Vehicle Monitoring System

By

Caleb Perkinson; perknsn2

Ishan Ahuja; iahuja2

Samuel Utomi; sutomi2

Final Report for ECE 445, Senior Design, Spring 2016

TA: Cara Yang

5/4/2016

Project No. 3

Abstract

Our goal with this project was to design a Vehicle Monitoring System (VMS) that can read characteristic data about how a car is being driven to create a skill profile of the driver. This is accomplished by reading data from the On Board Diagnostics (OBD) port on the car, along with several sensors to provide a detailed view of driving abilities. We currently transfer this data to an online server over Wi-Fi using HTTP commands. The data is stored on the server and can be viewed using a web interface that automatically updates every 5 seconds to show the most current data. The entire device is contained on a single printed circuit board and requires no intervention from the user after initial setup. The device has an auxiliary battery which is used to monitor data when the car is off.

Table of Contents

Abstract 1
1. Introduction 1
1.1. Statement of Purpose 1
1.2. Objective 1
1.2.1. Benefits and Features 1
1.2.2. Goals and Functions 1
2. Design 2
2.1. Hardware 2
2.2. Software Flowchart 3
2.3. PCB Layout 4
2.4. Power Circuits 5
2.4.1. Buck Converter 5
2.4.2. Battery Circuit 6
2.5. Data Acquisition and Transmission 8
2.5.1. Accelerometer 8
2.5.2. Gyroscope 8
2.5.3. OBD-II Transceiver 8
2.5.4. GPS 9
2.5.5. Wi-Fi 9
2.6. Web Server 9
3. Design Verification 10
3.1. Accelerometer Sensor 10
3.2 Gyroscope Sensor 10
3.2. Microcontroller 11
3.3. OBD-II Transceiver Module 11
3.4. Buck Converter Circuit 11
3.5. Battery 12
3.6. Power Regulator Circuit 12
3.7. Software 12
4. Costs 13
4.1. Parts 13

4.2. Labor 13
5. Conclusion 14
5.1. Accomplishments & Uncertainties 14
5.2. Ethical Considerations 15
5.3. Safety 16
5.4. Future Work 17
References 18
Appendix 19
A. Requirements and Verification Table 19

1. Introduction

1.1. Statement of Purpose

The purpose of our product is to allow for constant monitoring of characteristic data about a vehicle and make it accessible in an easy way. This monitoring will allow managers of large fleets to easily keep track of their vehicles while also being able to quantify their driver's safe driving abilities. The product will allow for a much leaner fleet through increasing productivity by making it simple to utilize fixed resources at their maximum capacity. It will also help improve the image of the organization by promoting safe driving.

1.2. Objective

1.2.1. Benefits and Features

- Encourage safe and efficient driving behavior in fleet drivers
- Improve fleet productivity
- Keep drivers accountable
- Make fleet data easy to view

1.2.2. Goals and Functions

- Monitor:
  - Fuel levels (from OBD II messages)
  - Speed (from OBD II messages)
  - Acceleration & breaking (from accelerometer and gyroscope)
  - G-force while turning (from accelerometer and gyroscope)
  - Location (GPS) (from GPS chip)
  - Vehicle alerts (from OBD II messages)
  - Miles driven (from OBD II messages and GPS)
- Access recorded data through online web page
- Provide derived metrics such as
  - Miles driver over speed limit
  - Driver fuel efficiency
  - Quick or unsafe turning
  - Ignoring vehicle error messages
- No driver intervention necessary once plugged in

2. Design

2.1. Hardware

The high-level block diagram of the Vehicle Monitoring System is shown in Figure 1. This device is powered from the 12V rail on the OBD-II port that is available on any car manufactured after 1996. The 12V is filtered and stepped-down to charge the battery and also supply power for the peripherals that implement the system. OBD-II messages are also collected from the OBD port and passed to the microcontroller to be sent to the server, along with other vehicle data, using the Wi-Fi module.

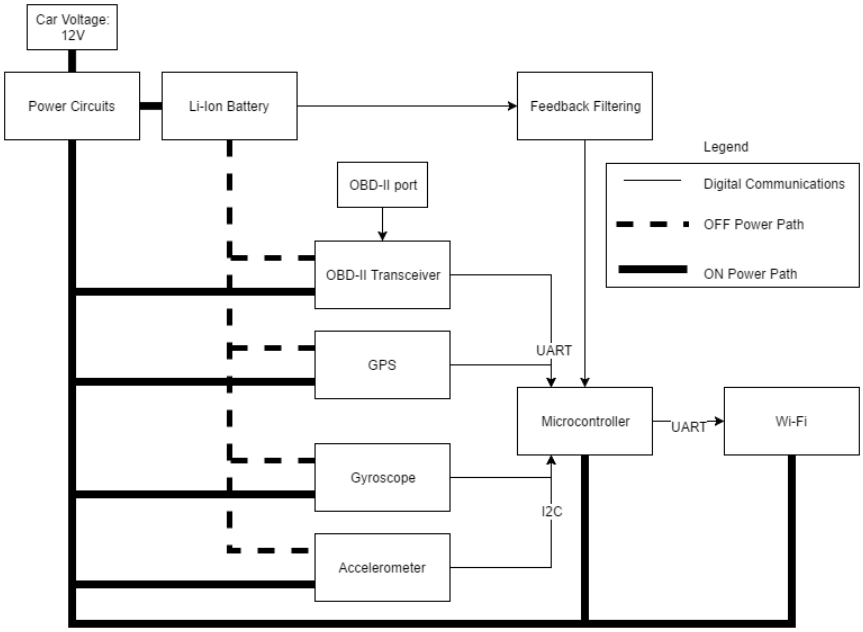


Figure 1. System Level Block Diagram.

