tablet or phone for real time data acquisition and if setup could autonomously transfer said information. It can use Wi-Fi or Bluetooth as wireless transmission options to communicate with other devices.

Dawn’s OBD Mini Logger

The Dawn OBD Mini Logger is another data logger that can be used on any vehicles after 1996. The DAWN OBD Mini Logger boasts that it is small and can fit in the palm of your hand. Additionally, this device covers all the necessary protocols to read the OBD-II codes. It can be outfitted with a MicroSD card, with up to 4GB of data available, if the user does not want data to be sent autonomously they can collect the OBD Mini Logger and upload the data manually. GPS is also used in the Mini Logger to track the location of the vehicle and the routes driven. The GPS data is stored in a CSV format, they combine the GPS data with the DAWN dashboard so that trips can be monitored through web based displays and geo-fencing. The 802.11 b, g, and n standards are supported so the use of hotspots can be used to transfer data over the internet to any specified host.

There are many different variations of OBD-II logger in the marketplace that achieve the same goals that the design team has set out to accomplish. This shows that our goals are attainable, that our device can be implemented in the real world and used to accomplish the study that FSEC hopes to conduct. Freematic's open source design also provides the design team with a great resource to use as reference when designing the Auto-Logger.

OBD-II Interfacing

On-Board Diagnostics, Second Generation (OBD-II) is a set of standards for implementing a computer based system to control emissions from vehicles. It was first introduced in the United States in 1994, and has been a required feature on all vehicles manufactured in the US since 1996. Other countries, including Canada, parts of the European Union, Japan, Australia, and Brazil have since adopted similar legislation. An overwhelming majority of modern vehicles support OBD-II or one of its regional flavors.

Among other things, OBD-II requires that each compliant vehicle be equipped with a standard diagnostic connector (DLC) and describes a standard way of communicating with the vehicle's computer, also known as the ECU (Electronic Control Unit). A wealth of information can be obtained by tapping into the OBD-II bus, including the status of the malfunction indicator light (MIL), diagnostic trouble codes (DTCs), inspection and maintenance (I/M) information, freeze frames, Vehicle Identification Number (VIN), hundreds of parameters updated in real-time, and more.

UART

The Auto-Logger will provide a serial interface using the ELM327 command set which supports all the major OBD-II standards including CAN and JBUS. STN1110 is an OBD-II to UART interpreter that can be used to convert messages between any of the OBD-II protocols currently in use and UART. This interpreter is fully compatible with the de facto industry standard ELM327 command set. Based on a 16-bit processor core, the STN1110 offers a more robust feature set and provides better performance than any other ELM327 compatible integrated circuit.

Interoperability

Because the Auto-Logger will be used on vehicles owned by students and faculty from UCF, the device must be compatible with vehicles of various makes and models. Our limitation will be that the OBD-II standard was only mandated on cars made on or after the year 1996 and because of that cars made before that year cannot use our device. All cars made before the year 2008 will be using any of the following communication protocols: ISO 15765-4 (CAN), ISO 14230-4 (Keyword Protocol 2000), ISO 9141-2 (European, Chrysler and most Asian vehicles), SAE J1850 VPW (GM vehicles), SAE J1850 PWM (Ford vehicles). After the year 2008 all vehicles were required to comply with the standardized CAN protocol, including the new electric motor vehicles developed by various automobile companies.

Power Routing

All vehicles that comply with the OBD-II standard must supply 12 volts through pin number 16 of the vehicle's male OBD-II port. The Auto-Logger will utilize this pin for power during regular vehicle operation as well as immediately after the vehicle is shut down. The Auto-Logger will be optimized to minimize its power requirements when the vehicle engine is shut down as not to deplete the battery before its next operation.

Power Supply & Regulation

Battery Types

The use of batteries is needed for the Auto-Logger because it can’t draw power from the car battery while the car is off. If the device is allowed to drain the car battery it could potentially deplete the charge contained preventing the car from starting. To counter this we will use a battery to keep the device active while the car is off so data is not lost. The design team has looked into several different batteries to fulfill the requirements. Possibilities include alkaline, lithium-ion, and nickel cadmium batteries. The alkaline batteries were too large and not rechargeable, which is problematic. The Auto-Logger requires a small battery that can fit into the device and is rechargeable. The nickel cadmium batteries were also too large to fit and did not deliver enough voltage. The lithium-ion battery was the only battery that fit all the requirements including size and power delivery.

The battery that was chosen was the CR2032 lithium-ion battery. The battery is small and round like a coin. Energizer CR2032 is the specific battery that the team has decided on using, characteristic information can been seen below in Table 8 - Characteristics of CR2032. The nominal voltage supplied by the battery is 3.0 Volts. The diameter of the battery is 20 mm and the height is 3.2mm. The weight is 3.2g. The usable temperature of the battery ranges from -20° to 70° Celsius. The usable temperature falls within an acceptable range for storage in a vehicle.

Figure 1 - Discharge Characteristics describes the discharge time under different loads.

Table 8 - Characteristics of CR2032

Characteristic	Value
Supply Voltage	3.0V
Diameter	20mm
Height	3.2mm
Temperature Range	-20℃ to 70℃
Weight	3.2g

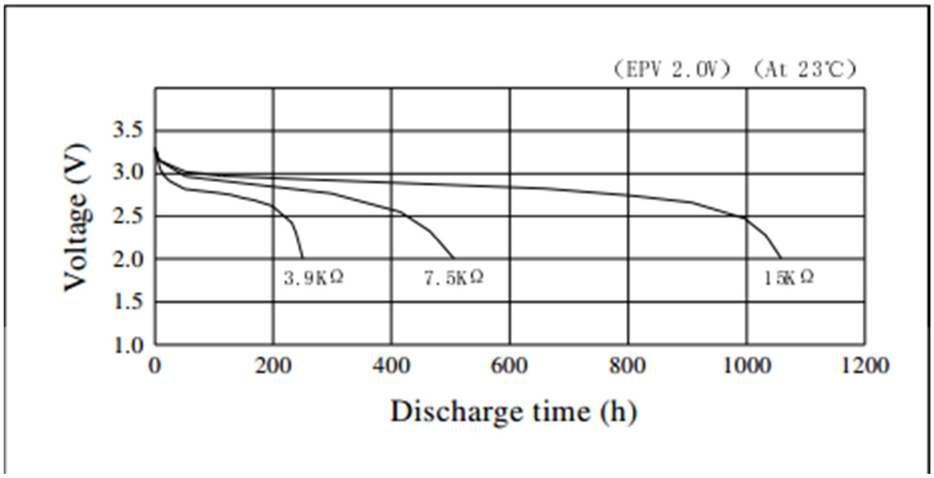


Figure 1 - Discharge Characteristics

Charge Capacity

The nominal charge capacity from the datasheet shows 210 milliamperes per hour while at 20° Celsius under a 15kΩ load. In Figure 2 - Load vs. Capacity, load vs. capacity is shown at different temperatures which fall within the usable temperature for the interior of a car even when turned off. The battery will only be used to keep the clock on. Under typical conditions the battery can last 700,000 hours, while operating near upper boundary conditions it can last 480,000 hours. The datasheet for the DS1307 crystal clock states that any battery with at least 48 milliamps per hour can power the DS1307 for at least 10 years while operating in low-power-mode.

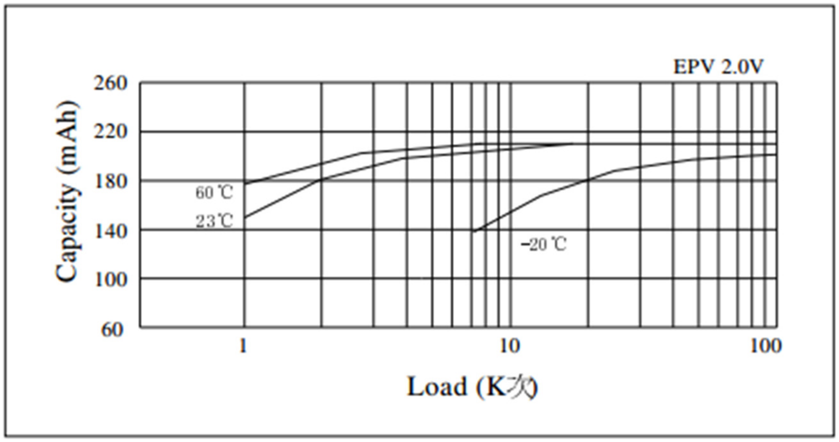


Figure 2 - Load vs. Capacity

Recharging

The CR2032 lithium-ion battery datasheet warns that the device is a primary battery and cannot be charged. The datasheet advises that diodes should be placed facing the load to prevent unexpected charging of the battery. A protective resistance is also needed to safeguard the battery from being charged by large surges and the CR2032 recommended value is at least 2.1k Ω.

Wireless Communications

As one of the most vital aspects of our design, the wireless communication system requires a stable connection to prevent packet loss and various other communication errors. Options for wirelessly transmitted communication protocols include Wi-Fi and Bluetooth.

Bluetooth

Bluetooth is a simple and reliable method of data transmission which could be used to integrate the logging device with the driver's cellular phone. However, due to constraints set by the project sponsor, which states the device should be capable of operating without intervention by the driver other than to plug the device in, data transfer via Bluetooth will not be implemented.

Wi-Fi

The University of Central Florida provides an excellent Wi-Fi infrastructure to keep all of their students and faculty connected at all times. In an attempt to significantly lower operational costs of this project, wireless communication through the UCF

Wi-Fi system is, ideally, the desired solution. Additional research must be conducted to ensure the Wi-Fi infrastructure is capable of propagating to various parking lots throughout the campus without significant degradation in the signal strength. Signal strength ranging from -60dBm to -70dBm is excellent and is desired for data intensive applications such as Voice over IP and video streaming, however a signal strength threshold of -80dBm should be adequate for reliable packet delivery.

Table 9 - Signal Strength Required for Application

Signal Strength	Network Capability
-30 dBm	Max achievable signal strength. The client can only be a few feet from the AP to achieve this. Not typical or desirable in the real world.
-67 dBm	Minimum signal strength for applications that require very reliable, timely packet delivery.(VoIP, Streaming)
-70 dBm	Minimum signal strength for reliable packet delivery.
-80 dBm	Minimum signal strength for basic connectivity. Packet delivery may be unreliable.
-90 dBm	Approaching or drowning in the noise floor. Any functionality is highly unlikely.

Since any vehicle participating in this study will be commuting to and from a UCF campus a Wi-Fi connection may not always be accessible. In order for the Auto-Logger to determine when data transmission should commence a repetitive polling of the desired signal must be performed. The only method for signal polling is by measuring the Received Signal Strength Indication (RSSI). In order to determine the RSSI of a signal the receiving Wi-Fi Network Interference Card (NIC) must measure the power of the signal. Deciphering the results of this measurement depends on the receiving NIC that is implemented. Each NIC has its own specifications for how RSSI measurements are initiated and utilized.

RSSI is defined in the IEEE 802.11 standard as a mechanism by which RF energy is to be measured by the circuitry on a wireless NIC. In this 802.11 standard documentation, RSSI measurement is specified as a numeric value, an integer, with an allowable range of 0-255. The direct relationship between the propagation strength of a Wi-Fi signal and the received power value permits RSSI. According to the standard, the relation between dBm and mW has linear characteristics in regions exceeding those pertinent to our application. Translating signal strength into an RSSI measurement is a metric determined by the NIC manufacturers. As a result, a compare and contrast analysis will be utilized in the selection of an ideal wireless NIC hardware.

ESP8266EX Wi-Fi Module

The ESP8266EX Breakout board manufactured by Espressif Systems is a module that offers a complete and self-contained network solution. This component excels when integrated with any micro controller-based design as it is designed to occupy minimal PCB area. Additionally, the ESP8266EX Breakout demonstrates outstanding features pertinent to the Auto-Logger design including adaptive radio biasing for low-power operation, spur cancellation and fast sleep/wake context switching for energy efficiency.

CC3100MOD Network Processor

The CC3100MOD Network Processor manufactured by Texas Instruments is an FCC, IC, CE and Wi-Fi certified module. This module integrates all protocols for Wi-Fi and internet, greatly reducing host MCU software requirements. The CC3100MOD can connect to any 8-, 16- or 32- bit MCU over the UART interface and also minimizes the host memory footprint requirements for a TCP client application.

Table 10 - Wi-Fi Component Feature Comparison

Wi-Fi Module	CC3100MOD	ESP8266EX
Wi-Fi Protocols	802.11 b/g/n	802.11 b/g/n/d/e/i/k/r
Rx Sensitivity	-94.7 dBm	-98 dBm
Rx Operating Current	53 mA	50 mA
Package Size	20.5mm x 17.5mm	5mm x 5mm
UART Transmission Limit	3.0 Mbps	4.5 Mbps

GPS Localization

A GPS module will be implemented to provide real time localization of the vehicle. This information will be used to determine whether or not the vehicle is currently operating within the bounds of a UCF campus. Active localization of the vehicle is required to identify pertinent transportation characteristics and parking habits as well as identify the need for UCF network acquisition. Because UCF has a number of campuses throughout the Central Florida region, vehicle localization will play a critical role in differentiating parking habits and transportation characteristics between these locations as this may dictate how the FSEC makes use of the acquired information.

Copernicus II Module

The Copernicus II GPS module is a complete 12-channel GPS receiver in a 19mm x 19mm x 2.54mm, thumbnail-sized shielded unit. The small, thin, single-sided module is packaged in tape and reel for pick and place manufacturing processes; 28 reflow-solder-able edge castellation provide interface to your design without costly I/O and RF connectors. Each module is manufactured and factory tested to Trimble's highest quality standards.

FGPMMOPA6B Breakout

The FGPMMOPA6B GPS receiver provides a solution that possesses excellent accuracy in position and speed acquisition, with high sensitivity and tracking capabilities in urban conditions. This GPS chipset, inside the Adafruit GPS Breakout Board, out performs competitors for the same price. The Adafruit Breakout provides the developer with additional functions and features as well as libraries for compatibility with various microcontroller technologies.

Table 11 - GPS Module Feature Comparison

GPS Module	Copernicus II	FGPMMOPA6B Breakout
Dimensions	19mm x 19mm x 2.54mm	16mm x 16mm x 6mm
Tracking Sensitivity	-160 dBm	-165 dBm
Supported Serial Protocols	NMEA 0183, TSIP, TAIP	NMEA
Simultaneous Operation	12-Channels	66-Channels
Update Rate	1 Hz	1 Hz
Accuracy (w/o SBAS)	Within 7.6m	Within 3m
Power Consumption (@ 3V)	132 mW	138.6 mW

LCD Display

Surface Mount Display

To accommodate the setup of the Auto-Logger the design team has decided to implement an LCD screen. The LCD screen is to provide easy visual aid to when selecting the vehicle make and model to run the correct OBD-II protocols on vehicles older than 2008. The design team as limited the choices to an HD44720U and a KS0108B. The HD44720U is a smaller screen at 16 by 2. The KS0108B is larger and easier to read at 64 by 128.

The HD44720U is an LCD screen that runs on 3 to 11 volts that can display one 8 character line or two 8 character lines. The HD44720U has up to 240 fonts while using 9920 bit character generator. The HD44620U meets all the design needs however it is not chosen because a larger screen is needed to be clearer and more visible to the user.

The KS0108B is a graphic LCD with a size of 128 by 64 and a LED backlight. The LCD driving voltage ranges from 8 to 17 volts. The LCD can be powered by the 12V from a vehicle through the OBD-II port. The LCD screen has 100 pins to connect to the Arduino and the power source to correctly display the prompts for setup.

Timing and Storage

Low Power Components

The low power device is a real time clock DS1307. The DS1307 function is to keep track of time while the car is off because that is a parameter FSEC wants Auto-Logger to track. The reason a separate clock is needed is because the Arduino clock shuts off while the car is off. For that reason a separate clock is added the DS1307 supplied by battery power. The DS1307 can keep track of seconds, minutes, hours, and the date with leap year compensation. The clock allows all the data to be time stamped to be able to retroactively track driving habits daily, weekly, or monthly. The DS1307 will be used as the clock even when the device is on unless the power consumption is too high, then the DS1307 will be turned off and the Arduino clock will be used.

The DC electrical characteristics of the DS1307 when the oscillator is on uses 3 volts from the battery and uses typically 300 nanoamperes with a max of 500 nanoamperes. Figure 3 - Supply Current vs. Battery Voltage shows the supply current with the volts to track how much of the battery is consumed while in use. When the device is using V_{cc} power the clock uses 4.5 to 5.5 volts. The SCL Clock can run from 0 to 100 kHz. The clock crystal used is 32.768 kHz and does not require any external resistors or capacitors to operate. The DS1307 can be run in either 12 hour or 24 hour mode in the clock. The time is tracked in 8 bits and 8 address lines. The first address line tracks the seconds, the second line tracks the minutes, and each line tracks the next iteration until line 8 is the year.

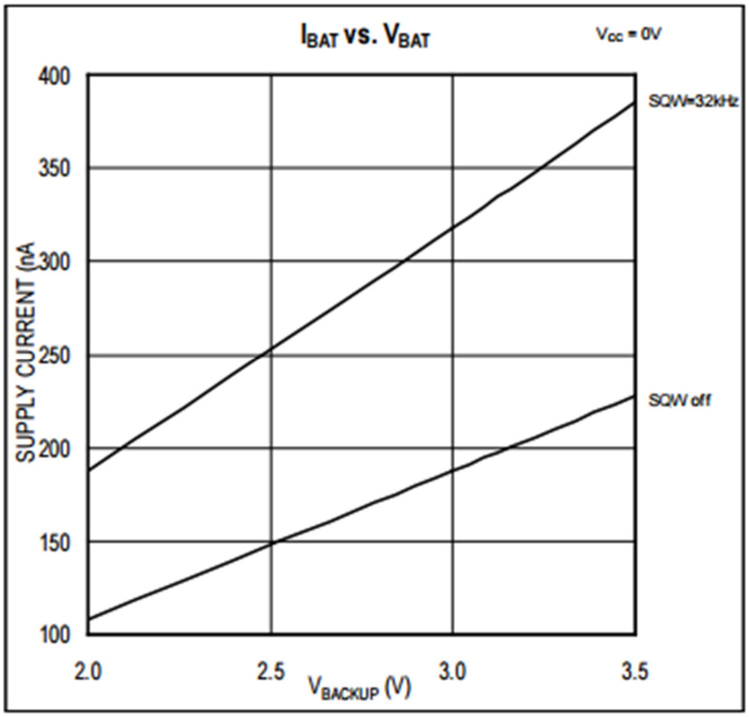


Figure 3 - Supply Current vs. Battery Voltage

SD Card Storage Interface

Storage is needed in the design of the Auto-Logger because the data collected needs to be saved in order to send the data to FSEC. The MicroSD card breakout will be used to interface with the MicroSD for storage. The MicroSD card breakout is compatible with the Arduino board. The breakout board needs 3 or 5 volts to be operational. The breakout has an LED to let the user know that the data is being written and read. There are 6 pins on the breakout board that are needed to successfully link to the Arduino. The MicroSD card used to save data that will be used will be at least 1GB so all data can be sufficiently saved before sending the data to FSEC servers.

