FAMU-FSU College of Engineering
Department of Electrical and Computer Engineering

DETAILED DESIGN REVIEW AND TEST PLAN

Project Title: SAE Baja Data Acquisition System
Team #: 5
Website: http://eng.fsu.edu/~willidew/baja

Student Team Members:
Chris Riker, Computer Engineering (Email: cdr11c@my.fsu.edu)
Hebe Perez, Computer Engineering (Email: hp09d@my.fsu.edu)
Dewey Williams, Computer Engineering (Email: dmw10g@my.fsu.edu)
Tyler Dudley, Electrical Engineering (Email: dudleyt2004@gmail.com)

Project Coordinator: Dr. Michael Frank

Additional Reviewer: Dr. Linda DeBrunner

Submitted in partial fulfillment of the requirements for
EEL4914C – ECE Senior Design Project II

February 6, 2015

Project Executive Summary

The Society of Automotive Engineers encourages college students worldwide to apply classroom knowledge to real world design concepts through their annual competitions. SAE International hosts five categories of competition that include Aero, Clean SnowMobile, Formula, Baja, and Supermileage. The SAE team at the FAMU-FSU College of Engineering actively participates in the Baja competition each year and subsequently builds upon their designs from previous competitions for vehicle improvement.

Previously, the SAE Baja Team has made changes to the vehicle based on estimates of the performance of the vehicle. This year, Team E#5 will add a quantitative approach to the Baja design process by adding a data acquisition system (DAQ). The DAQ will record values from various vehicle systems and will be stored for later use. DAQ subsystems will include measurement for speed, acceleration, and tire pressure. In addition to recording and storing data, the DAQ system will include a communication component to help the team during competition. The communication component will have a signal going from the driver to the pit crew to alert the pit crew of low tire pressure. There will also be a way for the driver to manually indicate to the pit crew that (s)he will be pitting. This will allow the pit crew to prepare for the driver and decrease the time it takes for the vehicle to get back on the track.

Currently, the design is in the testing and development stage, with preparations being made to begin the building phase as components are completed. There are only minor purchases and decisions left to be made. While some features have been tabled or removed, much of the original needs analysis still applies, and the team is striving to deliver a useful and quality product to the Baja team. The system will be easy to use by both the driver and the pit crew, requiring minimal setup to begin acquiring and displaying data.

The system will follow the competition design rules outlined by SAE International as well as hold up against the rough terrain of a Baja competition track. The system will be finished by the April 9th, 2015 competition date in Auburn, Alabama and will stay within the $750 budget.

Table of Contents

Introduction

1.1 Acknowledgements

The SAE Baja DAQ design team would like to acknowledge our project advisors, Dr. Frank and Dr. DeBrunner, for any advice, input and ideas offered on how to proceed during the design phase of the DAQ. The team would like to acknowledge the FAMU-FSU Society of Automotive Engineers for any sponsorship and advice they may provide and including this design team in this year's Baja car. The team would also like to thank Arrigo Dodge of South Florida for their contribution of four tire pressure monitoring system sensors to the Baja DAQ team, as well as Texas Instruments for generously donating all of the processors and development tools used in the project as well as several analog ICs. Finally, the team would also like to thank the FAMU-FSU Electrical and Computer Engineering department for their financial contributions towards the project.

1.2 Problem Statement

The acquisition of data can be a very important part of efficiently optimizing designs of many kinds, especially those involving mechanical systems such as automobiles, provided that the data collected is used in an appropriate manner. The design team aims to design and implement a Data Acquisition System (or DAQ) that will collect and store important data and ultimately make the FAMU-FSU Baja team's vehicle more successful in this year's competition.

In previous competitions, the SAE Baja team had little quantitative information to use when making changes to the vehicle. The team often wasted valuable race time by entering the pits when it was not necessary. The DAQ will inform the SAE Baja team of the proper changes and procedures needed to improve the vehicle mid-race, allowing them to focus less on problem identification and more on problem solving. The DAQ will inform the driver and pit crew of any issues with the vehicle, as well as provide rudimentary communication between the driver and the crew. Data collected by the system will be available in the pit wirelessly and in real time.

1.3 Operating Environment

The Baja vehicle to which the DAQ will be mounted will be driven on rough terrain in a harsh outdoor environment during competition. The system will potentially be subject to shock and vibration due to the rough terrain, rain, dust, dirt, mud, rocks, heat (both due to weather and ambient engine heat) and RF interference from other similar systems on opposing vehicles. The system will need to withstand temperatures up to at least 120 degrees Fahrenheit. In designing the system, the team will make considerations for the inclement conditions which the system may face during the SAE Baja competition.

1.4 Intended User(s) and Intended Use(s)

1.4.1 Intended User(s)

The intended users of the DAQ are the members of the FAMU-FSU SAE Baja team. As all members of the SAE Baja team must be at least 18 years of age (for liability reasons), any and all users of the DAQ must also be at least 18 years of age. Any user of the DAQ must also be adequately trained in the use of the system. One of the important attributes of the system, as outlined in the Needs Analysis and Requirements Specifications, was usability. This means that the system will be easy to use so that the Baja team can focus on the competition as opposed to the DAQ operation. Thus, just about any engineer on the Baja team will able to use the DAQ. Only DAQ team members should exchange or modify the hardware components of the system itself.

1.4.2 Intended Use(s)

The DAQ is designed to be used by the Baja team members to evaluate various data points about the vehicle and make changes based on the data collected. To begin using the DAQ, the Baja team will have to simply turn the system on and begin running it (the procedure to run the DAQ is TBD based on the design). Once powered on and instantiated, the DAQ will begin collecting and displaying data to the users. The received data will be written to a storage device so that the Baja team can analyze it after testing or competition. During the competition, the Baja team will be able to use this data to make adjustments to the vehicle as needed. Ultimately, it is up to the Baja team to determine what is to be done with the collected and displayed data.

