**Automotive Data Logger**

Michael Jang (myjang2) & Jacob Drewniak (jdrewni2)

ECE 445 Project Proposal – Fall 2018

TA: Evan Widloski

**1 Introduction**

* 1. **Objectives and Background**

The primary objective of the electronics subsystem of an FSAE team is collecting useful data in support of the design and validation process. This data can range from lateral and longitudinal acceleration to ignition timing to suspension strain values and is absolutely essential for a finely-tuned race car. Commercial data acquisition systems are extremely expensive and often times use difficult, slow, and proprietary software packages that don’t properly parse or format data. The team’s current data logging module is made by Vector but is nonideal for several reasons. The software required to download the data from this logger is difficult to use and quite slow. Additionally, the files it produces have a maximum size of 5.25 MB and are not parsed correctly for easy data extraction. Post processing is necessary to produce usable data. Finally, data is often completely lost or, at the very least, corrupted if the power supply is severed during downloads. Replacing this data logger with a different commercially available option is not within the team’s budget, so the best solution is to build our own with a simpler UI, better extraction methods, and faster speeds.

The goal of this project is to design a data logger that can save all incoming sensor data at a high frequency and then make it quickly and easily available to the user through a simple, streamlined interface. This ideally can be done through either USB or Bluetooth connectivity depending on the scenario and testing session length. The team already employs a custom-built sensor processor that converts analog and digital sensor data into CAN format, so the data logger must simply log incoming CAN messages from the sensor processor [11]. This simplifies the overall architecture to merely storing and transmitting the data to team members at the necessary times with several protection measures built in to prevent data corruption or losses.

**1.2 High-Level Requirements**

* The system must be able to store all the data from incoming CAN messages into non-volatile memory.
* The data must be able to transfer the data through USB and Bluetooth.
* In case the system loses battery power, it must be able to power itself for long enough to properly power down and ensure that no data is lost or corrupted in the process.

**2 Design** 

**Figure 2.1 Block Diagram**

This design has three main components: power supply, control unit, and user interface. The power supply module will drive the entire circuit with the proper voltages (3.3V and 5V). There will also be a feature in which we can temporarily power the entire system in the case that electrical power to the car is lost. The control unit has the microcontroller at the heart of it, and it is mainly responsible for all the data transfer and data storage. Finally, the user interface will allow the team to be able to read the logged data through different means: USB and Bluetooth.

**2.1 External System**

The external system is the vehicle that the logger will be receiving data from. This design only needs two things from the vehicle: a battery and sensors that communicate through CAN.

**2.2.1 13.5V Battery**

The car and all of its electrical components, including the proposed data logger, are powered by a lithium-ion battery that is nominally at 12V. Normal operating conditions usually range from 13V-14V. The battery can be charged using a lithium-ion battery charger when the car is off or by the car’s alternator while the engine is running.

**2.1.2 Sensors**

Our car has a large number of sensors that all communicate through CAN. These are the data points that the circuit will be logging. Sensors give the team important data points such as lateral acceleration, large current draws (water pump, radiator fan, etc.), steering angle, wheel speed, etc. All these data points are essential to log in order to calibrate/tune the car, validate mechanical and electrical designs, and make improvements for future designs.

**2.2 Power Supply**

The power supply subsystem will provide the necessary voltages throughout the board using two regulators with enough current capacity to properly power all circuitry. The main power supply will be provided by the car battery. There will also be a super capacitor stack as a backup powering method in the scenario that battery power is lost.

**2.2.2 Super Capacitor Stack**

A stack of multiple super capacitors will be used to power the board if the battery supply is cut off. The reason for a stack instead of just one is to achieve a high enough voltage across the capacitors to properly power the board. The purpose of the super capacitors is to temporarily power the system for up to 2 seconds in the case that the battery power to the car is completely lost. This is necessary so that the system can properly power down without losing or corrupting any essential data that has been logged [9].