Microcontroller

The Auto-Logger requires the ability to control different pieces of hardware and coordinate them all together in order to communicate with a vehicle's diagnostic system, store information, process data, and transmit said data when campus Wi-Fi is available. In order to effectively carry out such actions, a central hub will be implemented to manage and control the various peripheral components. There are many microcontrollers and driver boards available on the open market, however very few are heavily modifiable or customizable. The criteria Group 13 used to decide which microcontroller best applied is as follows:

- Adaptability - The microcontroller needed be easily modifiable in both its onboard software and hardware capabilities. It is supposed to be the main foundation of the device and as such it needs to be capable enough to communicate with a GPS receiver, Wi-Fi NIC, MicroSD memory card, and the OBD-II computer within the vehicle.

- Open Source vs. Proprietary - If the microcontroller was proprietary it would be very difficult to modify the board in order to meet the various design constraints placed on the project. If the microcontroller is open source the schematics, chipsets, and various other onboard components would be meticulously detailed in a manner which would facilitate custom modification without permission of the original creator. Also, if the microcontroller is open source there is a higher probability that other components would be compatible with it due to the larger community following of such open source projects and devices. These benefits lead to a preference in microcontrollers that were open source.

- Resource Consumption - The microcontroller has to consume a small amount of power in order to be viable while remaining inexpensive enough in order to fit our budget. Due to these constraints, we had looked at microcontrollers that were at or below the $50.00 mark that also possessed variable modes of power consumption, such as an idle or low power mode. If the microcontroller did not have the capability to be put into a low power mode, it would run the risk of completely depleting the vehicle's battery while the vehicle was left parked overnight. Assurance that the microcontroller could remain connected to the vehicle without depleting the battery, while also operating as autonomously as possible is a significant design constraint of this project.

ATMega328P

The Arduino Uno is a uniquely capable, open source electronics prototyping platform based on easy-to-use hardware and software. It is based on the ATmega328P and has 14 digital Input/output ports, 6 of which can support PWM outputs. Refer to Table 12 - ATmega328P Features for device information pertinent to the project goals.

Table 12 - ATmega328P Features

Microcontroller	ATmega328P
Operating voltage	5V
Nominal input voltage	7-12V
Digital I/O Ports	14
PWM ports	6
Analog ports	6
DC current per I/O port	20mA
DC current for 3.3V port	50mA
Available flash memory	31.5 KB
Clock speed	16MHz

ATMega2560

The Arduino Mega is an open source electronics prototyping platform based on easy-to-use hardware and software. It is based on the ATmega2560, has 54 digital Input/output ports, 15 of which can support PWM outputs. The main reason this board was preferred is because it has the capability to support 4 UART hardware serial ports. It has 7 times more flash memory available in the chip which provides the ability to write a larger program and have more available RAM at the user's disposal if the capability is needed. The Arduino Mega is a significantly more refined and advanced version of the Arduino Uno, and because the price is only 50% more this will be implemented in the final design. Reference to Table 13 - ATmega2560 Features for device information pertinent to project design.

Table 13 - ATmega2560 Features

Microcontroller	ATmega2560
Operating voltage	5V
Nominal input voltage	7-12V
Digital I/O Ports	54
PWM ports	15
Analog ports	16
DC current per I/O port	20mA
DC current for 3.3V port	50mA
Available flash memory	248 KB
Clock speed	16MHz

Printed Circuit Boards

The printed circuit board surface finish forms a barrier between the bare PCB and the components. The surface finish serves two critical functions, to protect the exposed copper circuitry and to provide a conductive surface when assembling components to the PCB. The factors when considering the different finishes for the Auto-Logger is cost, shelf life, durability, components used, and assembly method. The options for PCB include Hot Air Solder Leveling (HASL), Lead Free HASL, Immersion Tin, Organic Solderability Preservative, Electroless Nickel Immersion Gold (ENIG), and Hard Electrolytic Gold. Each of these options have their own characteristic advantages and disadvantages, however the typical thickness of the finishes are 70 to 200 microinches.

HASL and lead free HASL is the most common method of PCB finishing as well as the cheapest option. The process for the HASL method consists of dipping the board in a molten pot of lead or tin then removing the excess solder with hot air knives. One of the benefits of this method is that the board is exposed to high temperatures when dipped in the molten lead or tin which will reveal any unwanted delamination. HASL would provide a reliable solder joint and shelf life. The disadvantages of HASL is uneven surfaces and it is unsuitable for fine pitch components. HASL standard typically uses tin-lead finish, but HASL lead free uses tin-copper, tin-copper-nickel, or tin-copper-nickel Printed Circuit boards or PCB have several options for housing the chips for the Auto-Logger. The process for HASL can be shown in Table 14 - HASL Process Flow which is typically processed in production panel form.

Table 14 - HASL Process Flow

HASL Process Flow					
Clean	Microetch	Apply Flux	Solder Dip	Air Knife Leveling	Rinsing

Immersion Tin is a lead free option that is a metallic finish that covers the copper circuit board. The advantages for this method is lead free, flat surface, and is reworkable. The disadvantages for Immersion Tin is that it is easy to cause handling damage because the finish is fragile. The process uses a carcinogen, Thiourea, which could cause health risks if exposed. The exposed tin can corrode. Tin whiskers also pose a problem because when tin is used as a finish they can spontaneously grow conductive whiskers which can form electrical paths, affecting the operation of the device. The tin whiskers are very small and can't be seen by the naked eye. The small whiskers could be harmless or could cause short circuits if they grow in between two components. The immersion tin process flow can be shown in Table 15 - Immersion Tin Process Flow which is typically processed in production panel form.

Table 15 - Immersion Tin Process Flow

Immersion Tin Process Flow				
Clean	Microetch	Pre-dip	Apply Tin	Post-dip

Organic Solderability Preservative (OSP) or Entek is another option that is lead free. The OSP process protects the copper from oxidation by applying a thin protective layer of material over the copper with a automated process. The advantages of this method include lead free, flat surface, simple process, and repairable. The disadvantages of OSP is there is no way to measure the thickness, products have a short shelf life of 6-12 months, plated through holes don't work well with this process, and there is always exposed copper on the final assembly. The OSP process flow is shown in Table 16 – OSP Process which is typically processed in 1-up or array form.

Table 16 – OSP Process

Organic Solderability Preservative Process Flow			
Clean	Microetch	Pre-dip	Flood OSP

Electroless Nickel Immersion Gold or ENIG is a process that uses gold to protect the nickel. The gold dissolves into the solder during assembly. Gold thickness over 4 micro inches can cause solder issues. The advantages of ENIG is a flat surface, lead free, good for plated through holes, and a long shelf life. The disadvantages include ENIG is not able to be disassembled, expensive, black pad, and a complicated process. Typically processed in 1-up or array form the process can be shown in Table 17 - ENIG Process Flow.

Table 17 - ENIG Process Flow

ENIG Process Flow						
Clean	Microetch	Catalyst	Electroless Nickel	Rinse	Immersion Gold	Rinse/Clean

Hard Electrolytic Gold also known as Flash Gold and Tab Gold. For the electrolytic process all copper surfaces to be plated must be electrically connected to a power supply, unless gold as etch process is used. Un-plated copper must be covered by a mask. The advantages for Hard Gold is the durability, lead free, and a long shelf life. The disadvantages of Hard Gold include its cost making it very expensive, the extra processes make it labor intensive, the use of tape, and a difficulty with other surface finishes. The process flow can be shown in Table 18 - Hard Gold Process Flow which is typically processed in production panel form.

Table 18 - Hard Gold Process Flow

Hard Gold Process Flow								
Appy Tape	Clean	Microetch	Electroless Nickel	Rinse	Electrolytic Gold	Rinse	Strip tape	Clean

In the PCB layout for the board the software has to be chosen to layout the board to send to a manufacture. The software that the team has chosen to use is a free version of Eagle CAD to design the PCB board. The free version of Eagle will have enough features to complete our PCB be used to place the components on the board. Traces will be drawn from the pins to power, ground, and other corresponding components that need to be connected to each other. The Auto-Logger PCB will use through holes to mount the components and the soldering will be done by the design team to save money on surface mounted components.

Choosing a PCB manufacture has different pricing and availability as well as shipping time. Some manufactures are not based in the United States so the shipping cost and time could make a cheap PCB not worth the price. The design team has narrowed down the option of PCB manufactures to 4PCB, OSH Park, ExpressPCB, and the TI Innovation Lab.

4PCB is based in the United States and they have an engineering student program that offers sponsorship of a project. With the student special there is no minimum PCB buy requirement. For $33 each the boards come with 2 layers and takes 5 days to complete the request. There is also an option with up to 4 layers for $66. 4PCB offers their own free PCB file checker to find any potential mistakes in the PCB design.

OSH Park uses ENIG finish that is manufactured in the USA and provides free shipping. 2 layer boards are $5 per square inch which includes 3 copies of the board. They ship in under 12 calendar days for prototype boards. 4 layer boards are $10 per square inch also include 3 copies of the PCB. It takes 2-3 weeks to ship for a 4 layer board.

ExpressPCB is also located in the United States and requires that their software is used to create the schematic. Their prototype service comes with 2 layers at $166 plus shipping for 4 PCB copies. They also have an option for 4 layer prototype service at $195 for 4 PCBs. ExpressPCB ships within 2 business days.

The TI Innovation lab will serve as a critical lab to solder the components to the PCB. The TI Innovation lab is located on campus in Engineering 2, and they provide soldering stations, 3D printer, and laser cutting services. The lab has all the necessary tools needed to complete a project including oscilloscope, function generator, power supply, and a digital multi meter.

Remote Server & Database

The Florida Solar Energy Center will house the remote servers used to retain all data acquired through this project. The database will be housed, operated and maintained by the FSEC. The data storage structure will be managed by the logging device, however FSEC has the ability to reorganize the received data in any manner they deem necessary. The communication protocol must ensure no data is lost in transmission and prevent subsequent deletion on the device’s local storage.

File Transfer Protocol

The Auto-Logger must communicate with the Florida Solar Energy Center’s Server in order to transmit the reservoir of collected transportation information. File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a network. FTP supports secure transmission that protects the username and password used to associate the client with the server and also encrypts the content of the files sent. To ensure protocol security and efficient data management third party FTP software would have to be implemented on the Auto-Logger.

Project Hardware and Software Design Details

After the research above has been conducted, the parts and software specifications detailed below will describe the implementation of hardware and software in the final design.

OBD-II Interface

Many different manufacturers use their OBD-II port in different manners. Older Japanese cars used ISO lines while American manufacturers used PWM and VPW, and newer cars use CAN. On top of this variation, many manufacturers use the open pins for proprietary uses (e.g. Alfa Romeo OBD command to check supercharger performance). The Auto-Logger will ignore the pins that aren't mandated by SAE J1962 because our device wouldn't be able to interface with them without potentially damaging the vehicle's computer or itself.

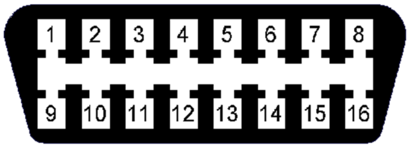
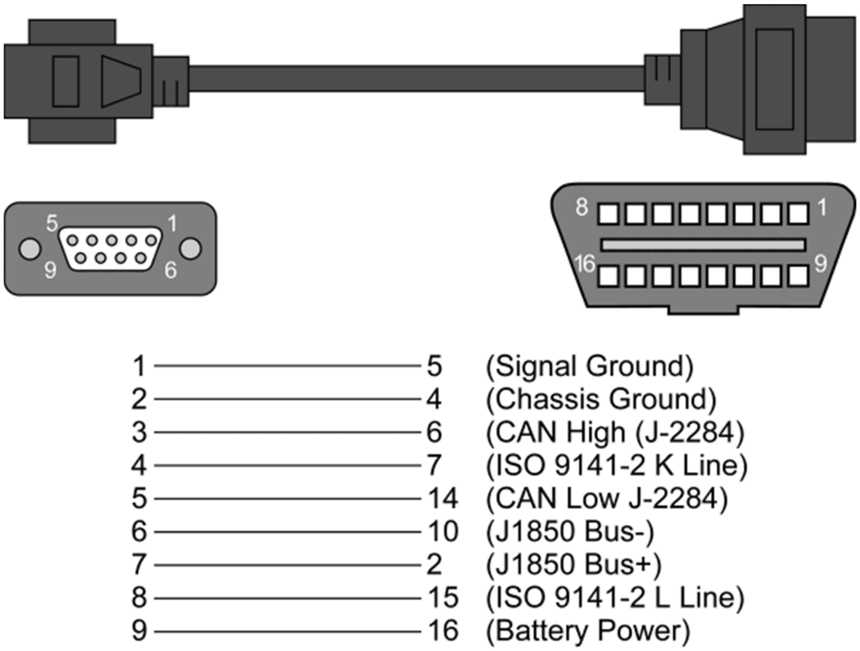


Figure 4 - OBD-II Pin-Out

Table 19 - OBD-II Interface Overview

Pin	Description	Pin	Description
1	Vendor Option	9	Vendor Option
2	J1850 Bus +	10	J1850 Bus -
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 L-Line
8	Vendor Option	16	Battery Power

In order to communicate to the OBD of the vehicle, the data transmitted and received from the car should be sent over an RS232 converter cable. Only 9 pins from the OBD-II port is needed in order call the SAE J1962 required OBD-II readings. So in order to minimize the thickness of the cable and the space required on the PCB, RS232 DB9 type cable was chosen.



1	5	(Signal Ground)
2	4	(Chassis Ground)
3	6	(CAN High (J-2284)
4	7	(ISO 9141-2 K Line)
5	14	(CAN Low J-2284)
6	10	(J1850 Bus-)
7	2	(J1850 Bus+)
8	15	(ISO 9141-2 L Line)
9	16	(Battery Power)

Figure 5 - OBD-II to DB9 Conversion Chart

There are 6 OBD-II Parameter IDs that are universal with all compliant vehicles. These modes, allow different data sets to be harvested by the Auto-Logger.

Table 20 - OBD-II Interface Modes

Mode Number	Mode Description
01	Show Current Data: Streams real time data from the OBD to requesting device
02	Show freeze frame data: Takes a snapshot of the current state of the vehicle
03	Show stored Diagnostic Trouble Codes
04	Clear Diagnostic Trouble codes and stored values
05	Test results, oxygen sensor monitoring (non CAN only)
09	Request vehicle information

OBD-II PIDs are hexadecimal based codes that are used to fetch data from a vehicle. The Auto-Logger incorporates only the PID commands that are outlined and standardized within SAE J/1939. The PIDs below outline the types of commands that will be implemented into the design. The modes that will be utilized along with their respective hexadecimal PID commands are detailed below.

Mode 1 used commands:

Mode 1 allows real time data acquisition from the time at which the PID command was sent from the Auto-Logger. PID commands 00, 1C, 20, 40, and 60, as seen in Table 21 - OBD-II Interface Mode 1 PID commands, will allow the device to identify which PIDs and standards, if any, are available to extract data from and will determine which commands will be valid with the vehicle currently being data mined. PIDs 04, 0C, 0D, 10, 50, and 5E are all used to determine internal combustion engine load on the vehicle. Under typical conditions, the device will only need PID 04 to read the engine control unit's (ECU) internally calculated value. However if the vehicle doesn't support PID 04, then the Auto-Logger will have to calculate the engine load by retrieving all the variables manually from PIDs 0C, 0D, 10, 50, 5E and using the same engine load formula that PID 04 would utilize.

PIDs 0C, 0D, 11, 1F, 5A, and 7F will be used to determine whether the car is currently being used by the driver. By using these metrics the Auto-Logger can be intelligent enough to determine if the driver had left the vehicle off and parked. This way the Auto-Logger can put itself into low power mode and keep itself from drawing too much power from the car battery. These readings such as vehicle speed can also aid in determining distance traveled and serve as a backup in the case that the GPS module has lost connection.

Table 21 - OBD-II Interface Mode 1 PID commands

PID (hex)	Description	Value Range
00	Checks which PIDs[1-20] can be used	00000000 - FFFFFFFF
04	Calculated engine load (%)	0 - 100
0C	Engine RPM (rpm)	0 - 16,383.75
0D	Vehicle Speed (km/h)	0 - 255
10	MAF airflow rate (gram/s)	0 - 655.35
11	Throttle position (%)	0 - 100
1C	OBD standards this vehicle conforms to	1 - 255
1F	Run time since engine start (seconds)	0 - 65,535
20	PIDs supported [21 - 40]	00000000 - FFFFFFFF
2F	Fuel Tank Level Input (%)	0 - 100
40	PIDs supported [41 - 60]	00000000 - FFFFFFFF
50	Maximum value for air flow rate from mass air flow sensor (g/s)	0 - 2550
51	Fuel Type	0 - 23
5A	Relative accelerator pedal position (%)	0 - 100
5B	Hybrid battery pack remaining life (%)	0 -100
5E	Engine fuel rate (L/h)	0 - 3212.75
60	PIDs supported [61 - 80]	00000000 - FFFFFFFF
7F	Engine runtime	(Not specified)

Mode 2 used commands:
Mode 2 takes a snapshot of the current state of the vehicle. The hex PID commands are identical to the Mode 1 PID commands listed about, with the exception that the data extrapolated from this mode is given from when the freeze frame was created. The freeze frame is initiated once the Auto-Logger puts the OBD-II into Mode 2.