1.5 Assumptions and Limitations

1.5.1 Assumptions

- All components used in the design of the DAQ will be chosen such that their specifications meet or exceed the requirements of the system.
  - The wireless data transmission module(s) will exceed a range of 3 miles urban.
  - The main power source and the subsystem power sources will be chosen such that they allow at minimum 8 hours of system operation.
  - The storage medium will be large enough to accommodate all collected data
    - It is estimated that 1 GB will be more than enough to accomplish this
    - 40 bytes per line * 18000 seconds (5 hours) * 1 Hz = 720KB
- The main MCU will have enough I/O pins to be able to interface with every subsystem of the DAQ.
- The main MCU will be mounted inside of a protective enclosure to shield it from the competition conditions.
- The end product will only be used on the Baja vehicle.
- The system will be subject to inclement conditions outlined above in **Section 1.3 - Operating Environment**

1.5.2 Limitations

- The DAQ will be limited in some ways by the Baja competition rules.
  - The engine of the vehicle cannot be modified in any way.
  - The DAQ must use its own power source separate from the Baja vehicle's power source.
  - The system cannot interfere with any essential vehicle systems.
- The system shall be designed within an initial budget of $750.
- The system shall be able to withstand temperatures up to 120 degrees Fahrenheit.
- The display shall be easily readable by the driver of the vehicle.

1.6 Expected End Product and Other Deliverables

Before April 9th, 2014, the team will deliver a data acquisition system to the SAE Baja team. This system will collect, use, and display important data about the Baja vehicle to the team during the SAE Baja competition. This data includes acceleration, speed and tire pressure. This data will be written to removable media so that the Baja team can import it to a computer and analyze it as they see fit. The system will consist of two sections, the collection system mounted other vehicle (outlined in section 1.6.1), and the pit station where a readout of the collected data will be given to the pit crew (outlined in 1.6.2).

The Baja team will also be provided with a copy of each deliverable report upon completion, so that they may stay up to date on the development of the DAQ. The Baja team will also receive schematics of the system, top-level block diagrams, and data sheets and information about each component used. Additionally, they will be given a copy of all source code used in development of the project, so that future Baja teams may learn and improve on the design further.

The end product will be a standalone data acquisition system that will collect data about the following subsystems:

- Acceleration
- Speed
- Tire Pressure

The data will be logged onto a removable memory host that can stored for later use and be sent to the pit crew so that they may monitor the status of the vehicle remotely. The data will also be displayed on the vehicle's subsystem for driver awareness. In addition, the pit crew can be alerted by the driver ahead of time if a pit stop is necessary, so that the crew can prepare.

1.6.1 On-vehicle Data Collection

The actual data acquisition system itself will be mounted to the vehicle. The unit will collect data about the vehicle from several sensors, store that data to a removable media device in a CSV file, and display some of that data to the driver and the pit crew. The vehicle will also have an on-screen indication that will warn the driver of low tire pressure. The unit will consist of the following:

- 1 - Hercules LAUNCHXL-TMS57004
- 4 - MSP430G2553IN20 Microcontrollers
- 2 - TI DRV5053 Hall effect sensors
- 4 - Schrader C4N3MF9 TPMS Sensors
- 1 - TPMS Sensor reader (TBD based on further research)
- 1 - LSM303 accelerometer
- 1 - Xbee-PRO 900HP wireless transceiver
- 2 - $LiFePO_4$ 24Ah batteries
- 1 - High-gain Dipole Antenna
- 1 - Adafruit 2.2" TFT display
    - Driven by an ILI9340 display chip
    - Onboard SD card slot

1.6.1 Pit Unit

The pit unit will include a display for the collected information as well as mirrored indicator lights that warn the pit when the vehicle requires fuel or tire pressure. This unit will include the following items:

- 1 - Raspberry Pi (B+)
- 1 - Xbee-PRO 900HP wireless transceiver
- 1 - Display
- Pit indicator light

1.6.2 Deliverable Reports

The DAQ team has completed 3 milestones in the Fall semester, and will complete 3 more in the Spring. The SAE Baja team is provided with copies of each deliverable report as they are completed. This is to ensure that the Baja team is satisfied with the progress of the DAQ and all of its components, and to ensure that the system fulfills the needs set forth by the team. The completed milestones are as follows:

- Needs Analysis and Requirements Specifications (Completed September 18th, 2014)
- Project Proposal and Statement of Work (Completed October 17th, 2014)
- System-Level Design Review (Completed November 14th, 2014)
- Detailed Design Review & Test Plan (Completed February 6th, 2015)

1.6.3 Documentation and Configuration Files

Along with the actual system itself, the Baja team will be provided with documentation about the implementation of the system. This includes a top-level block diagram, information about each component, and all source code for the project. Those items are outlined below.

1.6.3.1 Top-Level Block Diagram

The top-level block diagram of the system will provide the Baja team with an idea of how the system is laid out at a glance. This diagram will show each different DAQ components and illustrate how they interface with each other. The diagram will be stored as a PDF to allow for portability between platforms. This will also allow the Baja team to make comments on the document to share amongst themselves.

1.6.3.2 Component Information

The Baja team will also be provided with information about each component used in the system. For each component used in the DAQ, the Baja team will receive:

- Datasheet
- Part number
- Information about how the part is interfaced with the overall system

This information will allow this the current and future Baja teams to easily find and replace any failed components with the same components or components with similar specifications. In the event that future teams with to improve on the DAQ design, they will be able to choose new components that will be compatible with the rest of the system.

1.6.3.3 Source Code

All of the source code used to implement the system will be given to the SAE Baja team. This will allow future members of the SAE Baja team to learn from the DAQ team's implementation of the system and improve on the design further. The files will be added to a *.zip or *.rar archive and given to the team with all of the other deliverables.

2 System Design

2.1 Overview

A top-level block diagram for the overall design of the system is found below in **figure 1**. The design will have two sections, the pit unit, which allows the pit crew to monitor the vehicle remotely and the on-vehicle DAQ system, which collects, stores, displays and transmits data from the sensor network. The vehicle MCU will poll data from the sensors, format the data, write the data to an SD card, and transmit data to the pit unit via an XBee-PRO 900HP. The system will be powered by two 3.2V $LiFePO_4$ batteries connected in series. The on-vehicle display will be done using a 2.2" SPI LCD display, and will display the speed of the vehicle as well as other collected information.