2.2. Software Flowchart

Figure 2 shows the software data paths in our device, along with the communication protocols used at each interface. We had 3 devices that communicated using UART, which meant that we needed to use a multiplexer to switch between them since our microcontroller only had one UART input. We also had two devices communicating over I2C and all server communications were using Hyper Text Transfer Protocol.

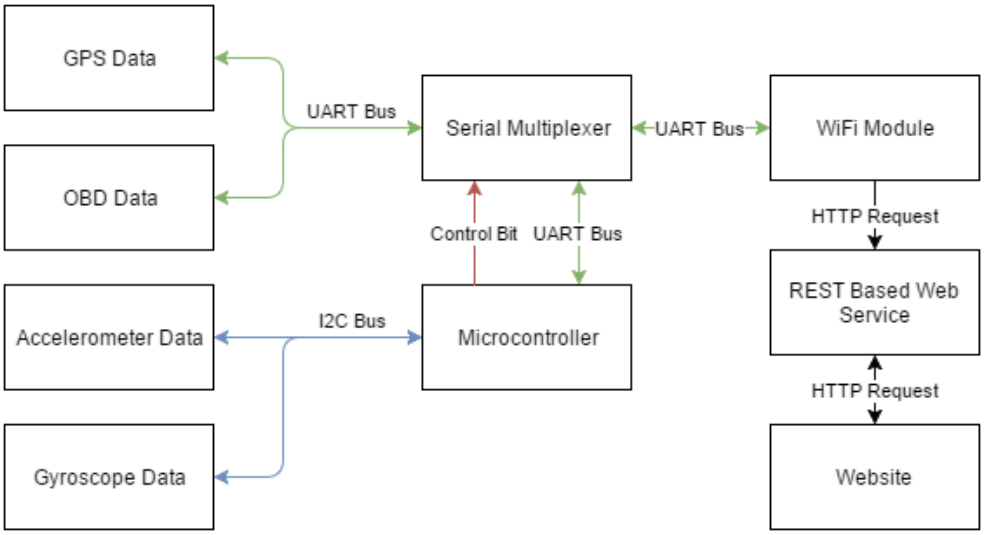


Figure 2: Software Data Flowchart

2.3. PCB Layout

Laying out the board was a fairly involved process due to the number of components and required separation of analog and digital components. In order to reduce the effect of high frequency digital signals coupling with analog signals, the digital and analog components were placed in separate areas, as shown in Figure 3. Decoupling capacitors were used extensively to ensure that fluctuations in the power supply or electromagnetic interference would not affect the devices in an intolerable way. Minimum trace lengths were used for the I2C bus due to the capacitance requirement for I2C buses. Devices that had antennas were placed as far on the outside of the board as possible to ensure the signal could propagate as intended and also not affect other components on the board.