*Requirement: The super capacitor stack must have a high enough voltage to power the board if battery power is lost. Because battery power is somewhere between 12V and 14V, the voltage supply from this stack will be close to these numbers. Most super capacitors have a maximum voltage of around 2.7V-3V because of dielectric breakdown at higher voltages. Because of this, multiple in series must be used to raise the overall voltage to usable levels. Four in series will produce between 10.8V and 12V, which is sufficient to power the other components on the board. The buck converter that will be used has a minimum supply voltage of 6.5V. Because the board consumes a constant current of around 500mA and we will conservatively require about 2 seconds to properly power down the board, the following equations can be used to calculate capacitance*

$I=C\frac{dV}{dt}=-500mA$

$\frac{dV}{dt}=\frac{6.5-12}{2}=-2.75$

$C=0.18F $

*Because four in series are to be used and accounting for a safety factor, each individual super capacitor will require at least 1.5F – 2F [9].*

**2.2.3 Super Capacitor Control**

This IC will be connected to all four super capacitors and will function much as a battery management system [9]. It will function as both a monitor and a controller. The monitor functionality will include measuring state of charge, voltage level, and current I/O. The controller aspects of this IC will include cell balancing, pack protection, and microcontroller interfacing [3].



**Figure 2.2 Supercapacitor System Schematic**

*Requirement: As mentioned above, this IC will require cell balancing, overage/underage protection, and pack monitoring. Cell balancing current is not a huge priority and the super capacitors don’t hold huge amounts of charge anyway, so a 10mA – 20mA balancing current should be enough. Overvoltage, overcurrent, and undervoltage protection are mandatory to ensure that the capacitors are not damaged at any point. An integrated coulomb counter and ADC of at least 12-bit accuracy are also mandatory for accurate voltage, current, and state of charge monitoring purposes.*

**2.2.4 5V Regulator**

A buck converter will be used to step down the 13.5V battery supply to a 5V voltage rail that powers several IC components on the board. A buck converter was chosen because of the higher efficiency at large voltage drops from supply to output. This prevents large amounts of power losses and heat dissipation into the board that could prevent proper functionality. This 5V regulator also feeds into a 3.3V regulator that powers most of the other components on the board. The LM2675 was chosen because it has a fixed output voltage of 5V and a rated output current of 1A [4].

*Requirement: The board will likely sink only up to 300mA during normal operation. The 5V rail supplies power to every component except the supercapacitors and their controller, so, accounting for safety factor, this buck converter will need to provide at least 500mA [4].*

*Verification: Output voltage and noise will be tested through an oscilloscope with varying loads*

**Figure 2.3 Voltage Regulator Schematic**

**2.2.5 3.3V Regulator**

As mentioned above, this 3.3V supply will be originated from the 5V regulator which will then power most of components throughout the system. Different from the 5V regulator, this regulator is a Low-Dropout (LDO) Linear regulator. Because the drop from 5V to 3.3V is fairly small, the efficiency of the linear regulator is reasonable and, therefore, it doesn’t dissipate much power. The system benefits not only from the simplicity and size of the linear regulator, but also from the cleaner, more accurate output signal when compared with switching converters [12]. The chosen linear regulator (TPS73633) has a fixed voltage output of 3.3V, and an output noise equation given as [5]:

$V\_{N}\left(μV\_{RMS}\right)=8.5\left(\frac{μV\_{RMS}}{V}\right)\*V\_{OUT}(V)$

Setting the output voltage as 3.3V, the generated output noise will be 28.05µVRMS. This is a very low noise output that will meet requirements. The noise is so low because linear regulators don’t have any switching noise [12]. The TPS73633 also has a rated output current of 400mA which perfectly meets requirements [5]. Because this topology is a linear regulator and not a switching regulator, power dissipations cannot be ignored. The dissipated power can be given as:

 $P\_{D}=\left(V\_{IN}-V\_{OUT}\right)\*I\_{OUT}$

The worst-case power dissipation is 0.68W which is within the rated power dissipation [5].

*Requirement: The 3.3V regulator provides power to about two-third of the board, including the microcontroller, which will be the most power-hungry component of the board outside of the supercapacitors during startup. This regulator will likely need to sink up to 250 mA, so we will require a regulator output of at least 400 mA. Because the difference in voltage between this regulator and the buck converter is only 1.7V, the dropout voltage should be no more than 1.2V and preferably lower. A high power supply rejection ratio and low noise output would be a nice bonus [5].*

*Verification: Output voltage and noise will be tested through an oscilloscope with varying loads. Board temperature will be monitor to ensure that not too much heat is being dissipated.*

**2.3 Control Unit**

The control unit for this system manages the incoming CAN messages and stores the data into an SD card through the use of a microcontroller and CAN transceiver. The microcontroller will transfer the data through Bluetooth and USB for user communication.