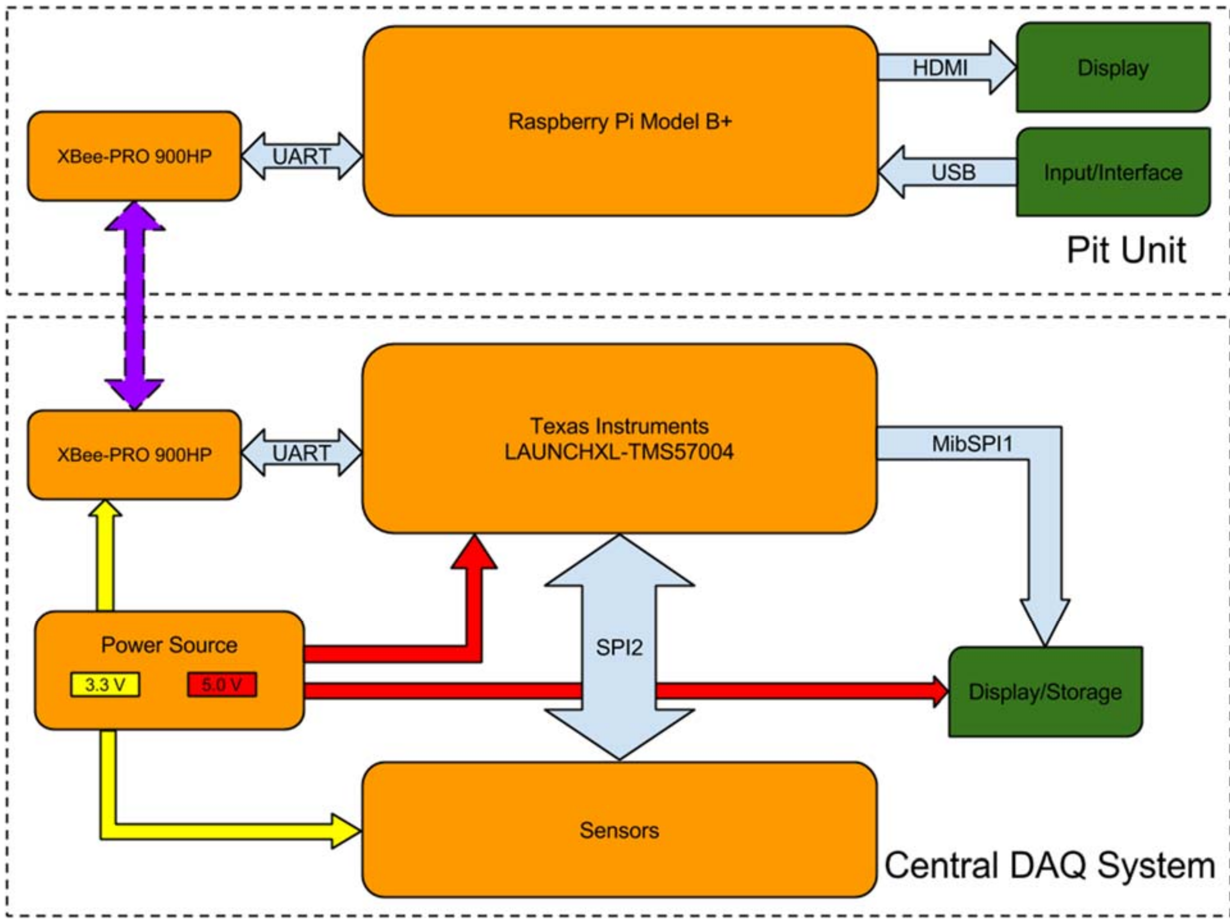


Figure 1: Top-Level Block Diagram

2.2 Major Components and Component Requirements

2.2.1 On-Vehicle Data Collection Unit

The unit mounted on the vehicle will contain the main microcontroller, a power source, a sensor network, a wireless transceiver and antenna, a microSD card slot, and a display as illustrated by the top-level block diagram in **figure 1**. The main microcontroller will poll data from the sensor microcontrollers--which will be continually collecting analog outputs from the sensors they are connected to, converting the analog inputs to digital representations, and converting the digital representations to the desired units (e.g. MPH, m/s^2, etc.). If polled, the sensor microcontrollers will then send their final calculated values to the main microcontroller via an SPI bus. Each sensor microcontroller will be powered by the regulated power source outputs via its own connection, and will not be powered by the development board's 5 V/3.3 V regulated outputs.

2.2.1.1 Main Microcontroller

The main on-vehicle microcontroller will be the central hub of information flow for the entire system. It will collect data from each sensor via SPI and transmit or relay that data to a number of endpoints including a display for the driver, a log file on an SD card, and the remote pit unit. It will ideally perform this data acquisition and display in real time. This will provide the basis for the DAQ system and allow the system to provide the most up-to-date information to the driver and crew, and store the most useful information to the log file.

The main MCU must be fast enough to perform data collection, display, and storage in near-real-time. It must have a number of interfacing options including hardware SPI, and it must have enough GPIO and function pins to provide support for all the current DAQ functions as well as any future additions the system might have. Additionally, the MCU must be reasonably priced and should have low power draw.

2.2.1.2 Accelerometer Module

The accelerometer, along with most of the other sensors used in the system, will be connected to its own microcontroller. The accelerometer itself will accessed using I^2C, and these values will be filtered by an MSP430 MCU and sent over SPI to the Hercules. The data on the bus will be sent in more readily usable format in order to reduce the load on the main MCU. The acceleration data will be used by the Baja team to increase performance in the acceleration portion of the competition.

The LSM303 accelerometer has a measurement range of +/- 2g which is plenty for (wheeled vehicles typically only reach accelerations of 0.4-0.6g, so a low measurement range will reduce cost and increase accuracy).

2.2.1.3 Magnetometer

The ST Micro LSM303 chip has an onboard 3-axis magnetometer as well which the DAQ will use to find magnetic north, and thereby determine the course heading of the Baja car. Since this feature exists on the same chip as the accelerometer, a single MSP430 MCU will be used to collect and relay both acceleration and compass data to the Hercules.