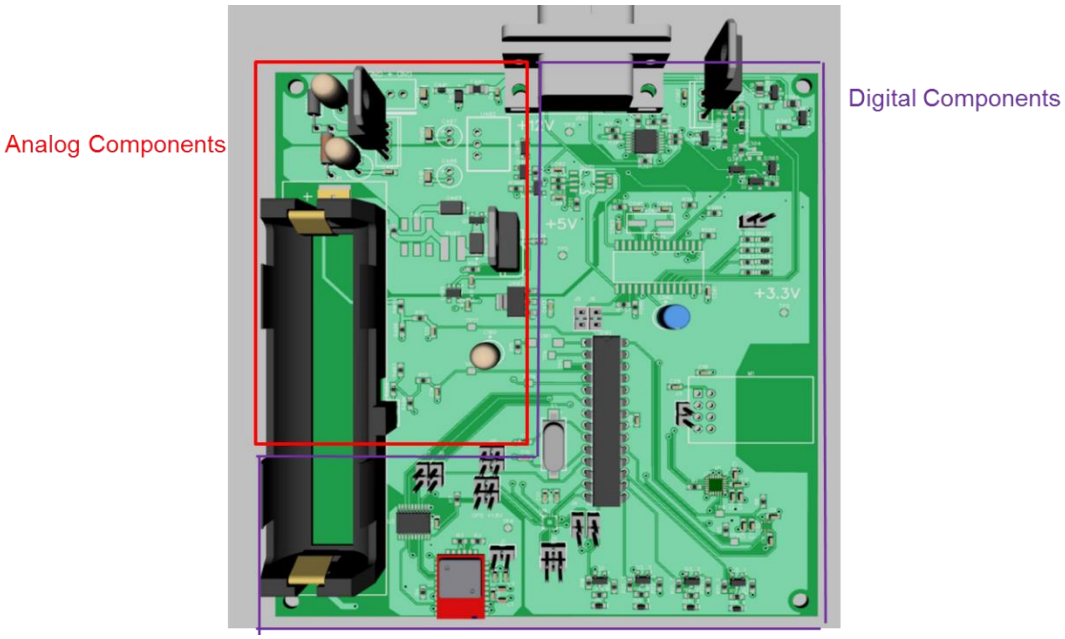


Figure 3: PCB Layout

2.4. Power Circuits

2.4.1. Buck Converter

We used a buck converter to step down the 12V supplied by the car to 5V. We chose to use a switching regulator because we desired high efficiency since most of are modules (e.g. accelerometer, gyroscope) drew current from the 5V line [3].

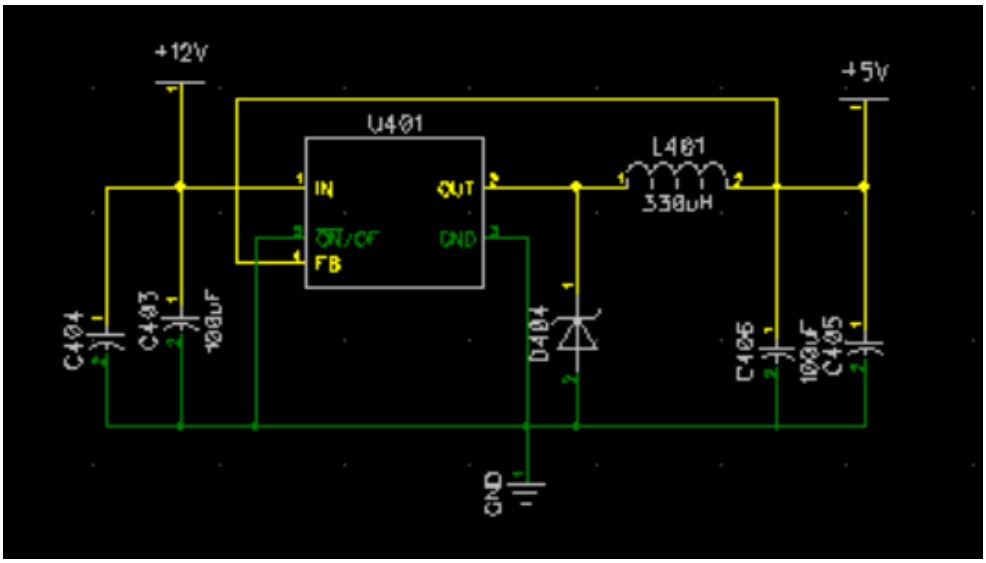


Figure 4: Buck Converter Circuit

The datasheet for the chip provides a guideline which is used to select the inductance. This guideline specifies the recommended inductance based on the expected input voltage and the maximum expected load current. Knowing that our current consumption would be less than 0.8A, we used this as our maximum expected load current and knowing our input voltage to be 12V, determined the inductance of our inductor to be 330uH [6].

The feedback pin (FB) on the chip monitors the output voltage and adjusts the switching frequency of the internal oscillator in order to maintain a 5V output voltage.

In order to show proper functionality of the buck converter, the oscilloscope capture of both the input and output voltage of the buck converter is shown. As can be observed, the output of the buck converter is not only maintained at 5V but also has a low ripple (around 400mV). A minimal voltage ripple is important because it ensures stable conditions in our modules which feed from the 5V line [2].

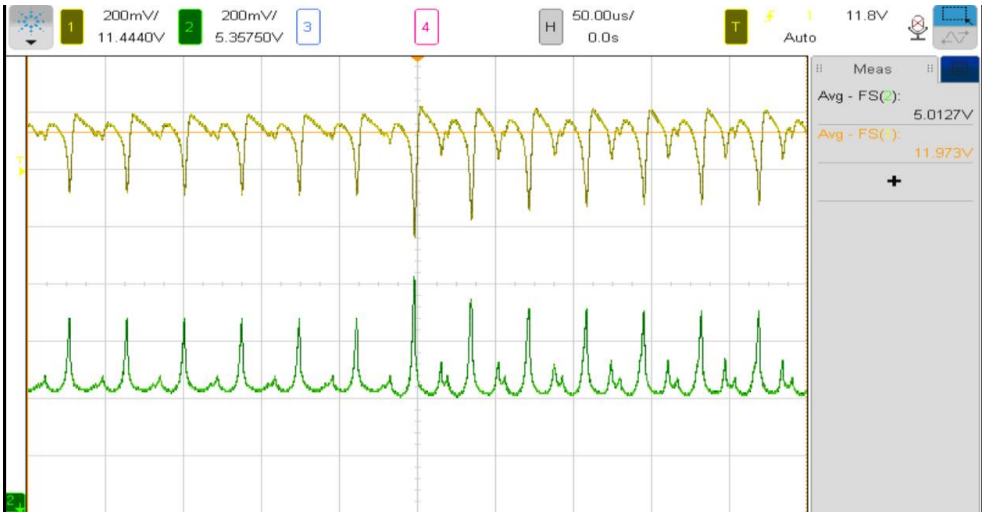


Figure 5: 12V to 5V Conversion

2.4.2. Battery Circuit

One of our device requirements, was for it to be able to run without the need for the car battery. Thus we decided to integrate a rechargeable battery in our design. The main requirement for the battery was to ensure that it could power our device by itself for longer than a day.