Mode 9 used commands:
Mode 9 commands are useful to identify the vehicle the Auto-Logger is operating on. This will be useful when trying to sort data from multiple vehicles using duplicate Auto-Loggers in the future in a qualitative manner. A vehicle identification number (VIN) is a 17 - character affixed to every automobile since 1981. A VIN is the most reliable way to track what kind of vehicle is being operated on because no two cars built within 30 years of each other can share the same VIN. This will allow FSEC greater flexibility in organizing the data retrieved from multiple Auto-Loggers in the future.

Table 22 - OBD-II Interface Mode 9 PID commands

PID (hex)	Description	Value Range
00	PIDs supported [01 - 20]	00000000 - FFFFFFFF
02	Returns Vehicle Identification Number (VIN)	17-20 character ASCII hexadecimal value
0A	Engine control unit (ECU) name	20 character ASCII hexadecimal value

Modes 3, 4, and 5:
Modes 3, 4, and 5 are irrelevant to project design specifications and do not aid in accomplishing any of the data procurement from the vehicle. These modes are only used when trying to diagnose "*check engine light*" errors by checking and clearing error codes from the vehicle's dashboard display. Because this device will not be used as a typical car-scanner, these modes and any data from them will be ignored by the device.

Power

The Auto-Logger will be powered solely from the OBD-II supplied 12V pin. Many devices such as the Wi-Fi module, GPS receiver, and MicroSD card circuits require 3.3V - 5 V to operate in nominal conditions. In order to reduce the supplied voltage to a sufficient level without requiring the use of a heat sink sufficient spacing must be introduced to accommodate for a significant temperature gradient.

Nominal Battery Discharge Rate Calculations:
A standard small car battery is about 45 Amp-hours. That means that it will supply over two amps for 20 hours. A battery should not be discharged at a higher current draw, or asked to deliver more amps than its amp/hour rating divided by 10 in order to get maximum capacity out of it. During Low Power Mode (LPM) the MCU, Wi-Fi NIC, LCD, and GPS module only draw a total of 0.56mA. During peak current draw the device's components draw up to 0.575 A.

To ensure that the Auto-Logger will not be a liability to FSEC or any of the participants in their study, proper battery discharge rates on the vehicle’s car battery are essential. The formula used to reach our conclusion is Peukert’s Formula, depicted below in Equation 1 - Peukert's Formula for Battery Discharge Time.

$$\mathrm{T} = \frac{\mathrm{C}}{I^n}$$

Equation 1 - Peukert's Formula for Battery Discharge Time

Where C is the nominal battery capacity, I is the amps drawn by the Auto-Logger, and n being the Peukert Number for the vehicle’s car battery. As previously stated, 45Ah is the nominal battery capacity, the current draw characteristics are depicted below:

Table 23 - Peak Usage Current Characters

Component	**Required Current (mA)**
MCU	200
Wi-Fi Module	170
GPS Module	75
MicroSD	80
LCD	50
Total	**575**

Table 24 - Low Power Mode Current Characteristics

Component	Required Current (mA)
MCU	.015
Wi-Fi Module	.05
GPS Module	.015
MicroSD	.015
LCD	.015
Total	**.56**

Typically, Peukert's Number is determined through iterative hardware analysis performed by the battery manufacturer. But due to the many types of car batteries drivers may have installed onto their vehicle in a multitude of various conditions, the number can fluctuate between 1.1 and 1.3, being the worst discharge rate. Because of the various conditions, the calculations will use the worst case scenario of 1.3.

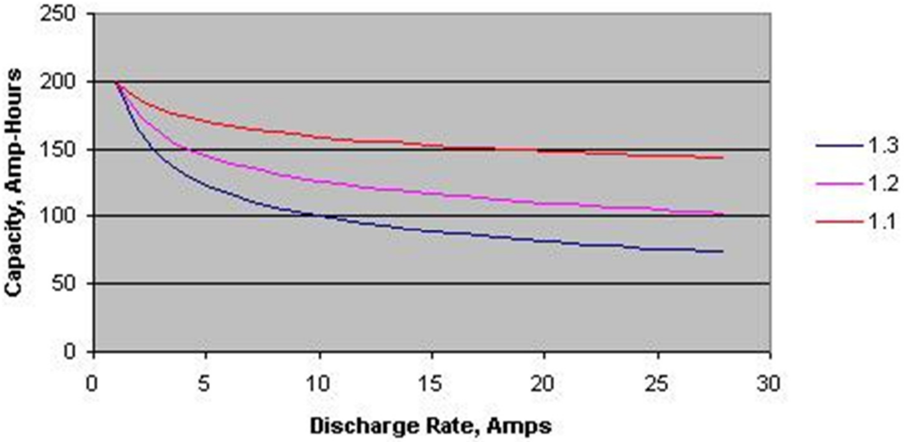


Figure 6 - Vehicle Rated Lead Acid Battery: Capacity vs. Discharge Rate (Awaiting Confirmation)

Generally batteries are measured in cycles or how many times they can be discharged and recharged before they will no longer take a full charge. The depth of discharge is by in large the main factor that determines the life expectancy of a battery. The main issue at hand is that car batteries are treated differently because they are not designed to even 20% of its charge capacity and will be damaged if dropped below that threshold. Thus the calculations below will prove that the Auto-Logger will not cause such issues even in the event that low power functionality is compromised.

Through application of Equation 1 - Peukert's Formula for Battery Discharge Time:

$$\mathbf{20\%\ \boldsymbol{Discharge\ Time} = \frac{45}{0.575^{1.3}} * (0.2) = 18.478\ \boldsymbol{hours}}$$

Equation 2 - Lead Acid Battery Discharge Time with Peak Load

$$\mathbf{20\%\ \boldsymbol{Discharge\ Time} = \frac{45}{0.00056^{1.3}} * (0.2) = 151,914\ \boldsymbol{hours}}$$

Equation 3 - Lead Acid Battery Discharge Time with LPM Load

The above calculations show that if the device is in LPM while the vehicle is off that the driver of the vehicle being operated on will not be in danger of damaging their vehicle battery. It also proves that if the device fails to switch to LPM for any reason, that the vehicle battery will not sustain any damages onto the car battery.

Wireless Communications

Hardware

The Auto-Logger will integrate the ESP8266EX Wi-Fi Module to provide the device with a reliable, power efficient internet interface. The ESP8266EX has a uniquely small package size with 32 individual pins. While module package will not take much room on the final board the team must plan to give additional space to accommodate the associated input pins, such as signal amplification antennas if desired.

It is important to note that the ESP8266EX requires 3.3V, a spike in voltage would likely lead to the device overheating. While the designed PCB does have a 3.3V Voltage Regular on board, we must be aware of the device’s capability in providing sufficient current to power this Wi-Fi module. This may be considered a design constraint, however insignificant it may seem.

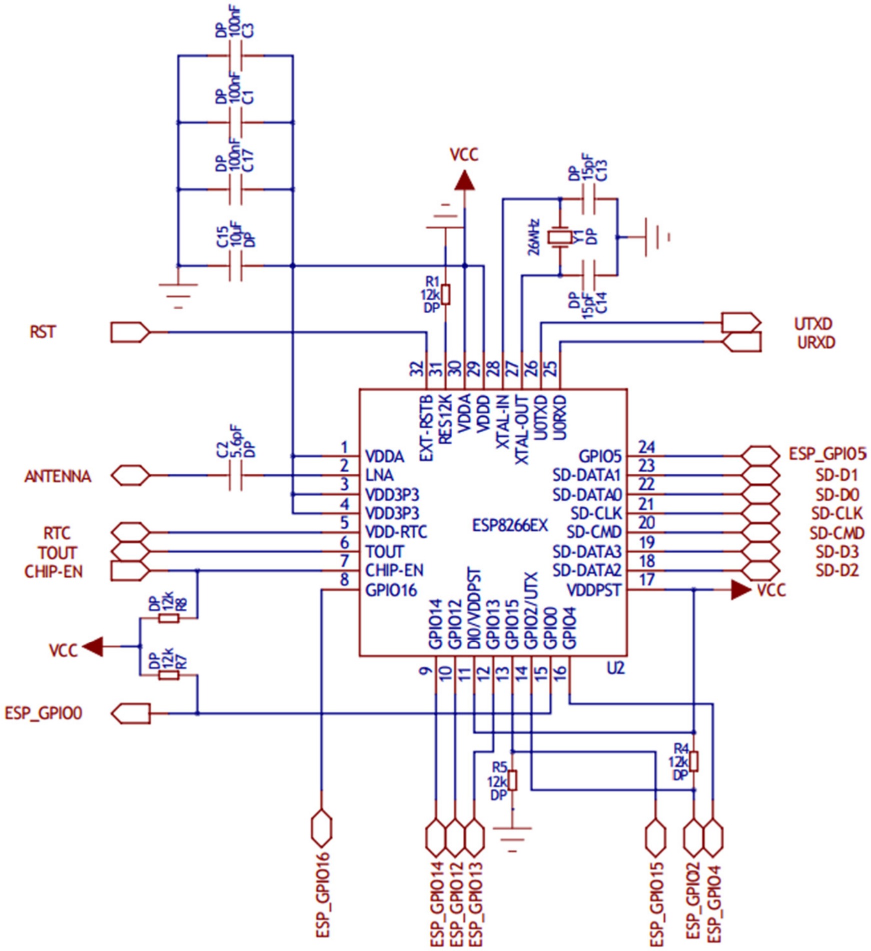


Figure 7 - ESP8266EX Wi-Fi Module LTSpice Schematic

The ESP8266EX Wi-Fi module will require some additional hardware, such as capacitors for voltage regulation and resistors for current management, in order to properly communicate with the MCU and the FSEC server. While this module comes complete with a built in antenna, we have put research into improving the capability of our device by reducing the required signal strength to establish a stable connection. In order to achieve this goal, an amplification antenna may be utilized to improve receive signal strength. This is important as the Wi-Fi signal in the parking lots and throughout campus is patchy which could result in a constant failure to establish a connection for some participants. However, precautions must be taken when selecting a specific device so as not to impose on the spatial necessities of the driver.

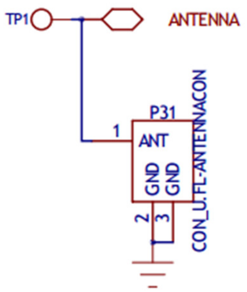


Figure 8 – Antenna LTSpice Schematic (Optional)

ESP8266EX comes equipped with a high frequency crystal oscillator, which can range from 26MHz to 52MHz, and a real time clock. Because the crystal oscillator can experience a large frequency drift based on its operating temperature, it is necessary to wait until the vehicle has reached a stable temperature to begin wireless communications. In order to meet this constraint, the Auto-Logger will only select and establish a wireless connection once the vehicle has stopped operation. This approach makes it possible for the device to remain asleep during rapid cooling of the cabin from the A/C, when the most significant frequency drift would be experienced.

The clock generator and all of its associated components, including inductors, loop filter and varactors, are integrated on-chip. This clock generates quadrature 2.4 GHz clock signals for the Receiver and Transmitter. Furthermore, the clock has built-in calibration and self-test circuits which will be useful to ensure a stable clock frequency even with temperature fluctuations. The ESP8266EX quadrature phases and phase noise are optimized on-chip with patanted calibration algorithms to ensure the best receiver and transmitter performance.

It is understood that the Auto-Logger, with a stable connection, will have the ability to dump its memory banks into the FSEC mySQL database within 15 minutes, while accomodating some miscommunications. Substituting values provided in into the total transmission time would come to 13 minutes 19 seconds. Values were attained by assuming “worst case scenario” conditions for transfer. Data attained per day should not exceed 10 MB, accumulating 500 MB of data would take almost 50 days of continuous on-campus navigation. Since data will only be collected for vehicle operation while within campus bounds, the amount of data queued for transfer should never exceed 100MB. Moreover, a requirement for participants in this study is they should be driving to campus **at least** twice per week. This will allow the Auto-Logger to keep its data transfer down to a minimum, transmitting the small magnitude of data it has collected on a regular basis.

Table 25 - Worst-Case Conditions

Variable	Value
Amount of Data	500 MB
Transmission Speed	5 Mbps

$$\frac{\boldsymbol{Amount\ of\ Data\ (MB) * 8}}{\boldsymbol{Tranmission\ Rate\ \left(\frac{mb}{s}\right)}} = \boldsymbol{Time\ (s)}$$

Equation 4 - Data Transfer Rate

Power Management

The Auto-Logger will perform all data transfer in a short period of time after the vehicle has been shut down while on the UCF campus. In adherence to the sudden disappearance of the system's typical power source, the module needs to be temporarily power by an external battery.

While not in use the ESP8266EX must be held in one of operation modes depicted below in Table 26 - Low Power Operation Modes. The MCU will initialize the interface with the Wi-Fi module and immediately put it into Light-Sleep operation. The Auto-Logger uses Light-Sleep operation, as opposed to Deep-Sleep operation, because there is sufficient power to support the system even with the ESP8266EX module running at all times. However, we believe that it is unnecessary for the device to operate on full power if not required. The Auto-Logger does not utilize the Deep-Sleep mode because it requires an interval based wake up setting. Because the vehicles will be used for personal activities it would be undesired for the Wi-Fi module to consistently wake up at predetermined intervals if the vehicle is nowhere near the transmission location. The Light-Sleep low power operation mode is the most power and computationally efficient of the selectable options.

Table 26 - Low Power Operation Modes

Mode	Modem-Sleep	Light-Sleep	Deep-Sleep
Action	Turn off Wi-Fi Modem CPU and other Peripherals are still running.	Turn off Wi-Fi modem, crystal clock and PLL CPU and other peripherals are suspended.	Only RTC circuit is working, all others are off, chip is in low power standby mode.
Current	10-20mA	.5mA	10-20uA
Wake-Up	Yes	Yes	Wake up based on predefined interval setting **ONLY**

Software

The Auto-Logger will implement the ESP8266EX Wi-Fi Module in order to connect to the UCF Wi-Fi network. The main MCU will initiate the state sequence once the GPS localization identifies the vehicle as “on UCF premises” and the vehicle switches into park. Once initiated by the MCU the Wi-Fi module will wake from Low Power Mode (LPM) and begin to poll the specified network ID. With the confirmation of a stable network connection established, the communications state sequence will progress as depicted in Figure 9 - Communication Sequence for Data Transfer.

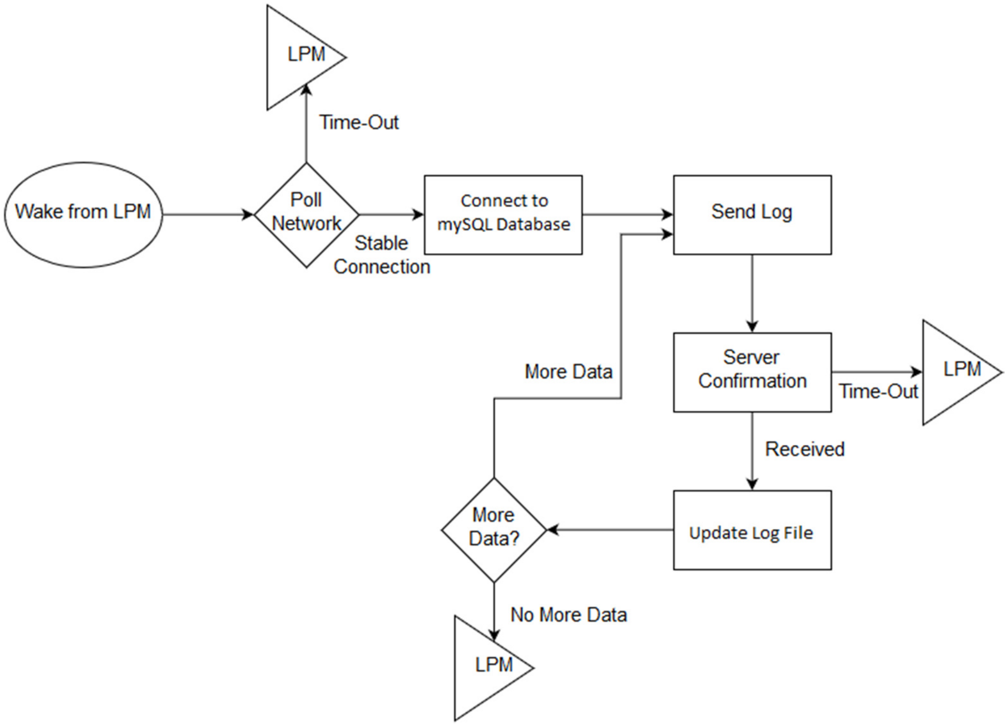


Figure 9 - Communication Sequence for Data Transfer

There are numerous possible contingencies and errors that can occur during wireless transmission, to manage these errors and more, watchdog timers and status checks will be incorporated to avoid infinite loops and misrepresentation of data transmission.