2.2.1.4 Hall Effect Sensor Module

Hall effect sensors increase their voltage output in the presence of a magnetic field. The Hall effect sensor will be used to measure the speed of the vehicle by affixing a permanent magnet to the drivetrain of the vehicle, and using the sensor to read when the magnet passes by a certain point. Once calculated, the speed of the vehicle will be sent to the main MCU and displayed on the on-vehicle display. The sensor will have its own microcontroller to calculate the speed of the vehicle and handle data transmission on the bus. The use of a discrete microcontroller to calculate speed is useful because polling of the sensor has to be done very frequently and at regular intervals in order to accurately determine rotational velocity, if it were to be done by the main MCU it would reduce the sampling rate of the system.

The Hall effect sensor should offer an analog output (as opposed to a switch), and should be through-hole or frame mounted. The attached MCU should be fast enough to poll the sensor at a sufficient rate to accurately determine the vehicle's speed, and the entire control flow should be fast enough to provide data to the main MCU as quickly as possible.

2.2.1.5 Tire Pressure Monitoring System Module

Tire pressure monitoring will be done by short range RF, using a transmitter/receiver pair. Battery powered sensors are placed in the tires which send data repeatedly for the duration of the battery life. Typically this data includes some unique unit ID, the pressure in the tire, and the temperature. The data for all 4 tires can be collected by a single unit and this data can then be repeated over the SPI bus to the main MCU.

2.2.1.6 Sensor Data Bus

The sensors and main MCU will be connected by a single data bus which simplifies programming, allows for future expansion, and reduces pin use on the main MCU. The bus used will be 4-wire SPI, using individual chip select lines to differentiate devices on the bus. For sensors, as soon as their chip select line is pulled low they will exchange data with the master. Other peripheral devices may require input from the main MCU to be sent on the bus before data can be sent.

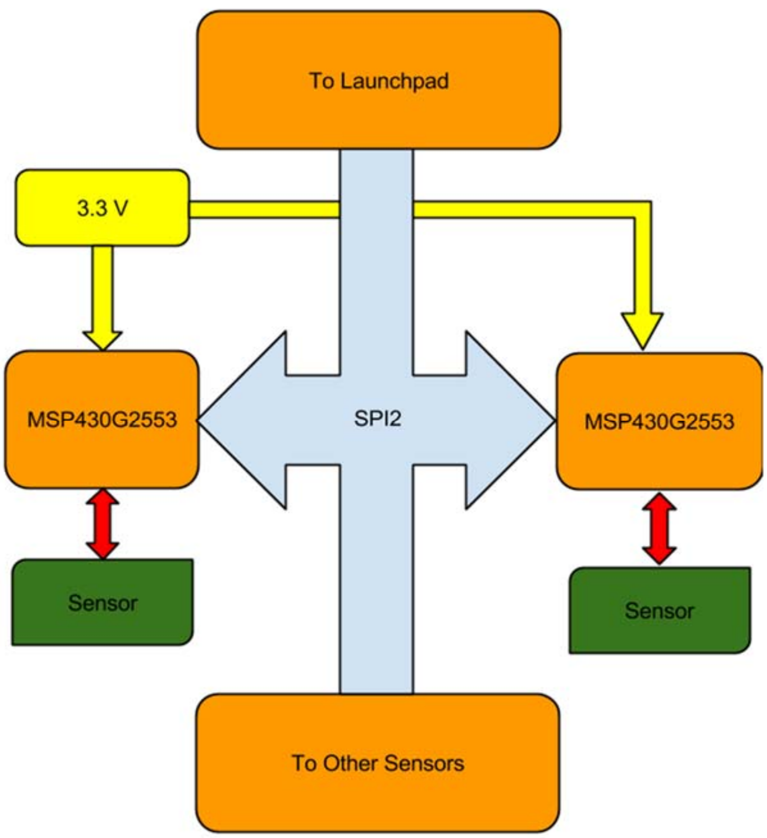


Figure 2: Sensor Network Block Diagram

2.2.1.7 Wireless Transceiver

The wireless transceiver will allow the on-vehicle system to communicate with the pit unit remotely, providing the pit crew with valuable data while the vehicle is racing. This will allow the crew to prepare for pit stops in advance and thereby reduce the overall time spent in the pit. There will be two identical transceivers (one at each endpoint), and 2 way communication will be possible (although information will primarily travel from vehicle to pit)

2.2.1.8 Display

The display will allow the driver to read important information about the vehicle such as the speed. Other messages such as system errors or warnings can also be displayed. This will allow the driver to quickly determine when a pit stop is needed, and will aid the driver in dynamic competitions as performance data can be displayed in real time.

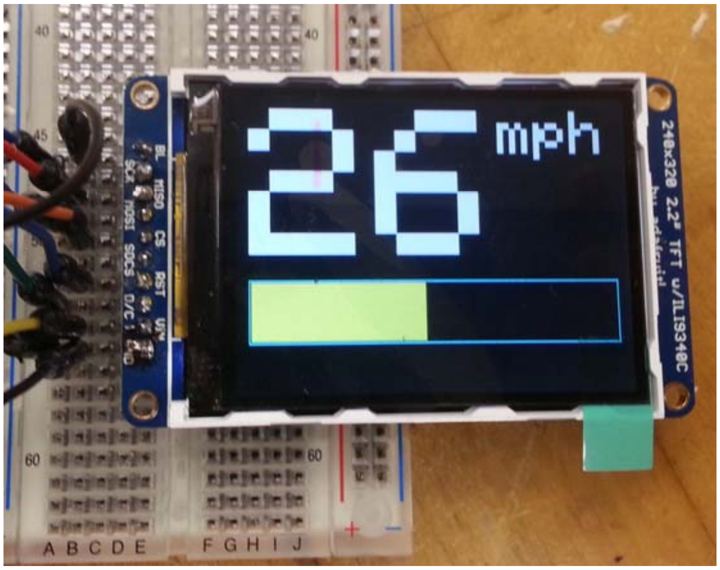


Figure 3: Driver Display Prototype

2.2.1.9 Storage

Removable storage will be used to log data from the DAQ system for later analysis. The log will be stored in CSV format to allow portability and ease of reading and writing. CSV is a very simple open format which is readable by a very wide range of free and proprietary software. An SD card will be used as the removable media due to its ease of implementation and portability.