We eventually decided on the 18650 lithium ion battery which could be charged up to 4.2V and had a battery capacity of 3600mAh. When the car is off, only the following devices are in use:

- Atmega microcontroller: 15.5mA (awake mode)
- GPS module: 31mA (tracking mode)
- Wifi module: 80mA (operating current)

Thus the total current consumption when the car is off will be around 126.5mA. This means it discharges at 0.035C rate. The battery should be able to discharge current for 28.4 hours which satisfies our requirement.

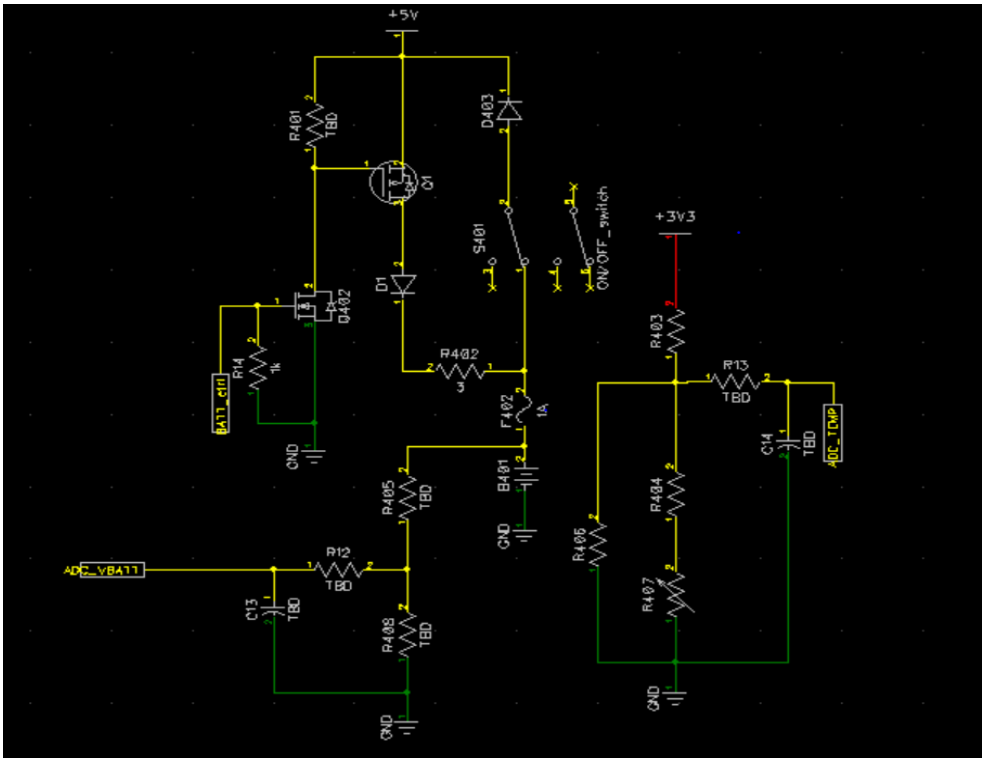


Figure 6: Battery Charging Circuit

The charging circuit shown above, allows for the charging and discharging of the battery as well as cutting off the charging path when the battery has reached its maximum voltage [4]. In order to achieve this functionality, we desired a constant voltage on the source side of the MOSFET which opens or closes the charging path. This is because turning the MOSFET ON/OFF depends on the difference between the gate and source voltage. So our initial thought of using just an NMOS was not sufficient as the voltage on the source side of the NMOS lay on the battery side which varied. Thus we decided to use a combination of the PMOS (Q1) and NMOS (Q402). Our circuit works as follows:

- BATT_ctrl = 0V: the NMOS stays off and there is no current flow through the resistor (R401). Thus there is no voltage drop through the resistor and the PMOS remains off (no charging)
- BATT_ctrl = 5V: The NMOS turns on and current flows through the pull down resistor to ground. There is a voltage drop and thus a difference between the gate and source voltage of the PMOS; this turns it on and creates a charging path for the battery.