Network Polling

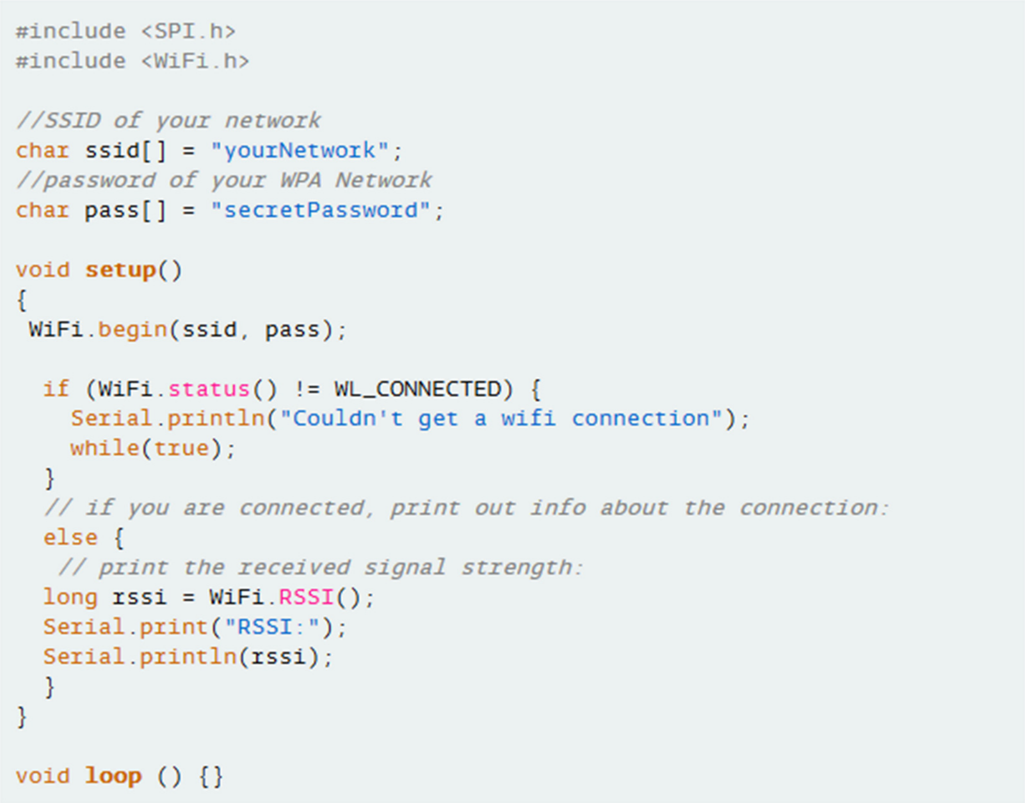
Polling of the UCF network is critical for the establishment of a reliable connection with the FSEC Server. As depicted in Table 9 - Signal Strength Required for Application, the received signal strength indicator (RSSI) is utilized to hold a tight threshold on the signal strength. To ensure a reliable connection, the Auto-Logger will enforce a threshold of ***-80dBm***. This threshold will guarantee accurate packet transfer and a drastic reduction in packet loss leading to repeated transmissions.

RSSI is a measurement of how well a device can hear a signal from an access point or router. Not to be confused with the associated transmit power of the access point or router, the RSSI is a value useful for determining the strength of a specified signal you can detect. The ESP8266EX will rate the specified signal, UCF WPA-2 in our case, on scale from 0 to 255 where the higher the number the strong the signal is. Through analysis of Equation 5 - Power Received (watts) to Signal Strength (dBm), we can see RSSI is perfect for creating a signal threshold to establish a reliable connection.

$$\mathbf{dBm} = \mathbf{10} * [\mathbf{log}\,(\mathbf{1000} * \mathbf{P})]$$

Equation 5 - Power Received (watts) to Signal Strength (dBm)

When the vehicle is placed into Park, the Auto-Logger believes this would be the most reliable time to connect to the UCF Wi-Fi because the device is stationary. When prompted by the GPS and OBD-II data streams, the MCU will initiate the connection sequence with the specified network, which will be the *UCF WPA-2* network provided throughout the UCF campus. After a connection is establish the MCU can begin to poll the signal strength through the application of Equation 5 - Power Received (watts) to Signal Strength (dBm). Open-source libraries for the ESP8266EX provide readily available RSSI interpretation for the ATMega2560, supplying the Auto-Logger with a clear and intuitive capability to study the signal strength and determine whether or not such a network could reliably support the transfer of our mission critical data.



```
#include <SPI.h>
#include <WiFi.h>

//SSID of your network
char ssid[] = "yourNetwork";
//password of your WPA Network
char pass[] = "secretPassword";

void setup()
{
 WiFi.begin(ssid, pass);

  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Couldn't get a wifi connection");
    while(true);
  }
  // if you are connected, print out info about the connection:
  else {
   // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("RSSI:");
  Serial.println(rssi);
  }
}

void loop () {}
```

Figure 10 - Example of Network Acquisition & RSSI for Arduino

Similar to Figure 10 - Example of Network Acquisition & RSSI for Arduino, once the MCU confirms an internet connection has been established it will queue the Wi-Fi module to begin polling the signal strength using RSSI. The MCU will store the values attained through the performance of RSSI, after 25 readings have been accumulated the MCU will determine whether the connection is suitable for data transfer by converting the data received into a signal quality measured in percent, this conversion is depicted below in Equation 6 - Signal Quality from RSSI value. Because the RSSI level can be limited to a range of -50dBm (pristine) to -100dBm (practically noise-floor), an accurate depiction of the signal strength can be represent as a percentage where a higher percent represents a stronger signal. This signal quality will be compared to the threshold quality level which, based on the influence of noise, will remain around 40% or -80dBm.

$$\boldsymbol{Signal\ Quality}(\%) = \mathbf{2} * [\boldsymbol{RSSI}(\boldsymbol{dBm}) + \mathbf{100}]$$

Equation 6 - Signal Quality from RSSI value

If the MCU determines the signal strength is up to par, the connection sequence will move forward to establishing a connection with the FSEC MySQL server in order to transmit data through secure TCP. However, if the signal strength fails to reach the threshold the MCU will revert the Wi-Fi module back into Light-Sleep operation mode. The MCU will reattempt to transmit data the next time the vehicle is park on a UCF campus.

Communication Protocols

The Auto-Logger will utilize secure Transmission Control Protocol (TCP) as its medium of communication with File Transfer Protocol (FTP) as it excels in the transfer of large files and comes equipped with many features which provide reliability support. FTP may run in active or passive mode, this determines how the data connection is established. Active Mode, which initiates a data channel between the server and client after an initial connection confirmation, will be utilized unless the FSEC or UCF firewalls create a problem. In the event that the firewall plays an issue the design will move to implement passive mode, which allows a data connection to be established between the client and server an arbitrary port.

Transferring data requires tight regulatory work, as such FTP must utilize one of supported data representations listed in Table 27 - FTP Data Representations. As recommended and made readily apparent by innumerable implementations, Image or Binary Mode is the representation implemented in the Auto-Logger because reliability is key in such a system. This data representation will also reduce the complexity of the serial transfer between the MCU and the ESP8266EX as well as the FTP communication between the client and server, as various conversion will not be required on both ends of the communication process.

Table 27 - FTP Data Representations

Data Representations	Description
ASCII	Used for Text representations. Encodes 128 specified characters and symbols.
Image (Binary Mode)	Client send each file byte by byte, server stores byte stream as received (Recommended)
EBCDIC	Used for plain text between hosts using the EBCDIC character set
Local	Allows two computers with identical interfaces to send data in a proprietary format without converting to ASCII,

Direct Server Communication
Communication with the FSEC server will require conjoined effort from the MCU, Wi-Fi module and the MicroSD Card in order to gather, organize, prepare and transmit the desired data logs. Analysis by the server will be required to ensure the file was received successfully. Instead of deleting logs as soon as they have been transmitted to the server, the Auto-Logger will implement a circular storage methodology. This technique will continue to save data logs on a per second basis until the entire storage unit has been filled to capacity. Once storage capacity has been reached, the device will begin to overwrite files starting from the oldest file and progressing towards the newest thus completing the circle.

The Auto-Logger will connect to the FSEC server securely with the use of a private username and password combination. Communication will utilize TCP port 443 as it, along with TCP port 80, remains open even when such a server is stationed behind a firewall. Once a connection has been established with the server the Auto-Logger will transmit the contained data files, saving them into a MySQL database provided by the FSEC. The Auto-Logger, through the use of a data transfer log, will keep track of transferred data ensuring the server only receives what is needed.

GPS Localization

Hardware

The Auto-Logger will utilize Adafruit's FGPMMOPA6B GPS module for accurate real-time vehicle localization. This module implements a MediaTek (MTK) chipset to provide features pertinent to the goals of the Auto-Logger such as data logging and acquisition rates which exceed 5 Hz. While the package for the device is very small, additional space on the PCB must be devoted in order to accommodate the modules associated hardware such as antenna ports such as external appurtenances. GPS Module architecture and FGPMMOPA6B interfacing schematic are depicted below in Figure 11 – GPS Module Block Diagram and Figure 12 - FGPMMOPA6B Schematic for UART Interfacing.

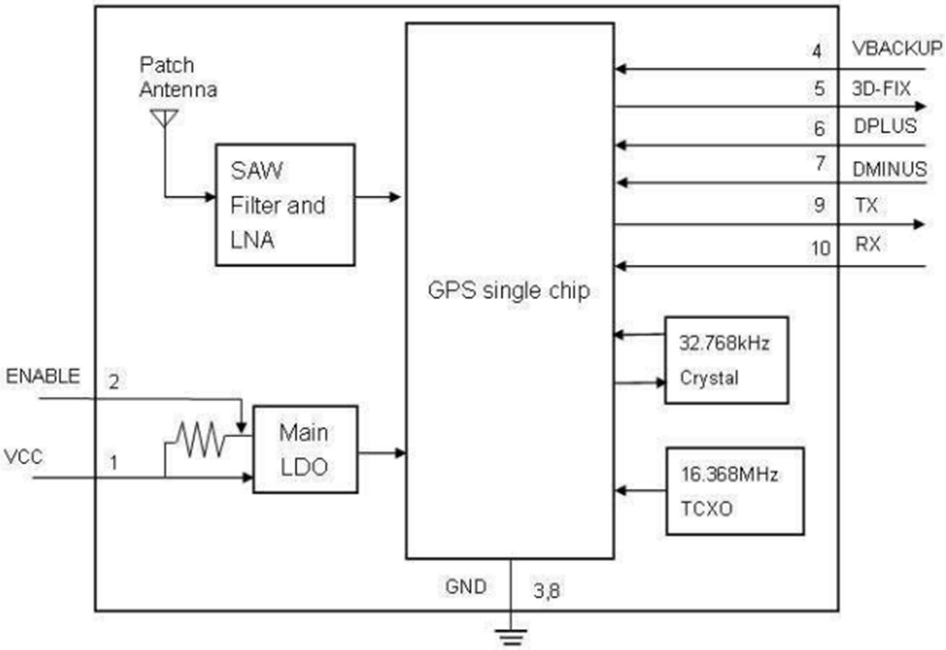


Figure 11 – GPS Module Block Diagram (From datasheet)

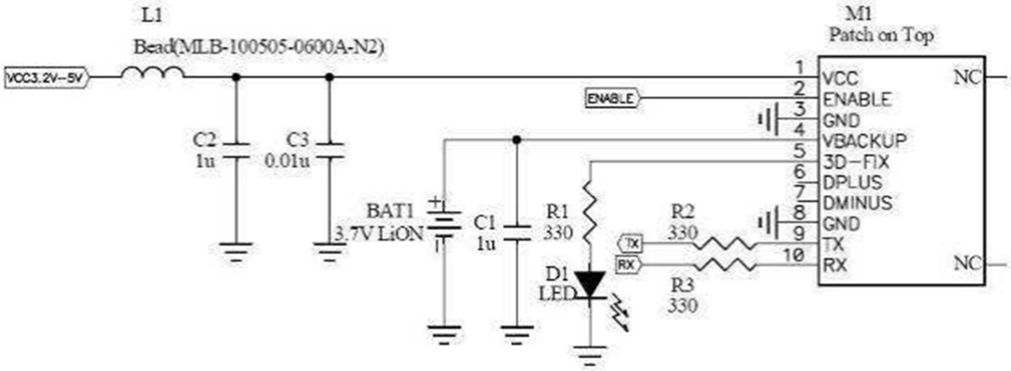


Figure 12 - FGPMMOPA6B Schematic for UART Interfacing (From datasheet)

The FGPMMOPA6B is an ultra-compact, patch on top, 66-channel GPS Engine board equipped with antenna module and MTK chipset. This receiver provides an RF solution that is high in position and speed accuracy performances, with high sensitivity and tracking capabilities in urban conditions which are ideal for the purpose of the Auto-Logger. This GPS module is an all-inclusive package providing the MKT chipset with a POT antenna, 32 MHz Crystal clock, 16 MHz temperature compensate crystal clock and main Low Dropout (LDO) Voltage Regulator.

The temperature compensate crystal clock (TXCO) is implemented in low-cost designs requiring a precision frequency source within a small space. This device has temperature compensation circuitry to suppress output frequency deviation caused by temperature changes in the surrounding environment. Such compensation is highly advantageous for the Auto-Logger since the device will remain in a vehicle for the entire product life-cycle. Atmospheric temperature in a vehicle can easily range from 25° Celsius to 70° Celsius. Experiencing the complete temperature range within an hour's time, up to several times per day is a result of typical vehicle usage, as such the TXCO is a significant asset to the reliability of the GPS acquisition system.

Temperature Compensated Crystal Clock oscillators with standard compensation techniques achieve fractional stabilities of around ±1 parts-per-million (ppm) for a temperature range of -40° C to 85° C. Practical technique harnessed by the majority of TCXO manufacturers is based on the use of varactor diodes in series with the crystal clock as depicted below in Figure 13 - Temperate Controlled Crystal Oscillator Schematic.

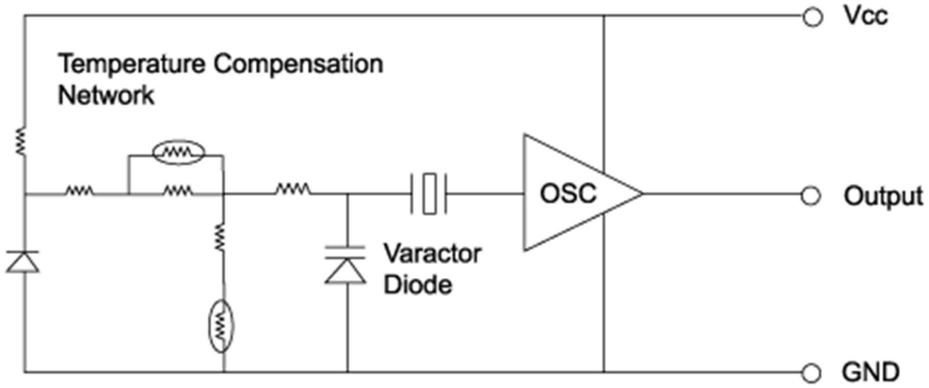


Figure 13 - Temperate Controlled Crystal Oscillator Schematic (awaiting approval)

Each individual TXCO requires personalized network compensation to be matched with its specific crystal. Subsequently, the cost of a TCXO is closely related to the difficulty of the frequency versus temperature specification. During operation, a voltage change causes a change in the capacitance of the varactor diode. This results in a change in the frequency of oscillation. The thermistor network is tailored to the crystal in order to force the voltage to vary with temperature such that the crystal's frequency variation characteristics versus its temperature change are adequately compensated for.

The 32 MHz crystal clock supplies the FGPMMOPA6B GPS module with the capability to sample global position variables up to a rate of 10Hz. Such an acquisition rate is much greater than what is required for proper operation of the Auto-Logger. Additionally, temperature fluctuation experienced in the vehicle could lead to significant variation in the frequency produced by the crystal oscillator leading to errors in GPS calculations and system data acquisition. Due to this fact, the Auto-Logger will received most, if not all, of its time management queues from the 16MHz TXCO.

The LDO voltage regular supplies the MTK chipset with performance capabilities optimized for battery-powered systems delivering low quiescent current, thus prolonging battery life. The LDO utilized in the FGPMMOPA6B GPS module is the Richteck RT9193 300mA, Ultra-Low Noise, Ultra-Fast CMOS LDO Regulator. This component provides the FGPMMOPA6B with a pristine power signal, regulated of all noise and jitter, fast switching speeds and ability to draw sufficient current even as battery voltage continues to deplete. Moreover, the LDO voltage regulator is driven with low-ESR ceramic capacitors thus reducing board space required to perform the same job. The RT9193 LDO voltage regulator helps the RPGMMOPA6B GPS module achieve such a level of efficiency and reliability, critical for low-power mobile applications.

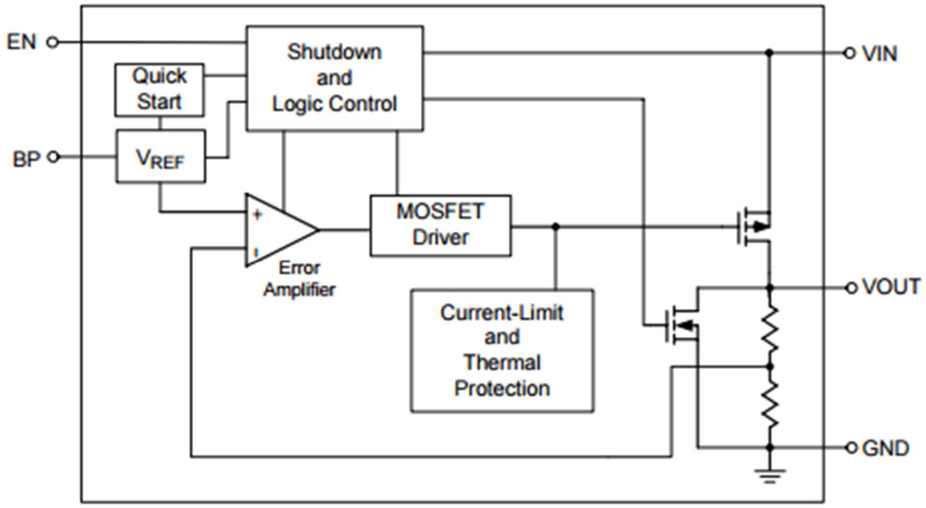


Figure 14 - LDO Voltage Regulator Block Diagram (From Datasheet)

Table 28 - FGPMMOPA6B GPS Module Pin Assignment

Pin	Name	I/O	Description
1	VCC	PI	Main DC power input
2	ENABLE	I	High active, or keep floating for normal working
3	GND	I	Ground
4	VBACKUP	PI	Backup power input
5	3D-FIX	O	3D-fix indicator
6	DPLUS	I/O	USB port D+
7	DMINUS	I/O	USB port D-
8	GND	P	Ground
9	TX	O	Serial data output of NMEA
10	RX	I	Serial Data input for firmware update

The National Marine Electronics Association defines and controls the combined electrical and data specifications for communications between marine electronics such as echo sounder, sonar, GPS and many other types of instruments. Most GPS receivers, including the FGPMMOPA6B implement the standard defined as ***0183 version 2***. These standards defined by the NMEA regulate pinouts, as seen above in Table 28 - FGPMMOPA6B GPS Module Pin Assignment, as well as GPS sentence output which is further discussed in the Software section. The Auto-Logger will make use of 8 of the 10 pins to insight GPS acquisition and communication with the MCU. The pins which will not be utilized will be 6 and 7, the pins used to connect and communicate between the GPS and another unit via USB.