2.2.2 Pit Unit

The pit unit will collect data from the DAQ system wirelessly and allow the pit crew to read the status of the vehicle remotely. It will consist of a wireless transceiver (identical to the vehicle's transceiver) to retrieve data, and a processor to record or display that data to the crew.

2.3 Performance Assessment

Most of the requirements set forth in the Needs Analysis and Requirements Specifications should be met given the design outlined in its current state in this document. Some requirements were tabled indefinitely due to the solutions not being financially viable, while others were tabled due to, among other things, their lack of usefulness.

2.3.1 Functional Requirements

Most of the functional requirements will be met with the design in its current state as outlined in this document. The system will accurately measure and display the speed of the vehicle (**REQF-001**) and the linear acceleration of the vehicle (**REQF-003**). There will be visual alerts (**REQF-007**) on the displays both on the vehicle and on the pit unit, alerting the pit crew and driver of low tire pressure (TBD based on whether or not the design team can interface with the TPMS sensors). The on-vehicle system will have an easy-to-read display that shows the speed of the vehicle (**REQF-009**). The data collected by the system will be written to an SD card in a *.csv file (**REQF-010**).

Some functional requirements were tabled due to their financial viability, their lack of usefulness, or engineering design concerns. The potential inability to meet **REQF-005** is explored below in **Section 2.2.4** but was ultimately resurrected after further research. The suspension travel measurement (**REQF-006**) was tabled indefinitely due to the cost of implementation using market solutions. It was determined that designing sensors was beyond the scope of this project given the time and material cost it would take to design and manufacture suspension travel sensors. The voice communication between the driver and the pit (**REQF-008**) was tabled due to the range concerns with the wireless transceivers. The design team met with the SAE Baja team and determined that the feature, which would have hampered the rest of the design, had limited usefulness. The Baja team determined that, if need be, a marketed voice communication system would be purchased and installed by them, independent of the DAQ team. The measurement and report of fuel level (REQF-002) has been modified after being informed of fuel tank modification constraints in the SAE international design rules. More information can be found in **Section 2.4.5.**

2.3.2 Non-Functional Requirements

Given the preliminary budget analysis and a budget of $750USD to date, the system should remain within the budget provided by the FAMU-FSU Department of Electrical and Computer Engineering, as per **REQN-001**, barring any unforeseen expenditures and given the complete mitigation of the financial risks outlined in **Section 6.1**. To date, the project is on schedule to be completed before the Auburn, AL competition on April 9th, 2015 (**REQN-002**). However, there could be setbacks that prevent the due date from being met. The on-vehicle system will be powered by its own power source, explored in detail in **Sections 3.1.3** (**REQN-003).** The user interface for the DAQ has not yet been finalized, but the selected hardware is powerful enough to support a fast and responsive UI without issues, and will be powered on or off by a switch located outside of the system's protective housing (**REQN-004**). Since most of the sensors are housed independently of each other and the main MCU box, subsystems can be taken out without affecting the operation of the overall system (**REQN-005**). However, the fact that most of them will be mounted to custom PCBs could make repairing those subsystems a lengthy procedure if extra modules are not on hand

and assembled. The hardware chosen for the system is powerful enough for real-time performance, and any code written will be optimized as much as possible to ensure that the system is running as efficiently as possible (**REQN-006**).

2.3.3 Environmental and Health & Safety Requirements

The DAQ's power source should be recyclable and should last for a large number of recharge cycles (**EHS-001**). The system must not impede the function of any essential Baja systems such as brakes, steering, or electrical systems that are on the vehicle for safety reasons (headlights, brake lights, reverse lights) as per **EHS-002**. The system will be securely attached to the vehicle to prevent creating a safety hazard for other drivers as per **EHS-003**. Project boxes for the main MCU as well as the discrete sensors have been chosen such that they will be protected from the elements as per **EHS-004**.

2.3.4 Usability Requirements

User interfaces for the DAQ system have not yet been finalized, but the selected hardware will be fast enough to support a fast and responsive UI without issue (**REQU-001**). The system's modular design allows the Baja team great freedom in placement of components around the vehicle, and will allow them to remove and replace system components as needed to perform vehicle maintenance or other work. The system will be powered on or off by a switch located on the exterior of the main DAQ housing so that the team can enable or disable the system as needed (**REQU-002**).

2.3.5 Reliability Requirements

The system is expected to run for many hours due to careful selection of components and battery technology (**REQR-001**) and allow for extra run-time for pre-race testing and adjustments, and also to compensate for extra power draw unforeseen by the design team. The modular design of the system will simplify component replacement in the event of component failure (**REQR-002**). The Baja team will be consulted on system mounting locations to ensure that the system is safe and secure on the vehicle (**REQR-003)**.

2.4 Design Process

All major design decisions have been made, and very few major purchasing decisions remain. The project is in the development and testing stage, and the team is preparing to begin the building phase.

2.4.1 Suspension

After discovering the high cost and low availability of suspension travel sensors, the team initially decided to design their own sensors using a potentiometer and steel tubing. After some discussion and research, it was determined that designing sensors was beyond the scope of this project given the time and material cost it would take to design and manufacture suspension travel sensors.

2.4.2 Pit Crew Unit Display

The Pit Unit should display data collected from the system on a screen so that the pit crew can see the status of the vehicle at a glance. The display used will be a 7” TFT LCD display which operates on 12 V power and has a composite video input which can be connected directly to the Pi’s composite out.

2.4.3 Software Development

The team is currently developing the software required to operate each subsystem. Code is being written primarily in C, and portable libraries are being developed and deployed wherever possible to increase software modularity. The primary development environment is TI’s Code Composer Studio on Windows, with Raspberry Pi development being done within Raspbian using GCC. Any code developed by the DAQ team is offered as open source code without a license, and is available on the team’s GitHub which is accessible through the team website. The link is provided below.

https://github.com/hp09d/SAE-Baja-Data-Acquisition

2.4.4 Tire Pressure

The TPMS had originally been tabled due to the lack of availability of marketed solutions for ATV tires as well as the lack of sensitivity of the marketed automobile sensors. Upon further research, the team was able to locate FCC documents pertaining to the sensors which provided more insight into their operation (the sensors are FCC ID# MRXC4N3MF9). They do not require an activation bit sequence as was thought before. Instead, they transmit pressure and temperature data once every 60 seconds when the wheels are in motion. The implementation of this subsystem is again under consideration, however due to some key information still yet to be found, and the additional cost and time requirements of developing this functionality, it is no longer considered a required feature for this year’s project. The state of the tire pressure monitoring system is explored in detail in **Section 3.1.2.6**.