The BATT_ctrl is an analog signal (0v/5v) that is sent from the microcontroller. The diode (D1) prevents reverse current flow and the resistor (R402) is present to control current flow. In order to monitor temperature, we used a negative coefficient 470kΩ thermistor (resistance drops as temperature increases). This is necessary as a safety precaution to shut off the charging path in the event of overheat due to a short circuit or excess charging [1]. From our battery specifications, we know the upper temperature limit is 40°C so by knowing the resistance of the thermistor at that temperature we can shut off the charging path when that condition is met.

$$\frac{Rt}{R25} = 0.46351 \text{ @ } 40°C$$

$$R25 = 470k\Omega, \;\; Rt = 470k * 0.46351 \approx 218k\Omega$$

2.5. Data Acquisition and Transmission

2.5.1. Accelerometer

The accelerometer used for this project will be the MMA8653FC manufacture by Freescale. This is a low-power device that talks with the microcontroller over I2C. The device is extremely flexible allowing for programmable interrupts, which will be used to detect when the vehicle is off and suddenly moved, potentially indicating that it is being towed. The device works on 3.3V and has a 10-bit digital output. The information obtained from this device, in conjunction with the gyroscope, will be used to determine if the user is accelerating too quickly or making dangerous stops or if it has been involved in an accident. The microcontroller will record this information and send the information in the next data packet over Wi-Fi.

2.5.2. Gyroscope

The gyroscope used in this project is the FXAS21002C manufacture by Freescale. This device communicates using SPI and has 16 bits of resolution measuring angular rates up to 2000°/s. Interrupts can be configured to generate when a threshold angular acceleration is reached. Using this device will allow the microcontroller to determine if the vehicle is turning at speeds considered unsafe.

2.5.3. OBD-II Transceiver

OBD-II (On-Board Diagnostic II) is a protocol which most vehicles use to relay information about the car to mechanics and others that would need the information. These messages are available from the OBD port on vehicles. The device will attach to this port and will use the OBD-II transceiver module to interpret the messages. The device used in this project is the STN1110. It is able to interpret most J1850, ISO and CAN messages that come across the OBD port. This device converts those messages into a UART message that is sent to the microcontroller.

2.5.4. GPS

The GPS device used in the VMS design is the M10478 manufactured by Antenova. This device communicates using a UART interfaces that implements the NMEA 0183 protocol that is standard for GPS receivers. This device would acquire GPS location of the VMS device which could then be transmitted back to the servers to allow fleet managers, or other users, to track the location of their vehicle. The M10478 operated on 1.8V and used a 1.8V to 3.3V level shifter to communicate with the UART channel on the Atmega.

2.5.5. Wi-Fi

To communicate data to our server from the microcontroller we decided on using the ESP8266 integrated Wi-Fi module. The reason we used Wi-Fi in this project was to emulate the use of GSM in the final version of our product. The module communicated using UART and sent messages to the online server using HTTP. It came pre-installed with firmware that allowed it to interpret messages from the AT command set. While this made it fairly easy to use, we did encounter issues with timing as UART communications are asynchronous. This meant that we had to include several delays every time the microcontroller communicated with the module to ensure that messages were properly delivered and received.

2.6. Web Server

Our web server runs a RESTful web service written in Java using the Jersey library. The service has the ability to call several APIs to store and return different data sets. These APIs are called based on the URI used to access to server. The server accepts HTTP requests and responds with JSON strings. Future expansion of the server would require greater security and support for multiple users and vehicles.

3. Design Verification

This section complements the Requirements and Verification table shown in Appendix A by describing the tests done to verify the functionality of our subcomponents.

3.1. Accelerometer Sensor

The tests performed to verify the accurate operation of the accelerometer module were as follows.

The first requirement, that the module detect 1g of acceleration with an error of $\pm$5%, was tested by holding the accelerometer on its 3 axes and confirming through a serial terminal that the accelerometer measured 1g. Since 1g is defined as the acceleration due to gravity it is easily verified that the acceleration can be 1g if the measurement axis is placed perpendicularly to Earth.

To ensure the accelerometer worked at a maximum temperature of 45°C a heat gun was used that tells its output temperature to heat the accelerometer. When held for a substantial amount of time such that the accelerometer was without a doubt 45°C, the accelerometer was positioned as in the test above and its output was read. It was found that at 45°C the accelerometer still works reliably.

3.2 Gyroscope Sensor

To verify that the gyroscope sensor worked with the Vehicle Monitoring System design the following tests were performed.

With the gyroscope powered and sitting still, the I2C output from the gyroscope was read. A read of zero would indicate that the gyroscope was sitting still as expected. Less than 5% variation from stand still was observed and so the verification passed.

To ensure that the gyroscope is able to meet the angular velocity requirements demanded by the device the requirement was made that the device would be able track a 45°/second turn with less than 5° lag. This was verified by using a protractor to rotate the device 90° and integrate the angular velocity output from the gyroscope. Should the device meet the requirements, it would be shown through the integration that the device in fact moved 90°. That was the case for the gyroscope used in the VMS design.

3.2. Microcontroller

The microcontroller serves many functions in the VMS design and is the most crucial component whose proper operation is absolutely necessary.

The GPIOs of the microcontroller must be able to source a 3.3V high signal and a $\leq$0.5V low signal such that it will be able to properly operate the NMOS for the battery control circuit and other components of the design. This was verified by properly turning on and off an LED.

The GPIOs must also be able to interpret high and low voltage signals appropriately so that it is able to interface with the many peripherals that the VMS design has. Proper operation is easily guaranteed by using a power supply to apply high and low voltages and printing out the corresponding values to the serial terminal.