Pin 5 of the FGPMMOPA6B GPS module is titled “3D-FIX” and is assigned as a fix flag output. The timing behavior of this pin can be configured by custom firmware for different application such as waking up a host MCU or initiating a sequence based on a set time interval, which the module will monitor and flag. For the purpose of the Auto-Logger, the 3D-FIX pin will not be utilized for two reasons. First, the MCU will control all peripheral components removing and inserting them into low power modes depending on the state of operation of the vehicle. Second, the Auto-Logger does not required 3D localization. In fact, including the additional elevation data into the output sentence of the GPS module, which is already has a base length of 82 characters, could introduce errors due to timing and throughput rates between the MCU and GPS module.

Since the design team has decided not to use this 3D-FIX feature, pin 5, the pin must be properly routed as determined by the manufacturer and illustrated in the module datasheet. According to the manufacturer, GlobalTechnology, the designer should keep the pin floating if they wish not to use these new 3D-FIX features.

Software

The Auto-Logger must integrate real-time localization from the FGPMMOPA6B into the main MCU in order to initiate other data logging processes which are geographically dependent such as time spent on campus, on campus parking location, network assessment and data stream connection.

Table 29 - NMEA Standardized GPS Output Sentences

NMEA Option	Output Description
GGA	Time, Position and fix type data.
GSA	GPS receiver operating mode, active satellites used in the position solution and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth and SNR values.
RMC	Time, date, position, course and speed data. Recommended Minimum Navigation information.
VTG	Course and speed information relative to the ground

As previously stated, the NMEA defines and regulates the standards for communications between all marine electronics, including GPS. The standardized output sentences, or strings of information, for NMEA regulated GPS modules are depicted above in Table 29 - NMEA Standardized GPS Output Sentences. The output sentences produced can be as long as 82 characters, to better understand the NMEA message structure a sample *Recommended Minimum Navigation (RMC)* output sentence has been analyzed in its entirety below in Table 30 - RMC Output Sentence Breakdown.

Table 30 - RMC Output Sentence Breakdown

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol description
UTC Time	064951.000		hhmmss.sss

Status	A		A = Data valid
			V = Data not Valid

Latitude	2307.1256		ddmm.mmmm

N/S indicator	N		N =North
			S = South

Longitude	12016.4438		dddmm.mmmm

E/W indicator	E		E = East
			W = West

Speed over Ground	0.03	knots	
Course over Ground	165.48	degrees	True
Date	260406		ddmmyy
Magnetic Variation	3.05,W	degrees	E = East; W = West (Need GlobalTop Customization Service)

Nide	A		A = Autonomous Mode
			D = Differential Mode
			E = Estimated Mode

Checksum	*55		
<CR> <LF>			End of message termination

NMEA regulated GPS output data produces a timestamp in the UTC time zone. There is a 4 hours difference between the UTC and EST time zones. In order to acquire the proper data and time, a simple conversion will need to be implemented once the RMC output sentence has been received by the MCU. Referencing the RMC example above, the received output is depicted below in Figure 15 - RMC Example Output Sentence:

$GPRMC,064951.000,A,2307.1256,N,12016.4438,E,0.03,165.48,260406,3.05,W,A*55

Figure 15 - RMC Example Output Sentence

The time and date values provided by this output sentence are 064951.000 and 260406 respectively, which illustrates the reading was taken at 06:49:51 AM UTC on April 26th, 2006. To properly convert the UTC time into an EST time one must subtract 4 hours. Thus, the value 40000, which corresponds to 4 hours as depicted in row 3 column 4, description of UTC time value, of Table 30 - RMC Output Sentence Breakdown, must be subtracted from the UTC Time value of 064951.000. This action results in a time of 024951.000 or 02:49:51 AM EST on 260406 or April 26th, 2006.

$$\boldsymbol{UTC = EST + 4hrs}$$

Equation 7 - Coordinated Universal Time to Eastern Standard Time

A geographic coordinate system enables every location on a spherical planet to be specified by a set of numbers or letters, or even symbols. Coordinates are often chosen such that one number represents vertical positioning while the other represents horizontal positioning. The standard depiction of global positioning uses longitude and latitude, elevation is sometime incorporated as additional information. Latitude depicts a point on the Earth's surface where the angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth. Latitude ranges from 0° at the Equator to 90° at the North or South Pole. Longitude depicts a point a point on the Earth's surface is the angle east or west from a reference meridian to another meridian that passes through that point.

Latitudinal and longitudinal information are usually communicated using decimal or Degree/Minute/Second notations. The FGPMMOPA6B GPS module implements a proprietary Degree/Minute/Second notation regulated by the NMEA standard ***0183 version 2***. Typical longitude and latitude, as well as proprietary NMEA regulated notations are depicted below in Table 31 - Geographic Coordinate System Notations.

Table 31 - Geographic Coordinate System Notations

Coordinate System Measurement	**Measurement Notation**	**Output Value Range**
Degree/Minute/Second	DD° - MM' - SS"	Degree (lat): 1-90 Degree (long): 1-180 Minute: 1-60 Second: 0.01-60
Decimal Degrees	DD.DDDDDD°	Decimal (lat): 1-90 Decimal (long): 1-180
NMEA Degree/Minute	DD° - MM.MMMMM'	Degree (lat): 1-90 Degree (long): 1-180 Minute: 0.00001-60

Due to NMEA restriction, GPS information clearly has an inherent error rate associated with it when compared to the typical coordinate system notations, specifically Degree/minute/second. The NMEA regulated notation sacrifices the high resolution associated with the *seconds* metric of the typical notation for a system tuned towards the abilities of the technologies implemented. In order to illustrate a greater localized accuracy the standard sacrifices localization resolution, the regulated notation removes the *seconds* unit, instead implementing a decimal into the *minute* unit. Due to the introduction of a decimal unit with 6 fields of accuracy the error associated with ignoring the *seconds* unit is bypassed, securing the error at the typical rate of acquiring a reading within 3 meters of its actual location 95% of the time.

In order to determine whether or not the participating vehicle is within campus bounds, the MCU must take the longitude and latitude provided in by the FPGMM GPS module and compare it to the designated campus footprint. The design team is working with the University of Central Florida to acquire realistic campus boundaries to maximize data accuracy. In the event that UCF is not permitted to release such information, the design team can acquire accurate coordinates through the United Stated Geological Services (USGS). The USGS provides free and highly accurate, up to 1 arc-sec, elevation and terrain data for the entire United States of America. Through the use of their applied resources, the design team can attain an accurate footprint of the UCF campus and its sister locations.

Microcontroller

The Atmel ATmega2560 is the implemented MCU within the Auto-Logger. It will interface with every single module embedded within the device PCB. The ATmega2560 does not come pre boot loaded with any firmware and will have to be manually programmed by using an AVR programming software or with another boot loaded ATmega2560. This process is needed to be able to upload the Auto-Logger software onto the MCU

The ATmega2560, has 54 digital Input/output ports, 15 of which can support PWM outputs. The main reason this board was preferred is because it has the capability to support 4 UART hardware serial ports. It has 7 times more flash memory available in the chip which provides the ability to write a larger program and have more available RAM at the user's disposal if the capability is needed.

The ATmega2560 is programmed by using the Arduino Integrated Development Environment (IDE) or its predecessor, Processing. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension (.ino). Arduino IDE will be used to debug and implement the final software and is readily available for free because of its open source nature. The ATmega2560 is only available in surface mount 100 pin Thin Quad Flat Pack (TQFP) packaging, as depicted in Figure 16 - ATMega2560 Pin Layout below or in Ball Grid Array (BGA) packaging. BGA packaging is very compact and allows for a very small footprint on the manufactured PCB, unfortunately it will be very cumbersome feat to attempt to troubleshoot any BGA package due to all the conductive contacts being hidden under the IC. I will also be a challenging undertaking to assemble the PCB even if there is not a need to troubleshoot the device. The TQFP package of the ATmega2560 will allow us to easily troubleshoot the IC because all the pins are exposed and easily probed. The team is also experienced enough in soldering to assemble the PCB while using TQFP components. Additionally, by using a TQFP IC the need for multilayered PCBs will diminish because the leads can be easily managed on the surface layer.

Because the ATmega2560 needs to interface with all the major modules of the Auto-Logger, it must be able to accommodate the use of several serial lines in parallel. Table 32 - ATMega2560 Pin Connections (Pt. 1) & Table 33 - ATMega2560 Pin Connections (pt. 2) below depicts the function of each pin on the ATmega2560.

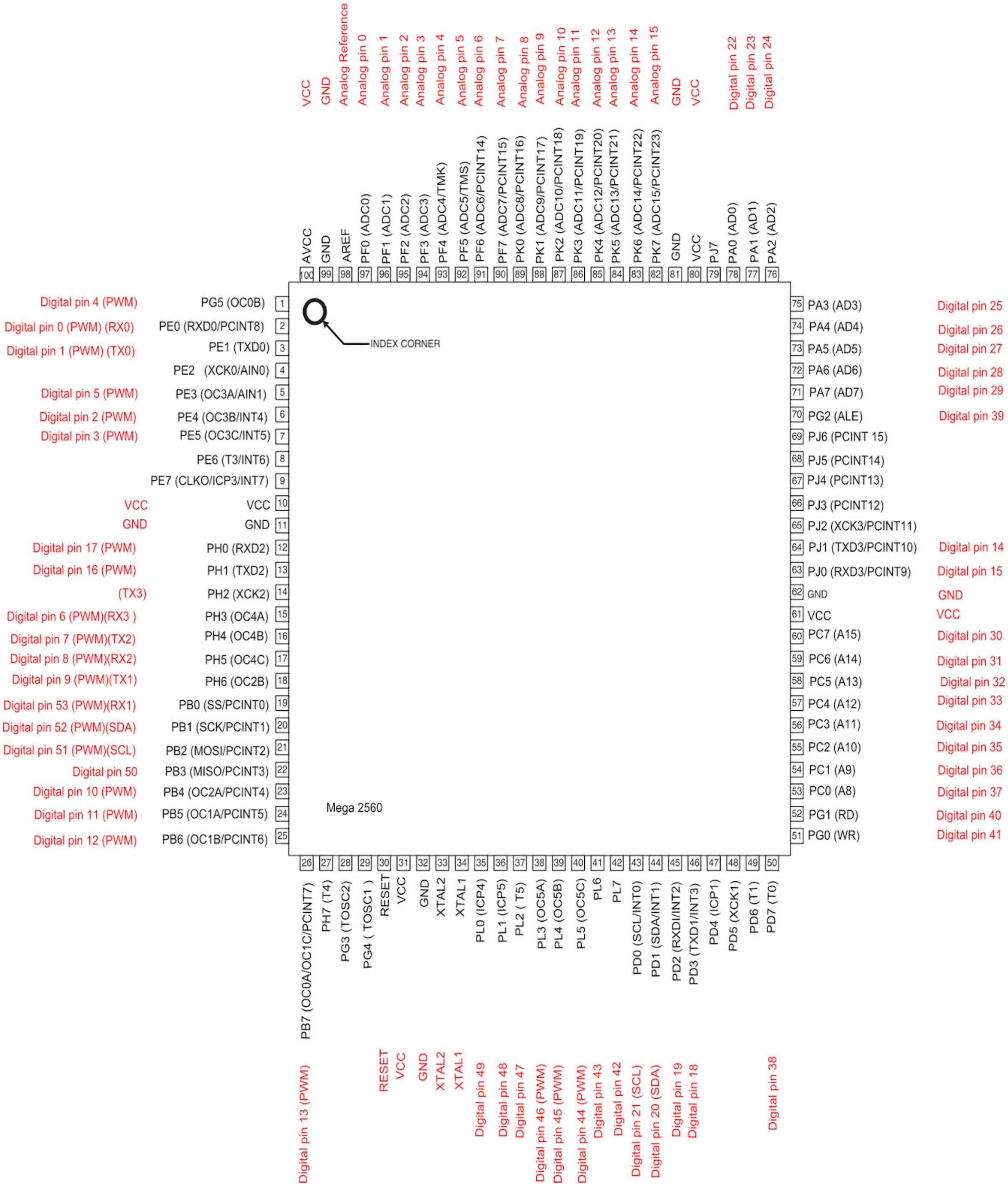


Figure 16 - ATMega2560 Pin Layout

Table 32 - ATMega2560 Pin Connections (Pt. 1)

Module	Module Input Pin	ATmega2560 Pin
GPS Module	VCC	3.3V
	ENABLE	(1.8 - 0)V
	GND	GND0
	VBACKUP	3.3V - pin must be connected for normal operation
	3D-FIX	Floating - See end of GPS Software Design for info
	DPLUS	N/A
	DMINUS	N/A
	GND	GND
	TX	RX1
	RX	TX1
Temperature Sensor	Vin	3.3 V
	Analog Vout	Analog pin 1
	GND	GND1
MicroSD	Vin	3.3 V
	DO	Digital Pin 50
	DI	Digital Pin 51
	CLK	Digital Pin 52
	CS	Digital Pin 53
OBD-II UART	TX	RX2
	RX	TX2
	GND	GND
RTC Module	Analog 1	Analog 2
	Analog 2	Analog 3
	Vin	5V
	GND	GND0

Table 33 - ATMega2560 Pin Connections (pt. 2)

Module	Module Input Pin	ATmega2560 Pin
Wi-Fi	TX	RX0
	RX	TX0
	GPIO1	Digital Pin 12
	GPIO2	Digital Pin 11
	3.3 V	3.3 V
	Chip_PWR	3.3V
	GND	GND0
	Reset	N/A
LCD Panel/Screen	V_dd	5V
	V_gg	5V
	V0	5V
	DB0-DB7	Digital Pins [32-41]
	CS2	Digital Pin
	CS1	Digital Pin 30
	RES	Digital Pin
	R/W	Digital Pin 31
	D/I	Digital Pin
	E	Digital Pin
	VEE	5V
	A	10V
	K	GND0

Temperature Sensor

The TMP36 sensor is a low voltage, precision centigrade temperature sensor. This sensor provides an output voltage which is linearly proportional to the centigrade temperature scale. The TMP36 can provide typical accuracies of ±2℃ over the -40℃ to +125℃ temperature range and does not require initial external calibration to deliver such accuracies.

The low output impedance of the TMP36 simply the interfacing process and is intended for single supply operation from 2.7V to 5.5V, which lays along the range the Auto-Logger will supply to its other components. This model of temperature sensor produces 125mV output at 25℃ and scales linearly as temperature increases or decreases by a factor of 10mV/℃.

The TMP36 temperature sensor, consisting of three pins, will interface directly with an analog pin on the ATMega2560. Using two simple conversions shown below, the MCU will be able to attain the ambient temperature of the vehicle cabin to within 1℃. First the MCU must convert the number into a voltage, depicted below in Equation 8 - TMP36 Output Value to Voltage (mV) Conversion, as the value produced by the TMP36 ranges from 0 to 1023 based on the temperature it is experiences. With the value stored as a voltage measurement, the MCU can now convert it directly into a temperature measured in ℃. See Equation 9 - Voltage (mV) to Temperature (℃) Conversion for details.

$$\text{Voltage at pin (mV)} = \text{(Analong Pin value)} * \left(\frac{\text{Supply Voltage (mV)}}{1024}\right)$$

Equation 8 - TMP36 Output Value to Voltage (mV) Conversion.

$$\text{Temperature } (°\text{C}) = [(\text{analog voltage in mV}) - 500] / 10$$

Equation 9 - Voltage (mV) to Temperature (℃) Conversion

To further improve the accuracy of the measurement, use the supply voltage a voltage reference to further cut down on noise. Route the Supply voltage pin to another analog pin on the AtMega2560. Use this pin as a voltage reference instead of defining the supply voltage as a static value within the MCU code. This method will require an additional resistor, to manage current draw of the voltage reference, as well as an additional register or variable to store the reference voltage value for temperature sensor conversion calculations. This method has the ability to increase the precision of the TMP36 up to ±0.1℃.

Since the TMP36 Temperature Sensor determines the temperature of its environment based on the physical properties of diodes, there are no time constraints on the availability of information, mean a temperature reading can be taken as often as the MCU can request the information. The property which this component takes advantage of states that as temperature increases, the voltage across a diode increases at a known rate. By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature.

Real Time Clock

The DS1307 real time clock will be used in low power mode when the car is off. The DS1307 serves to keep track of how long the vehicle is parked in one location until the vehicle is turned on and begins moving again. The DS1307 will use battery power in low power mode because the DS1307 cannot draw voltage from the vehicle's battery without potentially draining the battery. The DS1307 uses 8 bits and 8 address lines to keep track of time from seconds to the year. The day of the week register increments at midnight and the days are defined by the user, such as 1 equals Sunday. The accuracy of the clock is determined by the 32.786 kHz crystal oscillator. When reading and writing the time and date registers, secondary buffers are used to prevent error when the internal registers update. When the time and date is read the buffers are synchronized to the internal registers. The time information is read from the secondary registers while the clock continues to run. This would eliminate the need to re-read the registers in case the internal registers update during a read.

The DS1307 operates as a slave that needs to be controlled by a master device that generates the serial clock, controls the bus access, and generates the start and stop conditions. The data transfer can only be initiated only when the bus is not busy. While data is being transferred the data line must remain stable whenever the clock line is high. Changes in any data line while the clock line is high will be interpreted as control signals.