2.4.5 Fuel Level Measurement

Upon reading through the updated rulebook for the SAE Baja series competition, the design team discovered that no holes (even patched holes) are allowed to be in the provided fuel tank. Given the relatively short notice, the design team has decided to forego implementing this subsystem. Many other SAE Baja DAQ teams have faced the same issue, and most have turned to simply implementing a timer instead of actual fuel level measurement, as most solutions seem to fail technical inspection, and thus have to be removed from the vehicles. For that reason, the team has decided to table this subsystem.

2.5 Overall Risk Assessment

2.5.1 Financial Risks

2.5.1.1 *Financial Risk 1: Component Destruction*

Description: Electrical components are very sensitive to the presence of static electricity, and can be permanently damaged by a static discharge. Being mounted on an off-road vehicle, some components could potentially be destroyed by a vehicle roll-over or inclement weather conditions. Components could also be destroyed if they are connected incorrectly, e.g. if a component is connected to the wrong voltage levels or draws more current than it is rated for. The Destruction of any components of the DAQ could potentially consume a large portion of the budget (depending on which component is destroyed) and/or prevent the team from delivering the fully functional system on time.

Probability: High

While destruction by static shock is the least likely situation to occur, it is still a potential problem. Being mounted on an off-road vehicle that will likely be rolled over frequently, components could potentially be crushed depending on where they are mounted on the vehicle. This is the most likely scenario for component destruction.

Consequences: Tolerable to Serious

The consequences of component failure vary depending on which component is destroyed. Destruction of an XBee, a Raspberry Pi, or one or both of the batteries would be catastrophic, as they are the most expensive components in the system. The XBee and the batteries will be mounted on the vehicle. As such, they are the most expensive, most vulnerable components in the entire system. On the contrary, destruction of a small sensor or some other cheap component would have tolerable consequences, as their financial impact is minimal.

Strategy:

When choosing housings for the components found on the vehicle, the structural integrity of the enclosures will be carefully considered, as will their mounting points. Components will be mounted in places that are less likely to bear the full weight of the vehicle upon a roll over. Since components could also be destroyed by static shock, anti-static wrist bands will be worn whenever expensive electrical components are handled. Additionally, special care will be taken when connecting components to power sources to ensure that they are not overloaded with voltage or current.

2.5.1.2 *Financial Risk 2: Component Failure*

Description: At any time, electrical components can fail for a variety of reasons related to manufacturing defects or heavy use. As with component destruction, component failure could potentially consume a large portion of the budget and/or prevent the team from delivering the fully functional system on time.

Probability: Low
Electrical components are manufactured to be very reliable and seldom fail spontaneously under normal use. However, considering the project's intended operating environment, it may be possible for a component's longevity to be affected by heat or vibration.

Consequences: Tolerable to Serious
For the same reasons as component destruction, the consequences of component failure vary from tolerable to serious depending on which of the parts fails. Failure of expensive parts could heavily impact the project budget, while failure of inexpensive parts would have minimal impact. Any failed part will affect the project's progress and may negatively affect its completion time.

Strategy: Since component failure would be caused by manufacturing defects, mitigation of this risk is near-impossible. The design team has no control over the manufacturing of the components used. The only mitigation strategy relating to overuse would be to limit excessive component use, although the amount of use required to cause a spontaneous is very high. As such, failure due to overuse is extremely unlikely.

2.5.1.3 *Financial Risk 3: Project Cost Overrun*

Description: There is a potential that the actual cost of the project may be above that of the preliminary estimations. As with any project, unforeseen issues and changes can occur during project development which could cause the project to cost more than anticipated.

Probability: Low
The preliminary budget estimate came very close to the originally allotted $600, and did not take into account, among other things, pin headers, voltage regulators, component housing, mounting hardware, programmers or antennae. The final budget was increased to $750, but overrun could still occur due to unforeseen expenses or component replacement.

Consequences: Serious
Budget overrun would require the team to explore other funding options to complete the project. If extra funding is not found, or the funding is insufficient then features may have to be cut, or project specifications may not be fully met. If an overrun is anticipated, the team will begin seeking external funds as soon as possible to avoid a delay in project progress.

Strategy: The preliminary budget analysis was done using a worst case scenario, providing some monetary cushion in case unexpected expenses arise. The team has taken steps to attempt to procure some components from Texas Instruments for free, potentially eliminating some of the costs of the project.

2.5.3 Safety Risks

2.5.3.1 Safety Risk 1: *System Mounting Failure*

Description: It is possible that the system could become detached from the vehicle via a mounting failure. This poses a safety risk to not only the driver of the FAMU-FSU Baja car, but every driver in the competition as well as spectators of the competition.

Probability: Low
As there are no components that are particularly heavy or bulky, mounting is expected to be simple and secure. Even so, unexpected complications can occur, and given the operating environment of the vehicle and the location of some subsystems mounting could prove to be more difficult or precarious than anticipated.

Consequences: Catastrophic
The safety of the Baja team's driver as well as the other competing drivers is of the utmost importance in the SAE Baja Series. A system mounting failure could pose potential safety risks to all of the drivers in the competition. As such, a system mounting failure should be avoided at all costs.

Strategy: In order to mitigate this risk, the team will work closely with the SAE Baja team when working on mounting hardware and mounting points. If necessary, a member of the Baja team will provide welding services in order to ensure that the system is safely and securely mounted to the vehicle at all times. Additionally, all screws used will be secured with thread locking adhesives.

2.5.3.2 Safety Risk 2: Water Damage

Description: During competition or in testing, the Baja car may be used in inclement weather or in situations that require navigation of a water hazard. Water poses certain risks to electronics and electrical systems and steps must be taken to ensure the safety of the system and anyone working on it in such conditions.

Probability: Low
The components of the DAQ system will be housed in water resistant housings to prevent water or other materials from interfering with the DAQ's performance. Components will be placed with care to maximize the safety of the components and prevent unnecessary exposure to physical risks.