The analog to digital converter is used in measuring the temperature and voltage for the battery. Proper operation is validated using a power supply to apply a voltage to the ADC pins and sweep it from 0 to 5V. The value of the ADC is printed to the serial monitor to ensure compare against the value of the power supply. The ADCs also worked as performed in field test when monitoring the battery voltage and temperature.

3.3. OBD-II Transceiver Module

The OBD module is another very important component in the VMS design. Proper operation was verified by connecting the board to the OBD port of a Chrysler Minivan. A serial terminal was used to communicate over UART with the OBD chip directly. Commands were sent to the OBD chip and if the proper return string was received then the OBD was deemed to be operating successfully.

3.4. Buck Converter Circuit

From Figure 5, it can be observed that the buck converter successfully converts 12V to 5V. It also produces a low voltage ripple of approximately 400mV which meets our requirements. And by using a 5Ω power resistor as the load, the buck converter was able to supply 1A of current.

3.5. Battery

We were able to charge the battery to 4.2V successfully and discharge using a 10Ω resistor as the load. To charge the battery, we used a DC power supply with an 8V input voltage and a 10Ω resistor in between the source and the battery to control current flow. With this we were able to get the data in the table below.

Time	Input current	Battery Voltage
10:40pm (start)	370mA	3.65V
11:40pm (end)	343mA	3.92V

For discharging the battery, we used a 10Ω resistor as the load and got the data in the table below.

Time	Input current	Battery Voltage
2:26pm (start)	330mA	3.90V
3:56pm (end)	322mA	3.70V

3.6. Power Regulator Circuit

In our design, we used a 3.3V output regulator which had a voltage ripple less than 5% as expected. Using a DC supply with 5V input voltage to the 3.3V linear regulator and a 5Ω power resistor as the load, we were able to confirm that it could supply current greater than 400mA.

3.7. Software

To verify the software, we went through all of the verification steps that we initially decided. We first powered the WiFi module and configured it to connect to our phone hotspot. This connection was verified by seeing the module as a connected device on the phone. Next we obtained messages from the accelerometer and gyroscope. These were decoded to ensure that the readings were accurate to the physical changes that we were imposing on the devices. Finally, we were able to send, store and retrieve data from the web server using both HTTP messages and also a JavaScript based web interface.

4. Costs

4.1. Parts

Item Description	Part Number	Vendor	Quantity	Total Cost
IC MCU 8bit 32KB Flash 28PDIP	Atmega328	Digikey	2	$6.76
3-axis gyroscope	FXAS21002CQR1CT	Digikey	2	$7.12
3-axis accelerometer	MMA8653FCR1CT	Digikey	2	$2.18
RF ANT MOD GPS SIRFSTARIV	627-1052-1	Digikey	1	$17.16
OBD-II to UART Interpreter	STN1110	Scantool	1	$9.99
Wifi Module	Esp8266	Amazon	1	$6.95
IC XLATR 8 bit low voltage translator	568-12295-1	Digikey	2	$2.04
USB Switch ICs USB 2 0 HS/UART 3:1 MUX	ISL54216IRUZ-T7A	Mouser	2	$2.82
Crystal 16.0MHz 20pf	631-1404-1	Digikey	2	$0.76
IC DIP Socket	ED3050-5	Digikey	2	$0.66
P channel Mosfet	DMG2305UX	Digikey	2	$0.76
P channel Mosfet	ZXMP6A13FCT	Digikey	2	$1.08
Schottky diode	MBRA130LT3GOSCT	Digikey	2	$0.92
NPN transistor	MMBT2222LT1GOSCT	Digikey	4	$0.52
General purpose diode	BAS16-E3-08CT	Digikey	4	$0.84
Probe pad	A1061444CT	Digikey	12	$1.68
IC transceiver	MCP2551-I	Digikey	2	$2.04
Step down switching regulator	LM2576	Texas Instr	1	$3.39
Thermistor	495-2084	Digikey	2	$1.58
Resistors		Digikey	24	$2.69
Capacitors		Digikey	98	$15.35
LEDs		Digikey	8	$2.08
TOTAL				$89.37

4.2. Labor

Name	Hourly Rate	Hours Invested	Total*2.5
Caleb Perkinson	$35	350	$30,625
Ishan Ahuja	$35	350	$30,625
Samuel Utomi	$35	350	$30,625
Total			$91,875

5. Conclusion

5.1. Accomplishments & Uncertainties

Currently we have a completely stand-alone PCB designed which can be plugged into and read data from a vehicle. We are able to transmit this data over Wi-Fi to our server for on demand storage and retrieval. The system can function for longer than a day on battery power alone, and the battery charging and discharging circuit is functioning as intended.

One function which we weren't able to get functional this semester was the GPS chip. We were able to isolate the issue and figure out that the problem was with the level shifter that we had to use to communicate between the microcontroller and the GPS chip. The reason we needed a level shifter was because the GPS chip operates at 1.8V and the microcontroller operates at 3.3V. The issue we had was the data became undecipherable after being passed through the level shifter, so we couldn't decode our location.