**2.3.1 Microcontroller**

 The chosen microcontroller (PIC32) intakes the message that the CAN transceiver sends, and it will store that data into an SD card through SPI communication. It can then send the logged data out to a Bluetooth IC through an SPI bus and a USB through a UART converter IC [1]. The microcontroller is also responsible for communicating with the super capacitor controller to set protection limits and to monitor and make decisions if power loss or overage occurs [3].

*Requirement: The PIC32 has been historically used on all our team’s boards, so several code libraries have already been built and the functionality we require has already been demonstrated. The requisite functionalities include programmable analog and digital I/O; high-speed CAN, SPI, I2C, and parallel communications protocol functionality, external oscillator compatibility, an internal ADC, and 2 MB of flash memory [1].*

**2.3.2 CAN Transceiver**

 The CAN transceiver (MCP2562) will read all the high-powered 5V CAN messages coming from throughout the car via CAN bus and produce low-powered 3.3V Tx/Rx signals that are readable by the microcontroller [2].

*Requirement: This IC must be able to convert the 5V differential CAN signals from other boards on the car to 3.3V single signals that can be read by the microcontroller and vice versa.*

**2.3.3 Non-Volatile Memory (NVM)**

 Some form of non-volatile memory (either a flash chip, SD card, or potentially some other form of NVM) is necessary to hold all the data that the microcontroller and CAN transceiver process. This non-volatile memory can then communicate back to the microcontroller so that the user can read the data through Bluetooth or USB.

*Requirement: Our car runs a 1Mbps baud rate CAN bus. Each CAN message is about 100 bits long and includes approximately 80 bits that require storage. Assuming 8 hour testing sessions without pulling data, the NVM must have minimum storage of 2-3 GB with a data transfer rate of at least 1 Mbps. It also must have some sort of wear leveling for large numbers of reads/writes. It would also preferably have an easy to use communications protocol such as SPI or I2C.*

**2.3.4 Temperature Sensor**

The only sensor in this circuit is the temperature sensor (MAX6610). This will communicate with the microcontroller for circuit diagnostic purpose. The logger will likely be placed in a high-temperature environment near the fuel tank of the car, and this sensor will relay the temperature of the board to ensure that all chips are still within their temperature operating range [6].

*Requirement: Must have a read range larger than the operating range of the board. Driving conditions go as low as -20 degrees Celsius, and the board could be exposed to temperatures up to 80 degrees Celsius on a hot day if it were placed near the fuel tank or some other hot component [6].*

**2.4 User Interface**

The user interface module will make it possible for any user to read the data that has been stored in NVM. There are two methods to communicate the data: USB and Bluetooth. The Bluetooth will be useful for when smaller data sets are desired between test runs. The USB interface will be used for transferring large sets of data from longer testing sessions because the Bluetooth transfer rate is much slower.



**Figure 2.4 SD Card Reader & CAN Transceiver Schematic**

**2.4.1 USB-UART**

This IC will communicate with the microcontroller through some form of serial communications bus to transfer data from storage. The IC will then convert these messages to the USB protocol so that the data can easily be transferred to a laptop with a simple cable [7].

*Requirement: Simple communications protocol with the microcontroller is a must. It also must have a very fast data rate (at least 50-100 MBps) to download large data sets in a reasonable amount of time. Preferably uses USB 3.0 rather than USB 2.0 to achieve fastest speeds possible [7].*

**2.4.2 Bluetooth**

 The Bluetooth IC will take SPI signals from the microcontroller (originating from the NVM) and convert them to the necessary signals for the antenna to be able to transmit the data wirelessly to an external device.

*Requirement: This is tougher because nearly any Bluetooth chip would work, but ideally it would be able to transmit at near the theoretical limit at 3Mbps and range of around 10-20 meters [10].*

**2.4.3 Antenna**

The antenna will communicate with the Bluetooth IC that has produced to necessary signals to transmit. The antenna will then wirelessly communicate with an external device.