Consequences: Catastrophic

Water damage to any subsystem could cause it to become inoperable or in the worst case, cause electrical damage to other parts of the system as a whole. Water present in the system may pose safety risks to the vehicle operator or any crew working on the vehicle including electrical shock, fire, or chemical leaks from the batteries. Additionally, this damage may be irreversible and could pose a financial risk as well.

Strategy: In order to mitigate the risk of water damage, great care will be taken to properly house and protect all components. Fuses will be placed in key locations of the power system to prevent any electrical damage or overall system failure in the event of a short circuit. Additionally, great care must be taken by the Baja and DAQ teams when disassembling or modifying any electrical systems on the vehicle.

2.5.4: Design Risks

2.5.4.1 Design Risk 1: Tire Pressure Monitoring System

Description: The team found during their research of potential solutions for the Baja TPMS that there exists virtually no marketed solutions for ATV tires. While the project team was donated 4 automobile TPMS sensors, there is a chance that the team will be unable to interface with the sensors, as they require a certain command to be sent in order for the sensors to return their 69-bit long data stream. If the team is unable to get the sensors provided to return usable data, the TPMS may have to be scrapped, as designing custom TPMS sensors would require more time and money that the design team does not have.

Probability: High
The sensors that were donated to the team, as well as basically any other commercially available automobile TPMS sensors, were designed using a proprietary activation code. As such, they will require a significant amount of reverse engineering and guess work in order to interface with them properly. If the team is not able to successfully interface with the sensors, the tire pressure measurement may be shelved entirely.

Consequences: Tolerable
The tire pressure monitoring is one of the less important subsystems of the DAQ. The Baja team rarely has problems with the tires that are not immediately evident by the driver based on the feel of the car, or the pit crew based on visual inspection - even from a distance. The tire pressure monitoring system on the Baja vehicle was almost scrapped completely upon learning that the sensors available on the market may not be sensitive enough to measure below 10 PSI. With that in mind, failure to produce this part of the system would not be detrimental to the project.

Strategy: The team is in contact with representatives from a company that manufactures replacement TPMS sensors, and has gathered some information regarding data types and frame information. The team will continue to reach out to industry professionals and companies in order to find a workable solution.

2.5.4.2 Design Risk 2: Wireless Vehicle-to-Pit Communication

Description: The team was hard-pressed to find information about the urban data transmission range using the 2.1 dB dipole antenna. The XBee-PRO 900HP is capable of a 4-mile line-of-sight with a 2.1 dB antenna at 200 kbps, but no information was given about the urban range. If the transmission range does not meet the requirements, the pit may only get a transmission once per lap, or whenever the vehicle is in range. Depending on the topology of the endurance course, this could last anywhere from seconds to minutes.

Probability: Moderate
When considering wireless transceivers, the design team came across a benchmark for a wireless transceiver module—the XTend—that listed a 28-mile line-of-sight range using a 2.1 dB dipole antenna, which gave optimism to the team when considering the needs and requirements of the project. During component selection, however, the design team found that the specific model that was rated at 28 miles LOS was very expensive, well beyond the means of the project given the provided budget. Given that, the design team had to choose a less-powerful XBee module, which is only rated at 4 miles LOS, leading to concerns about the unit's ability to reach the required 3-mile LOS range that was given in **CONS-005**, outlined in the Needs Analysis and Requirements Specifications.

Consequences: Tolerable
Since the 28-mile range modules are very expensive--$180 per unit--they are out of the range of attainability for the design team. The modules would cost almost half of the given $750 budget. Even if the team were to obtain outside sponsorships, receiving the amount that is needed to use the XTend modules is unlikely. As such, the team will have to simply work with the hand they were dealt. If the XBee-PRO 900HP does not reach the proper range, the pit crew may only receive data transmissions once per lap, or only when the unit is in range.

Strategy: If the XBee-PRO 900HP does not give the design team the required range, the pit unit may only be able to receive data transmissions once per lap, or whenever the vehicle is in range. Due to the unknown topology of the endurance courses, whether the transmissions occur once per lap or during the whole race will remain unknown until the time of the race. Unless the design team can receive funding for ~$400, the team will simply have to live with what the XBee-PRO 900HP gives in terms of range and transmission accuracy.

2.5.4.3 Design Risk 3: In-house Sensor MCU Printed Circuit Boards

Description: The team has decided to buy discrete MCUs for each sensor and fabricate custom PCBs for each one. If the PCB milling is not possible or fails in-house, the team may have to use perfboard to make their own circuits, which would be fairly time consuming, or seek outsourced manufacturing, which could potentially add a significant amount of cost to the project.

Probability: Moderate
There is a chance that the PCB milling facility at the FAMU-FSU College of Engineering will be unable to fulfill the requirements of the design team. The traces may be located too close together in the CAD design for the milling machine to mill the PCB without issues. Additionally, surface mounting components is not possible at the in-house PCB milling facility.

Consequences: Tolerable
Failure to have the PCBs created in-house could have a large financial impact on the project. This financial impact could be mitigated by changing the design paradigm of the system, which could ultimately affect the timely completion of the project. If the in-house PCB milling is not possible, the design team will be tasked with deciding whether to pay a potentially large sum of money to have PCBs manufactured elsewhere or to build the circuits using perfboard.

Strategy: The actual milling of the printed circuit boards is out of the control of the design team. The team has received the mill tolerances from Donte Ford and will adhere to these specifications in order to maximize the possibility that the in-house PCB milling will be successful. If the in-house milling is not successful, the design team will first look into outsourced manufacturing before deciding to build the circuits by hand on perfboard.

3 Design of Major Components/Subsystems

3.1 Central Data Acquisition Unit

Figure 4 below shows a top level pin diagram on the on-vehicle DAQ system. It shows how components are connected on a pin-by-pin basis, and shows which components are connected to which interfaces on the main MCU.