Another big issue we had was the multiplexer getting shorted to the 12V power supply during testing. This happened near the end of our project, and we didn't end up having enough time to order and re-solder a new one. As a result, we were unable to read data from the OBD port and post data over Wi-Fi concurrently.

A big change we made in our project was shifting from using the TI CC3200 microcontroller, as initially planned, to the ATMega328. The reason we made this change was because we had incessant issues with trying to flash our firmware over JTAG. The chip is only available in a QFN package, which meant that we couldn't test anything concretely until it was soldered onto our printed circuit board. After dealing with these issues for weeks we decided it would be wise to shift to the ATMega382 which was available as a DIP package making prototyping much easier. This shift also meant that we had to scrap our Azure server and access interface for a custom made one running on an Amazon Web Services Virtual Machine since Azure doesn't have built in support for the ATMega328. We were able to redesign and order a new printed circuit board to incorporate this change in time for our demonstration.

5.2. Ethical Considerations

Our team and project will strive to comply with the following stipulations in the IEEE Code of Ethics [5]

1) We accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

2) To be honest and realistic in stating claims or estimates based on available data; 3) to reject bribery in all its forms;

4) To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

5) To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

6) to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;

7) To avoid injuring others, their property, reputation, or employment by false or malicious action;

8) To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

5.3. Safety

Since our device has very low power consumption, there are very few components that will pose a threat to the user's safety. One major concern is that since we are using a primary lithium ion battery to power the device when there is no vehicle power, we need to take safety precautions to avoid damage to the device. The two types of hazards relevant to using rechargeable lithium ion batteries are chemical and electrical. [1]

- Chemical Hazards
    - Spillage
        - To prevent rupturing of the battery causing a spill, we ensure that the battery is sufficiently encased and isolated from user intervention.
    - Gas Emission
        - Since our device is not rated to function in the temperature range where gas emissions would be a possible (i.e. greater than 180oC), this is not a concern.
- Electrical Hazards
    - Joule Effect
        - Current flowing through the battery during charging/discharging will heat the device. To prevent this, we restrict battery charging and discharging to a safe range based on the battery specification.
        - We ensure that the current drawn is within an acceptable range, by thoroughly considering any possibilities of short circuits.
    - International Battery Standards
        - The battery we used is compliant with regulatory standards.
    - Damage due to overcharging
        - Overcharging is prevented by monitoring battery voltage.
    - Damage due to heat
        - Ambient temperature is monitored, and battery cutoff in the case of temperature reaching unsafe levels. To ensure that all relevant safety requirements met, all team members trained on battery safety by reading IEEE Standard for Rechargeable Batteries for Cellular Telephones, in addition to the lab safety and electrical safety quizzes taken online for the course.

5.4. Future Work

There are several things that we would like to update and continue to work on future renditions of this project. The first aspect that needs the most improvement is the software. The software can be broken down into firmware (board code) and user-facing/server-side software.

For the firmware, energy-saving modes were never taken advantage of. This is a huge update that could be made. By putting devices like the microcontroller, gyroscope, and other devices into low power massive power saving can be made. A state machine needs to be developed such that certain devices enter a power saving mode when the car is off. Devices like the OBD transceiver are no longer relevant because the car will not be sending OBD messages when the car is off and so the OBD could be put into a deep sleep mode. The gyroscope and accelerometer also have the capability to be put into sleep modes that only require current on the μA scale. These devices can then be woken up by sudden physical occurrences and can generate an interrupt that can be sent to the microcontroller to wake it up from sleep mode.

An enhanced user-interface is also desired such that users would have unique logins so that they can access their own data, in its current configuration only one device is allowed. Data analytics to provide the user with a better summary of the acquired data would also be another area to work on.

On the hardware side, a more compact design is desired. Increased space saving and cost-efficiency could be realized by using more advanced technologies such as reflow or wave soldering rather than hand solder. This would also enable the use of smaller package components. A reworked, more efficient power system would also be desired to allow for more efficient charging and more elegant recovery from a depleted battery situation.

References

[1] Safety of Lithium Ion Batteries [Online], Available: http://www.rechargebatteries.org/wp-content/uploads/2013/07/Li-ion-safety-July-9-2013-Recharge- .pdf

[2] Philip T. Krein, Elements of Power Electronics

[3] Design Trade-offs for Switch-Mode Battery Chargers [Online], Available: http://www.ti.com/lit/ml/slyp089/slyp089.pdf

[4] A Guide to Battery Charging [Online], Available: http://www.operatingtech.com/lib/pdf/A%20Guide%20to%20battery%20Charging.pdf

[5] IEEE Code of Ethics, Retrieved February 2016, Available: http://www.ieee.org/about/corporate/governance/p7-8.html

[6] Basic Calculation of Buck Converter Power Stage http://www.ti.com/lit/an/slva477b/slva477b.pdf