*Requirement: Bluetooth operates at a frequency 2.4 GHz so the antenna must be compatible with this frequency. This wireless data transfer is not used for live telemetry while the car is driving, so ideally the range and power of the antenna will be quite small (<= 30 ft) so that accidental download doesn’t occur if the car drives too close. The antenna would ideally radiate with equal power in all directions.*

**2.5 Risk Analysis**

The modules that will be the toughest to implement will be the Bluetooth IC and accompanying antenna. The Bluetooth module will require pretty significant software implementation to program proper data transfer wirelessly, and it will require software to be written on the receiving side as well. Additionally, care will need to be taken with the RF traces, PCB routing, and passives selection between the Bluetooth IC and the antenna to ensure proper transmission and impedance matching of the 2.4 GHz signal. If we decide to design our own antenna, the size and shape must be carefully chosen to transmit a quality, omnidirectional signal that meets the range requirements [10].

**3 Ethics and Safety**

The Lithium-Ion car battery is extremely dangerous if the terminals are shorted together. Extreme caution is needed when handling the battery near metal tables or anything capable of being a conductor. Lithium-Ion batteries can also explode if over-charged or if they are exposed to temperatures outside the recommended operating conditions. The team ensures that the battery is well ventilated and never excessively charged under IEEE Code of Ethics #1 (safe, healthy, and environmentally safe) [8].

The whole point of this project is to design a faster, lighter, and safer race car by improving the usage of sensor data. This data can be used to validate mathematical and computational models or provide relevant numbers where models do not exist. This is in direct compliance with IEEE Code of Ethics #5 (to improve understanding through technology) [8]. To reduce the chances of violating IEEE Code of Ethics #3 (to be and realistic with data interpretation), the team always tries to validate all data through both basic hand calculations and complex computational models [8]. If the numbers vastly disagree, we expend every effort to discover why.

**References**

[1] Microchip Technology, ‘PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Family’, 2018. [Online]. Available: [www.microchip.com/mymicrochip/filehandler.aspx?ddocname=en574992](http://www.microchip.com/mymicrochip/filehandler.aspx?ddocname=en574992)

[2] Microchip Technology, ‘High-Speed CAN Transceiver’, 2014. [Online]. Available: [www.microchip.com/mymicrochip/filehandler.aspx?ddocname=en561044](http://www.microchip.com/mymicrochip/filehandler.aspx?ddocname=en561044)

[3] Linear Technology, ‘High Current Supercapacitor Backup Controller and System Monitor’, 2014. [Online]. Available: [www.analog.com/media/en/technical-documentation/data-sheets/3350fc.pdf](http://www.analog.com/media/en/technical-documentation/data-sheets/3350fc.pdf)

[4] Texas Instruments, ‘LM2675 SIMPLE SWITCHER® Power Converter High Efficiency 1-A Step-Down Voltage Regulator’, 2016. [Online]. Available: <http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=lm2675&fileType=pdf>

[5] Texas Instruments, ‘CAP-FREE NMOS 400 mA LOW-DROPOUT REGULATORS WITH REVERSE CURRENT PROTECTION’, 2014. [Online]. Available: <http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=tps73633-ep&fileType=pdf>

[6] Maxim Integrated, ‘Precision, Low-Power, 6-Pin SOT23 Temperature Sensors and Voltage References’, 2003. [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/MAX6610-MAX6611.pdf>

[7] Future Technology Devices International, ‘FT230X (USB to BASIC UART IC)’, 2016. [Online]. Available: [www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\_FT230X.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf)

[8] IEEE, ‘IEEE Code of Ethics’, 2018. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>

[9] R. Roderick, ‘A Look Inside Battery-Management Systems’, 2015. [Online]. Available: <https://www.electronicdesign.com/power/look-inside-battery-management-systems>

[10] SparkFun Electronics, ‘Bluetooth Basics’, 2013. [Online]. Available: <https://learn.sparkfun.com/tutorials/bluetooth-basics>

[11] Kvaser, ‘CAN Protocol Tutorial’, 2018. [Online]. Available: <https://www.kvaser.com/can-protocol-tutorial/>

[12] D. Knight, ‘Introduction to Linear Voltage Regulators’, 2016. [Online]. Available: <https://www.digikey.com/en/maker/blogs/introduction-to-linear-voltage-regulators>