To start a data transfer change the data line from high to low while the clock is high. To stop a data transfer, change the data line from low to high while the clock is high. The master device must acknowledge the reception of each byte by generating an extra clock pulse with the acknowledge bit. A device that acknowledges must pull down the SDA line during the acknowledge clock pulse so that the SDA line is stable low during the high period of the acknowledge-related clock pulse. On the last byte the MCU must signal an end of data. To do this, the slave must leave the data line high to enable the master to generate the stop condition. To run without power from the car the DS1307 will use a CR2032 battery to operate. The CR2032 is a small lithium coin that has sufficient voltage output of 3V to operate the DS1307. The DS1307 is needed because when the ATMega2560 goes into low power mode the clock shuts off failing to track the time that has passed while in low power mode.

Liquid Crystal Display

The Liquid Crystal Display (LCD) module that will be used as a physical display for the information pertinent to proper device operation is the Graphic LCD 128 by 64 STN LED. This LCD module uses the GDM12864HLCM, KS0108B, and KS0107B as the driver for the 64 Channel Segment. LCD screens and panels work through the manipulation of the physical properties of liquid crystals, which do not emit light directly, and their interaction with light (photons). Because LCD panels produce no light of their own, they operate by uniquely manipulating the polarizing filters, which are oriented perpendicular to one another. The liquid crystals between these filters permit the passing of light that would otherwise be blocked due to the physical properties of light. When a large enough voltage is applied to a segment, the liquid crystals untwist, thus no longer rotating the light which leads to a dark segment, or pixel. This is how LCD panels and screens operate on a very basic level.

The LCD screen has six layers that create the display that is visible. The top layer is a polarizing filter with a vertical polarizing axis and without it the display cannot be seen. The following layer is the glass substrate with vertical smooth ridges followed by the liquid crystal. Another Glass substrate is behind the liquid crystal except this glass substrate has horizontal ridges. The fifth layer is a polarizing film with a horizontal polarizing axis. The last layer is a backlight that serves as a light source to produce the images. To display data treat the display screen as RAM and write a byte to an address. The address is the rows and columns in a 128 by 64 with the columns being 1 to 128 and the rows from 1 to 64. To display select the desired address in the code and input the value that will be displayed.

The GDM12864HLCM is the liquid crystal display module, it incorporates 20 pins in its minimalistic design. There are 8 pins, 4 to 11, allotted for data transfer, these pins are used specifically for data transfer of the display characters. The supply voltage required to operate the LCD is typically 5V with an upper and lower threshold voltage of 5.25V and 4.75V respectively. The display control instructions for the LCD Module are depicted below in Table 34 - Specified Function of Control Instructions. Table 35 - Display Control Instruction from GDM12864H Datasheet breaks down how the LCD Module reads/writes the data to/from the MCI and displays that data to the LCD screen for the operator.

The Y address is automatically increased by 1 when read/write operations display data. When DB0 is set to 1 the data will be displayed on the LCD, and when DB0 is set to 0 the display data is still stored in the RAM. In

Table 36 - PIN connections for LCD from GDM12864H Datasheet the pin number with the description of function can be shown. These pins will be connected to the Arduino so the data retrieved from the car can be properly displayed to show proper transmission of data. The Auto-Logger won't use the write function as much in the function because the LCD won't be taking any inputs just displaying data from the OBD-II through the Arduino. Figure 17 - GDM12864H LCD Module Block Diagram illustrates the connection of the various pins to their corresponding LCD segments. This information is critical in its use as a reference to correctly connect the Arduino and LCD Module, preventing the display of errors on the LCD due to routing errors. IC1 and IC2 are represented by the KS0108B, and IC3 represents KS0107B.

Table 34 - Specified Function of Control Instructions

Instruction	Function
Read Display data	Reads data from display data Ram to the data bus
Write Display Data	Writes data into display data RAM. After writing Instruction, Y address is incriminated by 1 automatically
Status Read	Reads the internal Status: BUSY 0: Ready 1: In operation ON/OFF 0: Display ON 1: Display OFF RESET 0: NORMAL 1: RESET
Set address(Y address)	Sets the Y address in the Y address counter
Set Display Start Line	Indicates the display data RAM display at the top of the screen.
Set address(X address)	Sets the X address at the X address register.
Display on/off	Controls the display (ON or OFF). The internal status and the DDRAM data is not affected. 0: OFF 1: ON

Table 35 - Display Control Instruction from GDM12864H Datasheet

Instruction	**RS**	**RW**	**DB7**	**DB6**	**DB5**	**DB4**	**DB3**	**DB2**	**DB1**	**DB0**
Read Display data	1	1	Read Data							
Write Display Data	1	0	Write Data							
Status Read	0	1	Busy	0	ON/OFF	Reset	0	0	0	0
Set address(Y address)	0	0	0	1	Y address (0-63)					
Set Display Start Line	0	0	1	1	Display Start line (0-63)					
Set address(X address)	0	0	1	0	1	1	1	Page(0-7)		
Display on/off	0	0	0	0	1	1	1	1	1	0/1

Table 36 - PIN connections for LCD from GDM12864H Datasheet

Pin Number	Symbol	Description
1	V_{dd}	Supply Voltage for logic and LCD
2	V_{ss}	Ground
3	V0	Operating Voltage for LCD
4-11	DB0-DB7	Data Bit 0-7
12	CS2	Chip Select signal for IC2
13	CS1	Chip select signal for IC1
14	RES	Reset Signal
15	R/W	Read/Write
16	D/I	data/instruction code
17	E	Chip enable signal
18	VEE	Operating Voltage for LCD
19	A	Backlight Power supply
20	K	Backlight Power Supply

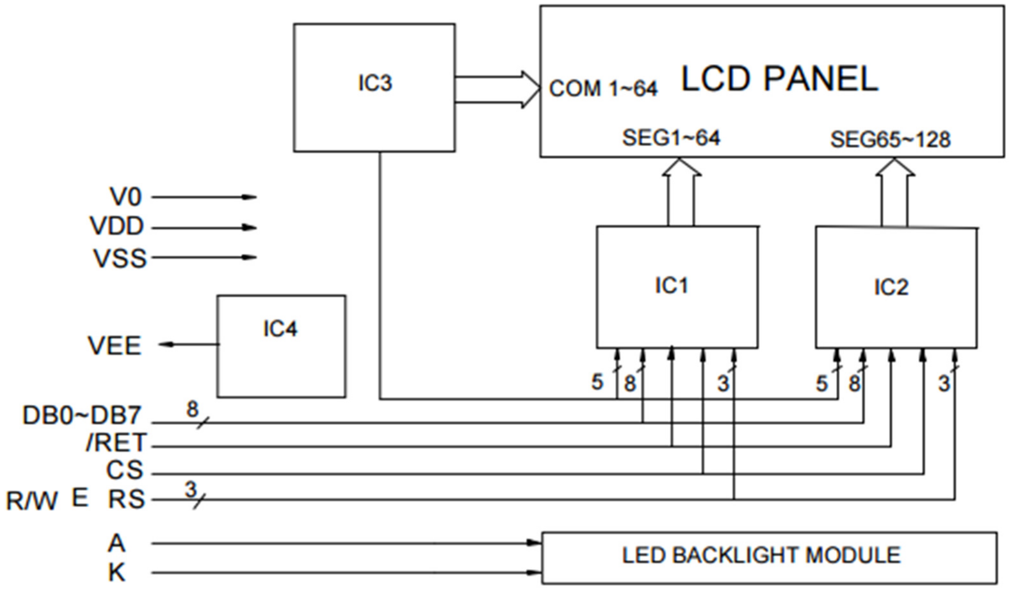


Figure 17 - GDM12864H LCD Module Block Diagram

Database

Local Storage

SD Card Pin Setup

The Data collected from the OBD-II port will be stored locally in a MicroSD card. The data needs to be stored locally because the vehicle will not always be able to transmit data, so it needs to be stored, so it can transferred to FSEC servers at a later date. The MicroSD card can vary in size depending on how long in between data is sent to the FSEC servers. The MicroSD Card will have to be of sufficient size so space is not limited and run the risk of losing data due to lack of memory. The MicroSD Card Breakout board will be used to read and write to an MicroSD Card. The Breakout board will be connected to the Arduino. The Breakout Board has 8 pins that can be shown in Table 37 - Micro SD Card Pins. The CD is the card detect pin and will short to ground when a card is inserted. The CLK pin will be connected to pin 52 on the Arduino. DO will be connected to pin 50. DI will be connected to pin 51. CS will be connected to pin 53 on the Arduino. The MM74HC 4050M hex inverting logic level down converter , LP2981-N is a low dropout regulator from Texas Instruments, and a MicroSD and MMC card reader are the components used to read the MicroSD card on the breakout board that will have to converted in Eagle to the Auto-Logger PCB. In Eagle the PCB design for the MicroSD card can be shown in Figure 18 - MicroSD Card Module Schematic.

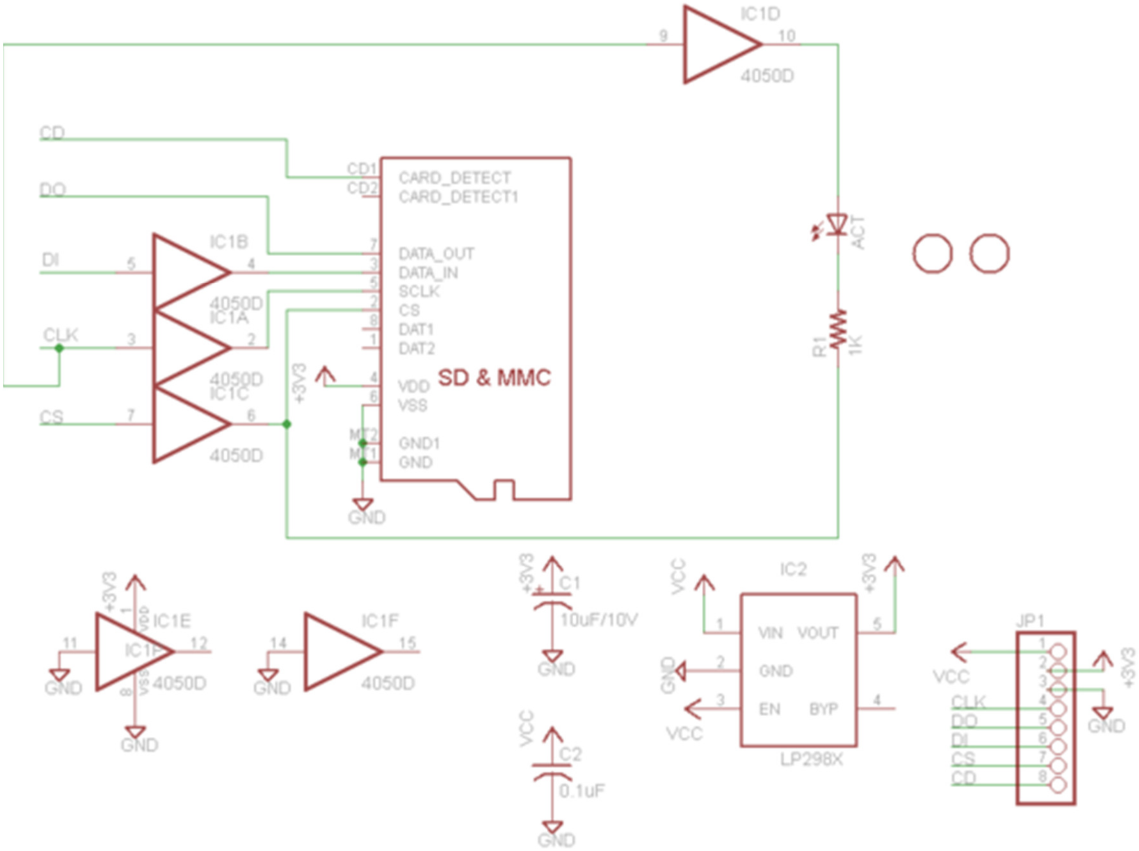


Figure 18 - MicroSD Card Module Schematic

Table 37 - Micro SD Card Pins

PIN	Function
5V	For 5 volt microcontrollers
3V	For 3 volt microcontrollers
GND	Ground
CLK	Clock
DO	Data out
DI	Data in
CS	Chip Select
CD	Card Detect

SD CARD Formatting

The MicroSD Card will be formatted using FAT32 filesystems from the Arduino SD library which is directly compatible with the Auto-Logger's ATMega2560 chipset. The 32 bit system is compatible with the microcontroller. Using the Arduino library we can format the MicroSD card and see if it is formatted correctly. The chip select pin should be connected to pin 53 so the value in the library should be set to 53. The MicroSD card will be inserted into a sketch to test if the volume type is FAT32. There are several functions that are shown in Table 38 - ATMega2560 functions for SD Card memory management that can be used in the Arduino library when formatting the MicroSD card.

Table 38 - ATMega2560 functions for SD Card memory management

Function Name	Purpose
SD.exists(“example.txt”)	Returns true or false on whether a file exists
SD.remove(“example.txt”)	Deletes selected file
SD.mkdir(“/example”)	Creates a subdirectory
Seek()	This will move the reading/writing pointer to a new location
position()	This will return where in the file you currently are located.
size()	This will return the size of the file in bytes
isdirectory()	Returns true or false if a file is a directory/folder
openNextFile()	This will open the next file in the directory

MicroSD Card Interface

To speak between our MicroSD card and ATmega2560 MCU the Auto-Logger will utilize a Serial Peripheral Interface (SPI). SPI is a synchronous serial data protocol used by microcontrollers for communicating with devices over a short distance. There are four lines common to connect from the ATmega2560 to the MicroSD card reader MISO (Master In Slave Out), MOSI (Master Out Slave In), SCK (Serial Clock), and SS (Slave Select). The MISO line is the Slave line for sending data to the master. The MOSI line is the master line for sending data to the peripherals. The SCK relays the clock pulses which synchronize data transmission generated by the master MCU. The SS is the pin that the master uses to enable and disable specific devices such as additional storage or I/O components. When the SS pin is low the MicroSD card reader will communicate with the ATmega2560. Alternatively when the SS pin is high the MicroSD card reader will ignore the ATmega2560. This allows for multiple devices to be shared on the same MISO, MOSI, and CLK lines. The SS pin could be utilized during the transmission sequence. Enabling the SS pin will permit the MCU to dictate the MicroSD card module to send the specified data file directly to the Wi-Fi module. That is, only if the Wi-Fi module were connected to the MicroSD card as a slaved device. The SPI has three parameters for each device the speed, idle low or high, and shifted by MSB or LSB. For our MicroSD card it communicates at 25MHz. The data is shifted by the MSB first on the MicroSD. The third parameter is that the MicroSD card idles low. The SPI library will be used to transfer data from the microcontroller to the MicroSD card to be saved.

Remote Storage

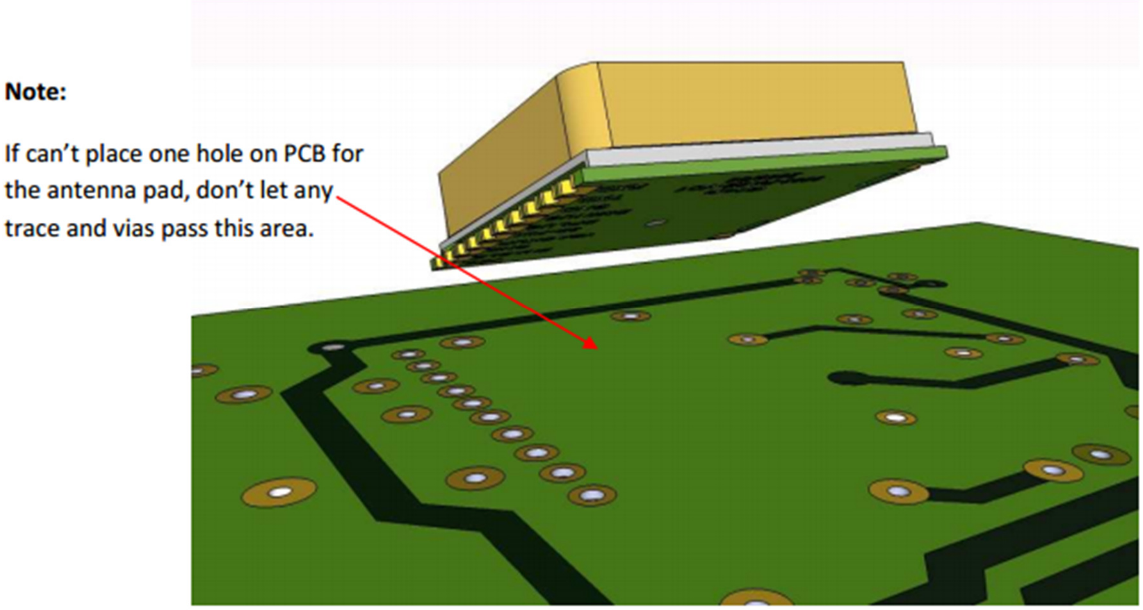
As recommended by the FSEC the Auto-Logger will call a web application and POST the recently collected data to a server hosted by FSEC. This server will be implemented using a mySQL database used to store and manage the dataset received from the Auto-Logger. All communications will be secured, sent through TCP port 443, and limited to authenticated users only. The server will require the capability to ensure received files have not been corrupted or altered in any way. The MySQL database provided by FSEC will be created such that these constraints and any additional concerns brought to the attention of either the FSEC faculty or the design team.

The Auto-Logger will act as a node application to save specified data files to the mySQL database in Cocoa Beach, FL. In order to efficiently communicate with this mySQL database, the Auto-Logger will excel at establishing a connection, pushing data, awaiting confirmation and managing the memory structure. In the event that some files have been corrupt on the server side, FSEC requests the Auto-Logger have the ability to keep track of all transferred data and only transmit what is needed by the server. Depending on the capabilities of the server the Auto-Logger may need to perform this action as a stand-alone device. The Auto-Logger should not continue to the next file to transmit until it has received confirmation from the server that the file was received and is in a functional condition.

Project Prototype Construction and Coding

PCB Assembly

Significant precaution must be taken when designing the PCB on which the FPGMMOPA6B GPS module will reside. A non-conductive via, or hole, 3.0mm in diameter must be placed beneath the GPS module for the antenna pad. If a non-conductive via cannot be placed beneath the module for any reason, the designer must instead prevent any traces or conductive vias to pass the area beneath the pack, as depicted below in Figure 19 - PCB Design constraint depiction for GPS Module.