Appendix

A. Requirements and Verification Table

MODULE NAME	REQUIREMENTS	VERIFICATION	Points	Verified
Accelerometer	1) Sensor should be able to detect acceleration within 1g with an error of ±5% 2) Sensitivity is maintained at the maximum acceptable temperature of 45°C	**1.** (a) Power microcontroller to 3.3V (b) Connect accelerometer to microcontroller (c) First record output data when there is no motion to detect bias error, then apply force to the device with hand and ensure output data changes **2.** (a) Using a heat gun, increase the temperature of the device (b) Using a thermometer, monitor temperature rise of device when not in motion and measure change in output during this process.	5	Y
Gyro sensor	1) Gyro bias error $\leq 1°/sec$ 2) Gyro lag $\leq 5°/sec$ when angular velocity greater than 45°/sec	**1.** (a) Power microcontroller to 3.3V (b) Connect gyro sensor to microcontroller (c) Record output data during no motion **2.** (a) Power microcontroller to 3.3V (b) Connect gyro sensor to microcontroller (c) Rotate gyro 90 degrees using a protractor and use the change in angular velocity to verify this	5	Y

Microcontroller	Digital Output 1. Digital 0 corresponds to $V_{out} \leq 0.5V$ 2. Digital 1 corresponds to $V_{out} \geq 3V$	**1.** (a) Power Microcontroller with 3.3V (b) Program all pins to output Digital 0 (c) Probe pins to verify $V_{out} \leq 0.5V$ **2.** (a) Power Microcontroller with 3.3V (b) Program all pins to output Digital 1 (c) Probe pins to verify $V_{out} \geq 3V$	5	Y
	Digital Input 1. Digital 0 corresponds to $V_{in} \leq 0.5V$ 2. Digital 1 corresponds to $V_{in} \geq 3V$	**1.** (a) Power Microcontroller with 3.3V (b) Set pins to input mode and print their value to the serial port (c) Apply 0.2V to each pin to ensure proper reading **2.** (a) Power Microcontroller with 3.3V (b) Set pins to input mode and print their value to the serial port (c) Apply 3V to each pin to ensure proper reading		
	Analog Digital Converter 1. Can read analog signals from 0V to 5V ±5%	**1.** (a) Hook ADC to power supply (b) Enable rated current on the supply to prevent damaging the microcontroller (c) Print the value of the ADC to the serial port (d) Sweep the voltage from 0 to 5V and verify proper quantization		

	Low Power Consumption 1. $I_{avg} \leq 500\,\mu A$ at 3.3V	**2.** (a) Power Microcontroller with 3.3V (b) Place ammeter in series with supply (c) Put microcontroller in low power mode. (d) Ensure $I_{avg} \leq 500\mu A$		
OBD-II Transceiver module	1) Should be able to decode messages that conform to the OBD-II standard	**1.** (a) Connect to the car and verify that OBD messages are being received correctly	5	Y
GPS module	1) Should be accurate within 5m	**1.** (a) Power microcontroller to 3.3V (b) Connect module to microcontroller (c) Compare coordinates received on server to google GPS location of phone	5	N

Buck converter charging circuit	1) 5V output voltage ±5% 2) Should be able to supply current up to 1A 3) Should stop charging once battery voltage is 4.2V	**1.** (a) Attach 10Ω resistor bank as load (b) Connect oscilloscope probes at both ends of the load and measure voltage ripple **2.** (a) Connect current sensor of the multimeter in between the output of the charging circuit and 10Ω resistive load and measure current **3.** (a) Connect battery to charging circuit (b) Attach analog output pin from Arduino (or preferred microcontroller) to the gate of the NMOS to control charging (c) Attach analog input pin of Arduino to battery for voltage sensing (d) Supply current to the battery by applying 5V output at the source (e) Once battery voltage is 4V, current to the battery should be cut off	15	Y
Battery	1) Should be able to last a day when discharging 2) Can handle charging current up to 0.5A without heating up significantly	**1.** Measure battery capacity and discharge current and ensure it can last the required time **2.** (a) Place a 3Ω 1W resistor between the DC supply and the battery. Then apply 5V from the supply. (b) Measure temperature using thermostat and ensure it does not go above 30C after an hour of charging		Y

Power regulator circuit	1) Output voltage is 3.3V ±5% with a total current consumption of up to 400mA	**1.** (a) Use DC supply with 12V input (b) Attach oscilloscope probes across 100Ω load (c) Ensure output voltage is within bounds		Y
Software	1) Able to connect to the internet. 2) Able to read data from I2C peripherals. 3) Able to read data over UART. 4) Able to send data to online server. 5) Able to store data in online server. 6) Able to retrieve and view stored data from online server.	**1.** (a) Power device with 3.3V (b) Run AT Commands to configure connection to Internet (c) Check connected devices on Mobile Hotspot to ensure connection **2.** (a) Power devices with 3.3V (b) Send I2C command to obtain device IDs (c) Ensure device IDs are valid. **3.** (a) Power devices with 3.3V (b) Send AT commands to obtain device IDs (c) Ensure device IDs are valid. **4.** (a) Power devices with 3.3V (b) Send test string to server (c) Ensure test string arrives at server **5.** (a) Check server data to ensure string stored is the same as what was sent during send test. **6.** (a) Retrieve server data using web application and ensure it matches test data.	10	Y