Note:

If can't place one hole on PCB for the antenna pad, don't let any trace and vias pass this area.

Figure 19 - PCB Design constraint depiction for GPS Module (awaiting use approval)

Additionally, Radio Frequency (RF) signals propagating to or from the GPS module could interfere with the functional operation of the Wi-Fi module, or vice versa. In order to minimize this interference the modules will, to the best of their ability, not operate at the same time. Furthermore, the PCB should be designed such that the GPS and Wi-Fi modules do not boarder on another. There should be as much physical distance between these modules as possible.

For the purpose of precaution and desired reliability, all possible solutions have been analyzed for minimizing interference between the RF modules on this device. Comparison and analysis of plausible solutions are listed below in Table 39 - Interference Mitigation Techniques:

Table 39 - Interference Mitigation Techniques

Solution	Description	Pros	Con
1	Incorporate additional signal processing components on GPS antenna (LNA & SAW)	- Low Noise Amplifier further reduces noise from all acquired signals - Surface Acoustic Wave filter is a passive device - Saw Filters have high reliability and ruggedness	- Significantly increased PCB footprint due to Saw filter footprint - Additional power required for LNA implementation - No guarantee for interference reduction
2	Time Multiplexing	- Can manipulate transmission sequence to accommodate - No additional hardware necessary - Doesn't require increasing the devices dimensions - Greatest reduction in interference	- Signal re-acquisition times could impose on design specifications - Re-acquisition rates for Wi-Fi are much greater than GPS
3	Physical Separation	- Simplest to implement - Passive interference reduction -	- Most negligible impact on interference - Separation distance impact device dimensions - No guarantee for interference reduction

To provide a reliably robust solution to the possible interference problem, the Auto-Logger will implement both plausible options 2 and 3, Time Multiplexing and Physical Separation. Time Multiplexing is currently incorporated into the design already, through the use of Low Power modes on the Wi-Fi module and the Enable Chip pin on the GPS. These functions/features allow the Auto-Logger to “tag in” the RF module it desires. During typical operation and data acquisition periods the

Wi-Fi module will be placed into the low power mode “Light Sleep”. This mode provides power only to the main CPU and clock within the module, refusing power to the PLL CPU, Wi-Fi modem and crystal clock. Similarly, when the vehicle is identified as non-operational and residing on campus, the MCU will turn the Enable Chip pin low, thus disabling function of the GPS. Once the GPS has been shut down, the MCU will initiate the transmission sequence of the Wi-Fi module.

Final Coding Architecture

The coding architecture of the Auto-Logger is described in great detail by following the data flow chart depicted below as well as the accompanying explanation detailing each step within the chart. The many modules, chipsets, and memory storage will be managed by the ATmega2560 by relaying commands to them.

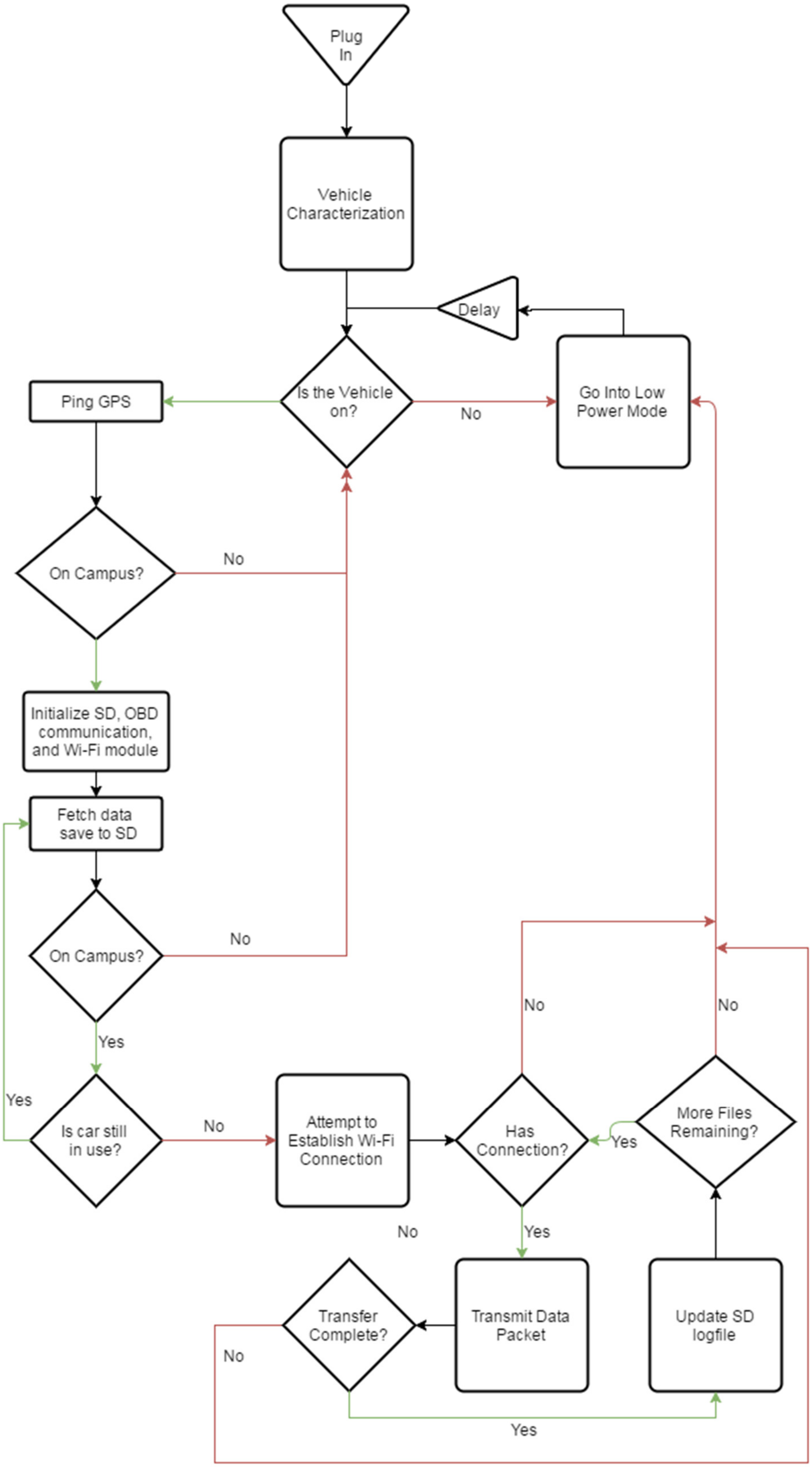


Figure 20 - Code Architecture Flow Chart

The final software architecture will incorporate a combination of PHP, C, and C++ to program the ATmega2560. PHP formatted commands are a necessity when communicating between the ATmega2560 and an Oracle based server. PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Many of the server side architecture will be handled by FSEC employees and is not completely within the scope of the architecture outlined above. The software on the Auto-Logger will only use PHP as a conduit to relay information to the server in manner that this specific server can understand and interpret. Any PHP code in a requested file is executed by the PHP runtime, which is located on the server side application.

Oracle implements an open source database relational database management system (RDBMS) named MySQL. FSEC has specified that they only have Oracle servers that will be receiving the data collected by the Auto-Logger. Because of MySQL's open source nature, documentation on how to communicate and implement the device will not be a large barrier to overcome.

The overwhelming majority of the Auto-Logger design and program flow utilizes C and C++. The multitude of open source libraries that are advised to use with the ATmega2560, Wi-Fi Network Interface Card (NIC), real time clock (RTC), and the chips within the OBD-II communications board, are all written in a stable form using a mixture of C and C++.

During first installation of the Auto-Logger it will retrieve data such as the VIN, identify what fuel type the vehicle uses, and which OBD-II standard communication protocol it uses to interface with to the vehicle's electronic control unit (ECU) such as CAN, VPW, ISO, etc. This makes sure the Auto-Logger does not use the wrong PID commands to retrieve the data from the vehicle.

The code-flow of the Auto-Logger attempts to activate Low Power Mode (LPM) whenever possible to decrease current draw from the vehicle's battery supply. By minimizing power draw at opportune moments, it will diminish thermal buildup in the separate ICs and power regulators on the PCB. This will increase the longevity of the hardware and allow FSEC to use the device for longer duty cycles subsequently gathering more data for their study. Furthermore, by ensuring the Auto-Logger uses LPM when the vehicle isn't in use the vehicle's battery will not reach the 20% depletion threshold when the driver leaves the device plugged in even if the vehicle is off.

The liquid crystal display (LCD) will be used to display to the driver what the Auto-Logger is doing. It will notify the driver when it starts logging data from the car, alert the driver when it had started saving his or her vehicle's location to the MicroSD. This way the driver knows when vehicle is being polled. The LCD can also be built upon on future iterations of the Auto-Logger to display useful information of the vehicle to driver or to notify him or her if the ECU detected a fault. Many of those features fall outside of the scope of the current design effort for this iteration but will be detrimental for future developers of this device if an LCD isn't ever incorporated into the initial design iteration. This feature also will be useful for the driver so that they are knowledgeable that any action taken while driving their car could be potentially recorded due to the vast amount of data that can be harvested from a vehicle's ECU through the OBD-II port.

The device must make sure that the vehicle is on campus before any data logging is performed because the FSEC specifically requested to operate as such. By keeping the GPS on low power mode and making intermittent pings to check if the vehicle had entered UCF campus grounds every second, data retrieval will begin to accurately and immediately store the data stream as soon as the vehicle crosses the campus boundary. This also shows that the device is indeed tracking the study participant's vehicle location at all times when the vehicle is in operation, but the privacy of the driver is kept intact because this location data isn't saved, in any form of permanent memory, on the FSEC database nor on the device's internal flash memory. The GPS location will be intermittently saved on temporary variables within the MCU to be compared with the predetermined campus bounds, saved within the MCU memory, to evaluate whether or not the participant's vehicle resides within the UCF.

Once the Auto-Logger has determined that the vehicle has indeed entered campus grounds, it will enable the MicroSD and OBD-II communication modules and prepare them to begin fetching data from the ECU on a second-by-second basis, as requested by the FSEC sponsors. The speed of the ECU on most vehicles permits the retrieval of 100 readings from the OBD-II port, through a serial to UART conversion interface, every second. A typical MicroSD card has a read and write data throughput of 4-6 MB/s which will easily save a single data point per second, as the typical cards have an average throughput of 4,050 KB/s which is more than enough to log multiple data points to a text file in one second.

The fact that the vehicle needs to use the UCF Wi-Fi to transfer data to the FSEC Oracle servers, the Auto-Logger requires the capability to detect when the vehicle is parked and off. This can be done through the implementation of several methods, the best of which are covered below.

The preferred method to track whether or not the vehicle is in use or not is by procuring the many available metrics from the ECU through the use of the OBD-II hexadecimal PID commands. These commands are all being utilized from Mode 1 PIDs so that the MCU can identify exactly when the vehicle is not in use with almost negligible time delay. PIDs 0C, 0D, 11, 1F, 5A, and 7F will be used to determine whether the car is currently being used by the driver. By using these metrics the Auto-Logger can be intelligent enough to determine if the driver had left the vehicle off and parked. This way the Auto-Logger can put itself into low power mode and keep itself from drawing too much power from the car battery. These readings such as vehicle speed can also aid in determining distance traveled and serve as a backup in the case that the GPS module has lost connection.

Table 40 - OBD-II Commands to Determine Vehicle Operation Status

Method	PIDs Used and Description	Pros	Cons
0	**00, 20, 40, and 60 to identify which commands below are available to use by the device**	**N/A**	**N/A**
1	0C to track RPM	Very precise indication if the vehicle is in use	Not all vehicles use this PID to track RPM (such as BMW)
2	0D to track speed over a period of time	Very easy to track time and speed and use calculation.	Similar to PID 0C
3	11 to track throttle position over a period of time	Similar to PID 0D	Cannot be used for any other metric other than engine on/off
4	1F to track how long the engine has been running since engine start	Should return 0 if the engine is not on	Not a commonly implemented by manufacturers
5	5A to track the accelerator pedal position	Can easily tell if the driver is within the vehicle if the pedal hasn't been used after a certain amount of time.	Cruise control may undercut this method if the driver uses it.
6	7F to track engine runtime	Similar to PID 1F	Very little documentation on this PID, very high chance that it will not be utilized by vehicle

The reason why the Auto-Logger will contain many OBD-II contingencies is because not all vehicles use every PID mandated by OBD-II. After first identifying which PIDs the vehicle can use by utilizing PIDs specified in Method 0 in Table 40 - OBD-II Commands to Determine Vehicle Operation Status, the Auto-Logger it will retrieve the data needed to determine which of the Methods can even be attempted. The methods are listed in .This makes sure the Auto-Logger does not use the wrong PID commands to retrieve the data from the vehicle. The methods are put into an order that depicts the most commonly to least commonly found on vehicle ECUs.

Many of the methods that depend on tracking a metric over a period of time will utilize the real time clock (RTC) embedded within the Auto-Logger PCB and will not be using the GPS time data gathered from satellite communication. The RTC determines time by using a crystal oscillator that generates a clock cycle that is sent to the onboard chip, DS1307. The DS1307 tracks how many seconds had passed since the UNIX epoch, in other words it tracks how many seconds since January 1st, 1970. The MCU will decode the time into year, month, day, hour, minute, and seconds and apply the correct conversion to into it to the data file, which is fetched from the ECU. The only inherent flaw to the DS1307 is that it does not account for leap year in any way.

The second method is through the use of GPS location tracking. The GPS module implemented in the Auto-Logger tracks many metrics such as, speed, longitude, latitude, and time; by utilizing these metrics over a period of time the Auto-Logger can determine if the vehicle is in use or not. If the vehicle had remained within the same coordinate value or the speed had remained 0 km/h over a certain period of time, the device can conclude that the vehicle is parked and not being used. The main downfall to this method is that GPS inherently contain an error margin of 10 meters. The car may always seem to be moving because of the shifting of coordinates being detected by the Atmel MCU. Additionally, the other problem with this method is that connection can be lost due to inclement weather or when driving within parking garages. The device wouldn’t be able to differentiate gridlocked traffic from being parked, and may switch to LPM and thus lose potential data points. Because of these errors not being able to be easily mitigated, this method will only be used as a redundant backup in the case that the vehicle does not utilize any of the OBD-II methods defined previously.

Polling of the UCF network is critical for the establishment of a reliable connection with the FSEC Server. As depicted in the received signal strength indicator (RSSI) is utilized to hold a tight threshold on the signal strength. To ensure a reliable connection, the Auto-Logger will enforce a threshold of -80dBm. This threshold will guarantee accurate packet transfer and a drastic reduction in packet loss leading to repeated transmissions.

RSSI is a measurement of how well a device can hear a signal from an access point or router. Not to be confused with the associated transmit power of the access point or router, the RSSI is a value useful for determining the strength of a specified signal you can detect. The ESP8266EX will rate the specified signal, UCF WPA-

2, in our case, on scale from 0 to 255 where the higher the number the strong the signal is. After the device determines that the RSSI is perfect for creating a signal threshold to establish a reliable connection it will determine that it can transfer the data without risk of data loss. The Auto-Logger will also log the last file that had been successfully transferred without loss of connection during transfer.

The MicroSD module will be saving the data points as separate data files in .txt format. They will be organized in chronological order of data retrieval and be name of each file will be the time that it was saved in the following format:

yymmddhhmmss.txt

This way when the data is sent to the FSEC server it will automatically be in chronological order. Initially the Auto-Logger saves the vehicle ID in the form of VIN onto a file on the MicroSD as stated previously by using the OBD-II PID Mode 9. The data will be transmitted to a folder on the FSEC server that matches the VIN retrieved from the ECU. The commands written within the Auto-Logger will be following PHP communication protocols with the server to send the data file. The code written to command the MicroSD via the MCU will be written in C++ due to the fact that many open source libraries that interface with MicroSDs are written in that particular coding language.

Even if connection is lost in the middle of data packet transfer Wi-Fi, the Auto-Logger will know what the last file successfully sent was because the data filename will be saved onto a log file stored within the MicroSD. That way the next time the Auto-Logger connects to the UCF Wi-Fi it will know which file to start transferring from within its memory. One of the stipulations to the project stated by FSEC is that they didn't want their server to be flooded with duplicate datasets, by implementing a log file onto the MicroSD, it prevents such a situation to occur.

After the Auto-Logger transfers the data packet it will scan through the MicroSD to determine if it contains anymore data files to send to FSEC Oracle servers. If there are any remaining files that haven't previously been sent it will attempt to do so. If there aren't any left it will end transmission and drop connection from the UCF network and put the Wi-Fi NIC into sleep mode

If the Wi-Fi Network Interface Card loses connection at any point during transfer it will end transmission and the Auto-Logger will then go into LPM. The reason why the device will not try to connect to the Wi-Fi provide by UCF campus grounds again is because it had already calculated RSSI signal strength and doing it again will be redundant because signal connection was already lost, proving that the RSSI should also return that the signal strength is weak.

After all data is sent and the Wi-Fi NIC had properly dropped connection and put itself into sleep mode the Auto-Logger will turn off power going to the MicroSD, LCD and put the GPS into sleep mode. After that, the MCU will put itself into sleep mode and intermittently poll the ECU over UART connection with the OBD-II communication module using one of the aforementioned methods. If the Auto-Logger had previously determined that none of the OBD-II methods are available then it will use the GPS data to determine when the car is turned on. These methods need to be used to check if the car is on because power is always supplied to the OBD-II port via the vehicle's lead acid battery. The vehicle also doesn't have any signal in the OBD-II port that trips once the vehicle is turned on.

Prototype Testing

Hardware Specific Testing

LCD

To test the GDM12864HLCM LCD module for proper functionality, the first step would be to test the PCB traces to ensure there were no mishaps during fabrication or post-manufacturing stress testing. This can be accomplished by connecting the LCD module to the ATMega2560 chip on a standalone breadboard. The ATMega2560 chip will transmit a simple "Hello World" program to display on the LCD screen. The data needs to be sent to the corresponding pins on the ATMega2560 chip in order to display the message in the proper sequence. This test program will clarify whether or not the pins on the LCD module are properly connected to the ATMega2560 chip as well as verify there were no oversights during fabrication. Additionally, iterative analysis must be performed on the LCD module to ensure all LCD segments are properly illuminating and none of the segments are dead or fail to illuminate, this will verify that the module was in no way damaged during shipment or handling. This test can be performed in a similar fashion to the first, however every available character will need to be displayed on every writable LCD segment to verify the functional operation of the GDM12864HLCM LCD module.

Vibrational Durability

Typical operation of a motor vehicle, depending on the quality of your suspension, subjects the vehicle to significant vibrational forces. Whether driving off the beaten path on unpaved roads, or traversing through the city the vehicle, including and its passengers and peripherals will feel the force of these vibrations. The severity of a vibration can be characterized as vibration amplitude. This measure can be quantified in several ways, including peak-to-peak level and peak level. The first is valuable for determining the critical or maximum stress of a machine part. In contrast, the latter is valuable for indicating the level of short duration shocks, however it doesn't not account for the time history of the vibrational wave. The most valuable measurement would be the RMS value as it gives an amplitude value which is directly related to the energy content and there for the destructive abilities of the vibration.

For the purpose of testing the structural integrity of the Auto-Logger, the RMS value of the vibrational force will be measured to ensure the connections between the internal components and the device package are able to withstand nominal forces, and more importantly the fastening of the package to the vehicle cabin itself. This is important to the project design, as a dislocation of the Auto-Logger could result in device, vehicular or personal damage as well as data inconsistencies which lead to errors.

Temperature Resistance

Cars get extremely hot after sitting in the sun all day long, especially in the Florida summer sun. The Auto-Logger design assures the device and all its internal components can withstand the harsh temperature the vehicle cabin may face on a daily basis. To guarantee the resilience of each component and the device as a whole mild temperature stress testing will be conducted. Temperatures as hot as 75° Celsius will be tested to ensure proper functionality.

Not only are peak and valley temperatures important, but the rate at which these temperatures fluctuate are also crucial to project design. Crystal clock frequency rates are prone to experience frequency drift as the temperature of the quartz crystal changes. Higher rates of change result in large frequency drifts, which could lead to a miscommunication between the MCU and its peripherals or between the Auto-Logger and the FSEC server. While measures have been implemented to avoid these transitional periods, not every scenario can be planned for. As such the design team must test the device and document the results for possible errors on inconsistencies caused by rapid changes in temperature.

Climate Impacts

As stated previously, Florida weather can be harsh and relentless from blazing heat to monstrous rains. Correspondingly, Florida humidity can play a critical role in the functional operation of the Auto-Logger. For the purpose of this study, the Auto-Logger will be operational in the UCF region where daily humidity levels average around 74%. Many components incorporated into the Auto-Logger are designed such that humidity and other atmospheric factors play no role in the functionality of that component. However, some components such as the GPS module specifically require high humidity and condensation protection. To this end, the Auto-Logger packaging will undergo testing to analyze the device's resilience to the penetration of humidity and the accumulation of condensation.

Software Testing

Wireless Access

The Auto-Logger must have the ability to properly communicate with a specified access point in order to gain access to the internet to establish a communication channel with the FSEC server in Cocoa Beach, Fl. If a connection cannot be establish, the FSEC will be unable to collect data until the end of the study. This will create significant delays in the processing and utilization of the information gathered and would be considered a malfunction of the product.

In order to test and troubleshoot the network protocol for proper implementation a dummy network will be utilized to test the adequacy. Once a stable connection can be establish, network security architecture can be introduced. This would be the last capability required to connect to our desired network. During the testing and troubleshooting process, the network should be disconnected at each state in the connection sequence to ensure the system does not “get lost” within any infinite loops or logical fallacies. A proper response from the system should be an eventual timing out of the system, which would then inform the MCU the ESP8266EX Wi-Fi module should be returned to low power operation or the entire system shut down for power conservation.

Electric Vehicle Testing

The design team will be required to test and troubleshoot the OBD-II data acquisition of the Auto-Logger on the FSEC provided electric vehicle, a Nissan LEAF. As this is the only vehicle the design team will have regular, legal access to it will be the only EV make/model which the Auto-Logger supports. Currently, electric vehicles are not required to follow the OBD-II standards and regulations because they are not a typical ICE vehicle.

When the Auto-Logger is implemented on an EV there are significantly different parameters for which it is responsible. Additionally, these readings may stem from different pins, different interface codes or they could possibly not even be supported, such as cabin ambient temperature. The design team will work hand-in-hand with the FSEC in order to establish a set schedule for continuous testing, troubleshoot and solution implementation. This process could prove tedious depending on the availability of interface documentation and testing periods.

Internal Combustion Engine Testing

Due to the across-the-board standardization and continuous regulation of the OBD-II interface for internal combustion engine vehicles, the design team will be able to test and troubleshoot the Auto-Logger on their personal vehicles. Testing will be conducted on the three makes and models listed below:

- 2001 Mitsubishi Eclipse
- 2004 Honda CR V
- 2016 Honda RAV4
- 2001 Ford Explorer Sport Trac

The 2016 Honda RAV4, since it was manufactured after 2008, will serve as a good test reference for the ISO 15765 CAN. The Honda CR V and Mitsubishi Eclipse will serve as a good test reference for the ISO 14230/KWP2000 protocol. The Ford Explorer can test the OBD-II protocol SAE J850 PWM which is only implemented by the Ford Motor Company. With these vehicles the team can test most of the protocols before 2008, however the design team needs to find a Chrysler model to be able to test all ICE OBD-II protocols with the Auto-Logger. As requested by the FSEC the Auto-Logger will produce information logs gathered either from the OBD-II interface, or from the implemented retrieval components such as the GPS module and temperature sensor, for vehicle location, speed, engine load, fuel economy, and fuel level.

Server Communication

Wireless communication is difficult, many things can go wrong causing the entire system to fail. In order to assure the FSEC the Auto-Logger is a reliable autonomous logging device, the team must test and troubleshoot the system on a number of known fault conditions. In addition to these known faults, the design team may run into unspecified or experience issues. More time and effort will be required to resolve these issues, as the team must first identify with certainty which device is creating the problem, whether it be the client or server. In the event that the issue does not stem from the client device (the Auto-Logger), the FSEC will need to work with the design to team to resolve these server-side issues.

Through the rigorous tests and trials, the design team will present to the FSEC evidence of the Auto-Logger's ability to successful and autonomously perform the following tasks:

- Proper Low Power initiation
- Connection Establishment
- Accurate RSSI evaluation
- File Transmission
- Successful data interpretation
- Server receipt confirmation
- Logging of successful transmission on client
- Proper module shutdown if any time-outs occur

While some of these tasks may seem to lay within other systems of the Auto-Logger, such as Logging of successful transmission on client, these tasks are critical to the proper function of the wireless communication system, and as such, will be comprehensively evaluated to provide the FSEC with ease-of-mind that the data acquired will most certainly be attained.

GPS Acquisition

Accurate GPS location acquisition is key to producing a function Auto-Logger. Rigorous testing and analysis will go into guarantee all information received by the GPS and communicated with the MCU will be reliable and free of error. Prior to incorporation into the device, the GPS module will go through a robust testing sequence initiated through Secure Shell(SSH) HyperTerminal on a standard PC. The HyperTerminal will be capable of testing the following GPS parameters:

- Serial Communication
- Satellite Acquisition
- GPS localization
- Date
- Time

Once the GPS module successfully completes SSH testing to ensure all of its pins and features are operational, the design team will need install and test the module once again. The second round of testing will be rigorous, as the team will be able to travel with the device, observing and tracking changes in GPS localization, speed and distance. The combination of these testing scenarios will give the design team the confidence to guarantee the Auto-Logger can successfully perform the following tasks.

- Accurate Latitude/Longitude Acquisition
    - Log while on campus
- Accurate time metrics
- Proper time zone conversion
- Successful localization within campus bounds
    - Recognize when to and when not to log data
- Proper shutdown once trip concludes

As previously stated in Temperature Resistance testing, the accuracy of the crystal clock employed by the GPS will require testing to ensure errors do not occur when the vehicle is first start while the cabin is cooling to room temperature from a typical standing temperature of 70° Celsius.

SD Card

The TS256M MicroSD Memory Card module will be tested for correctly writing and reading data. The test will be conducted in the Arduino library. First we request the card info and make sure the chip Select variable is set to the correct location, pin 53. To test if the MicroSD card can open a file and write to it by using a sketch from the Arduino library. The sketch is a program that opens a test text file from the MicroSD card and writes some text into the file. To confirm that the file was correctly opened and written, open the file and see if the test text is located inside.

Automation Testing

Hardware

Some aspects of the hardware testing and troubleshooting process could be automated through the use of simulation software such as the temperature and vibrational analysis. Circuit analysis and design will be conducted solely through the use of simulation and design software such as EagleCAD and Xilinx.

Software

While all software would have to be conduct through the use of automated system such as compilers and libraries additional automation can be implemented through the use of hardware and network simulators for troubleshooting field issues without having to be in the field. Many of the enigmatic situations can be simulated and replicated on a computer using one of the various tools available to students and professional engineers in order to gain a better understanding and find a solution to the underlying problem.

Administrative Content

Milestone Discussion

Spring 2016
JANUARY 11, 2016 – MAY 2, 2016 (ROUGHLY 15 WEEKS)

Week 1 (Jan 11 - Jan 17)

- Completion of the *Independent Project Identification* Document. *
- Initial project research.

Weeks 2 - 5 (Jan 18 – Feb 14)

- Conduct research for the project to form a "body of knowledge". This research can be divided in the following manner: *
    - Real products available on the market which are relevant to the design of the Auto-Logger.
    - Pertinent hardware (Microcontrollers, Communication, etc.) discovered from prior projects and products.
    - Software architecture, implementation methods, and testing procedures that relate to the Auto-Logger.
- Completion of *Divide and Conquer – Initial Project Documentation*
- A transition from general project research (i.e. possible microcontrollers/modules) to more in-depth analysis (i.e. exact parts) will occur.

Week 6 - 10 (February 15 - March 20)

- Select parts and conduct extensive research, documenting all findings.*
- Each member begins to write their respective portion of the Final Documentation.

Weeks 10 - 12 (March 21 - April 10)

- Each member contributes to their respective portion of the Final Documentation, researching and collaborating when necessary.
- By the end of week 12, the Final Documentation will be written in its entirety, only subject to slight revisions thereafter.*

Week 13 - 15 (April 11 – April 28)

- All final revisions are completed during the last week.
- The Final Documentation is completed no later than April 28th.*

Summer 2016
MAY 16, 2016 – AUGUST 1, 2016 (ROUGHLY 10 WEEKS)

Weeks 1-4 (May 16 - June 12)

- Completion of a working prototype (Communication, Data Logging, Syncing). *
- Initial Internal Combustion Engine testing & documentation of prototype.
- Establish Electric Vehicle testing dates with FSEC

Weeks 5-8 (June 13 - July 10)

- Begin designing final version of the system.
- Continued testing/troubleshooting and documentation.
- Travel to Florida Solar Energy Center to perform EV log testing

Weeks 9-10 (July 11 - July 25)

- Final development and testing conducted to ensure proper functionality of the autonomous log and upload system.
- By the end of week 9, the project is declared finished as a working unit. *
- Any pertinent documents written during weeks 1-10 of Summer 2016 is now integrated into the Final Documentation*
- All required documentation is to be reviewed and finalized prior to July 25, 2016.
- The Final Presentation is constructed, revised, and delivered.*

NOTE: (*) denotes a Project Milestone

Budget and Finance Discussion

The Florida Solar Energy Center, alongside the University of Central Florida, have solicited for a senior design group to undertake this project with the promise that they, the FSEC, would pay for the entire project. Unfortunately, the design team must first purchase the materials list and, once the parts have been received, must present the Bill of Materials to the FSEC for reimbursement. According to the FSEC reimbursement for the Bill of Materials should arrive almost 2 months after submission. In order to cover the initial costs, before reimbursement, the team will split the total cost between themselves. It is possible the team could fundraise or look for additional project sponsors who might be willing to donate before the final purchase is made. This Bill of Materials is depicted below.

Device	Quantity	Price	Total
Custom PCB Fabrication	10	$140.00	$140.00
Atmel ATMega2560-16 AU	2	$16.55	$33.10
ESP8266EX Wi-Fi Module	3	$5.00	$15.00
OBD-II Connector(Male)	2	$4.00	$8.00
Crystal Oscillator	3	$1.50	$4.50
FGPMMOPA6B GPS Module	1	$21.25	$21.25
TMP36 Temperature sensor	5	$1.50	$7.50
TS256M MicroSD Card Module	1	$14.50	$14.50
LM317T Adjustable Voltage Regulator	10	$0.44	$4.40
Gain Antenna	1	$7.50	$7.50
STN 1110 OBD to UART Interpreter Chip	1	$9.99	$9.99
MCP2551 High-Speed CAN Transceiver Chip	3	$1.02	$3.06
GDM12864HLCM Liquid Crystal Display Module	1	$19.95	$19.95
LP2981-N Ultralow Dropout Regulator	1	$0.19	$0.19
MM74HC 4050M Hex Inverting Logic Level Down Converter	1	$0.45	$0.45
DS1307 Real time Clock	1	$3.89	$3.89
Passive Components	N/A	$40.00	$40.00
CR2032 Battery	1	$1.00	$1.00
ATS16B 16MHz Crystal Oscillator	2	$0.36	$0.72
LP2985-33DBVR precision 3.3 volt regulator	5	$0.55	$2.75
Total Shipping Cost Estimate	N/A	$50.00	$50.00
Total:			**$387.75**

Project Summary

Senior Design group #13, consisting of three Electrical Engineering students Hassan Siddiqui, Justin Wright and Zachary Ross, set forth to satisfy the needs of this sponsored solicitation. Their solution, titled The Auto-Logger, will autonomously interface with any internal combustion engine based vehicle as well as a single model of Electric Vehicle, track vehicle localization through GPS and time taken for the vehicle operator to complete various actions as well as some parameters associated with the vehicle ambient temperature and transportation speed. We take pride in the opportunity to work hand-in-hand with and develop a product for The University of Central Florida as they, along with the Florida Solar Energy Center, will be utilizing the data collected by the Auto-Logger to improve both the University and the surrounding community.

Appendix A – Copyright Permissions

All diagrams and figures present in this document are original creations by the Auto-Logger design team unless noted in the figure description.

Appendix B – Data Sheets

ESP8266EX Wi-Fi Module:
https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf

TI CC3100MOD Simple Wi-Fi Module:
http://www.ti.com/lit/ds/symlink/cc3100mod.pdf

TMP36 Temperature Sensor:
http://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf

FGPMMOPA6B GPS Module:
https://cdn-shop.adafruit.com/datasheets/PA6B-Datasheet-A07.pdf

Copernicus II GPS Module:
http://cdn.sparkfun.com/datasheets/Sensors/GPS/63530-10_Rev-B_Manual_Copernicus-II.pdf

Real Time Clock Datasheet:
http://datasheets.maximintegrated.com/en/ds/DS1307.pdf

LCD Datasheet:
https://www.sparkfun.com/datasheets/LCD/GDM12864H.pdf

MICROSD card Datasheet:
http://download.mikroe.com/documents/accessories/sd-cards/microsd-card-transcend-datasheet-v100.pdf

Battery CR2032 Datasheet:
http://www.farnell.com/datasheets/1496885.pdf

ATmega2560 datasheet:
http://www.atmel.com/images/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

Appendix C - Resources

1) http://playground.arduino.cc/Code/FTP
2) http://www.speedguide.net/faq/how-does-rssi-dbm-relate-to-signal-quality-percent-439
3) https://www.arduino.cc/en/Reference/WiFiRSSI
4) ftp://imall.iteadstudio.com/Basic_Module/IM141118001/0B-Beginner%20Guide%20to%20ESP8266_V0.4.1.pdf
5) https://cdn-shop.adafruit.com/datasheets/ESP12FCCtestreport.pdf
6) http://www.ti.com/product/cc3100
7) http://www.analog.com/en/products/analog-to-digital-converters/integrated-special-purpose-converters/integrated-temperature-sensors/tmp36.html#product-overview
8) https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor
9) http://www.4timing.com/techoscillator.htm
10) http://www.torexsemi.com/products/voltage_regulators/data/12-Basic_Knowledge_of_LDO_Voltage_Regulator_En.pdf
11) http://www.gpsinformation.org/dale/nmea.htm
12) https://forum.sparkfun.com/viewtopic.php?f=13&t=26845
13) https://www.adafruit.com/products/746
14) http://computer-solutions.co.uk/download/tcp_tutorial.pdf
15) https://en.wikipedia.org/wiki/File_Transfer_Protocol
16) http://www.microchip.com/forums/tm.aspx?m=326844&mpage=1&key=&#326844
17) https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer&hl=en
18) http://www.navsea.navy.mil/Portals/103/Documents/NSWC_Crane/SD-18/Test%20Methods/jstd020a.pdf
19) http://serverfault.com/questions/501025/industry-standard-for-minimum-wifi-signal-strength
20) http://www.rohsguide.com/
21) http://standards.ieee.org/about/get/802/802.11.html
22) http://www.tutorialspoint.com/wi-fi/wifi_ieee_standards.htm
23) https://espressif.com/en/products/hardware/esp8266ex/resources
24) https://www.law.cornell.edu/wex/fifth_amendment
25) http://cdn.sparkfun.com/datasheets/Widgets/stn1110-ds.pdf
26) https://www.sparkfun.com/datasheets/DevTools/Arduino/MCP2551.pdf
27) https://cdn.sparkfun.com/assets/learn_tutorials/8/3/ELM327DS.pdf
28) https://www.sparkfun.com/datasheets/Widgets/OBD-II-UART-v13.pdf
29) https://www.sparkfun.com/datasheets/Widgets/ELM327_AT_Commands.pdf
30) http://www.gizmology.net/batteries.htm
31) http://iosix.com/OBD-II%20Datasheet.pdf
32) http://www.hemdata.com/products/dawn/obd-mini-logger

Appendix D - Other

Tables

Figures

Equations