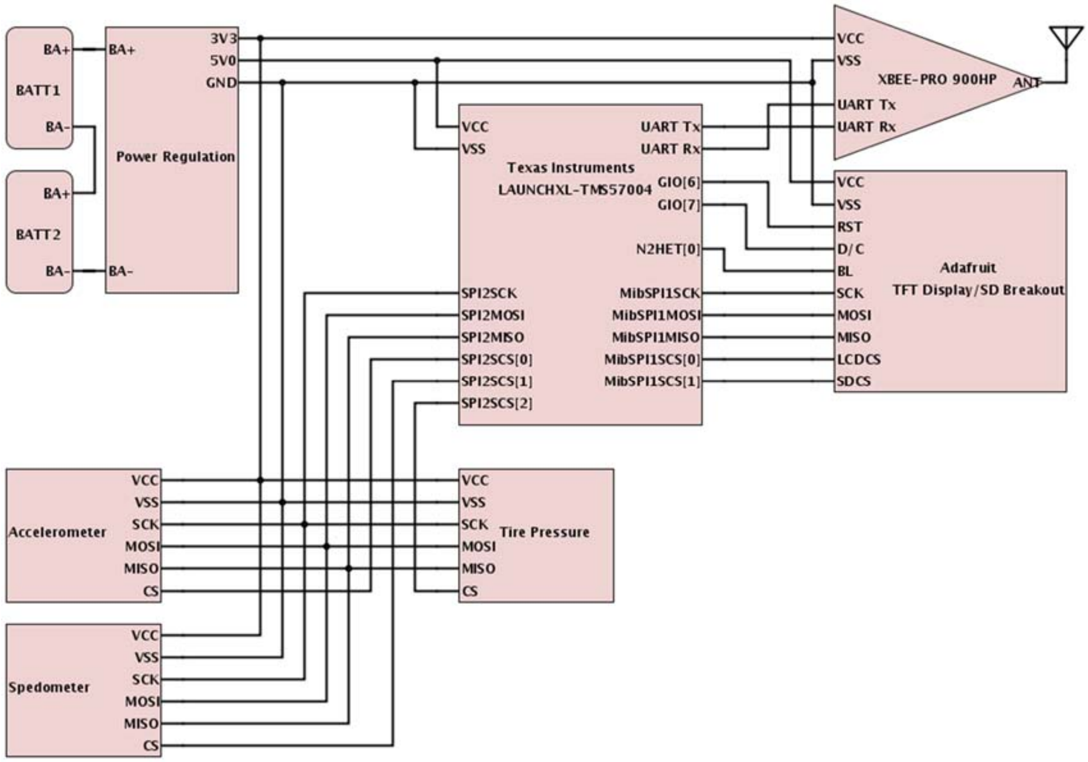


Figure 4: Top-Level Pin Connection Diagram

3.1.1 Main Microcontroller

The microcontroller will serve as the central intelligence to the data collection, and well execute the control flow shown in **Figure 5.** Information from the sensor bus will be sent into the MCU to be organized and stored into log file. Several criteria were considered when choosing the development board that will be used as the MCU.

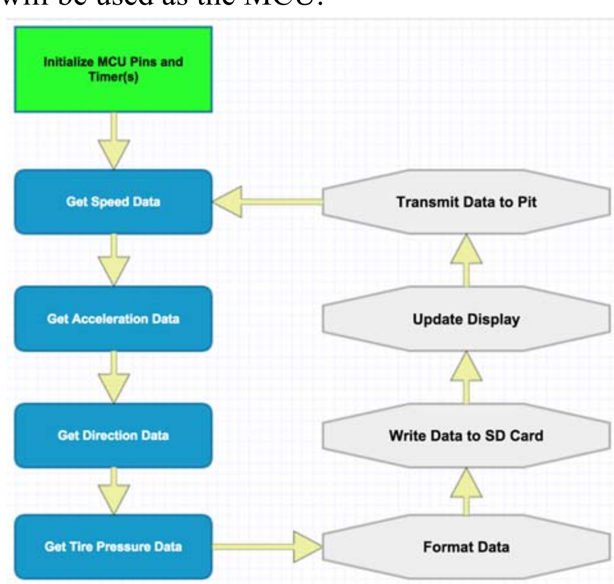


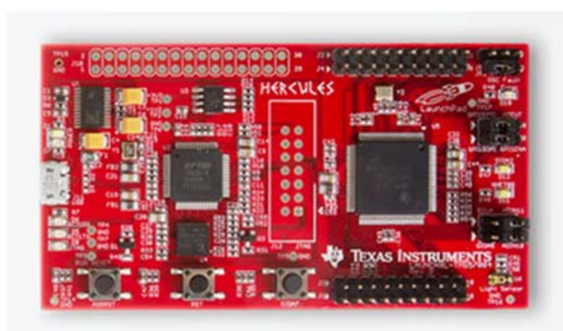
Figure 5: Control flow to be executed by the main microcontroller

There were four development boards considered for use as the central MCU. Of those four, two had features that better suited the needs of the main MCU. These two were the Intel Galileo and the Hercules LAUNCHXL-TMS57004 from Texas Instruments. The Intel Galileo was rejected due to the high cost ($65.25) and the Hercules was chosen because of its automotive and safety focus.

Table 1: Criteria/Justification for Main MCU Selection

Criteria	**Justification**
Flash memory size	The MCU flash memory size must be large enough to accommodate software written for the DAQ
GPIO pin count	A larger amount of GPIO pins will ensure that there will be enough pins for the sensors and future project expansion.
SPI support	After deciding to use SPI for data communication, choosing an MCU with built in SPI support would shorten and simplify development.
Clock speed	The MCU should be able to collect data in real time. Higher clock speed means less time is spent performing calculations and more time can be spent collecting data.
Price	The limited project budget was considered
Power Consumption	The system will be running off a limited battery supply. Power consumption from the MCU must be kept to a minimum.

The LAUNCHXL-TMS57004 hosts a TI Hercules TMS570LS0432 MCU utilizing the ARM Cortex-R4 architecture and a 32 bit word length. It features dual CPUs in lock-step with error correction on CPU outputs and onboard memory, and can notify the system of errors via a dedicated error state pin. One of the most useful features of this MCU is Multi-Buffer SPI (MibSPI) which stores up to 128 words of queued data to be sent or received while the MCU executes other instructions. Multi-Buffer SPI also includes features such as transfer groups (which can be sent from or received to via interrupts), and interrupt outputs which correspond to successful sends/receives or transmission errors.



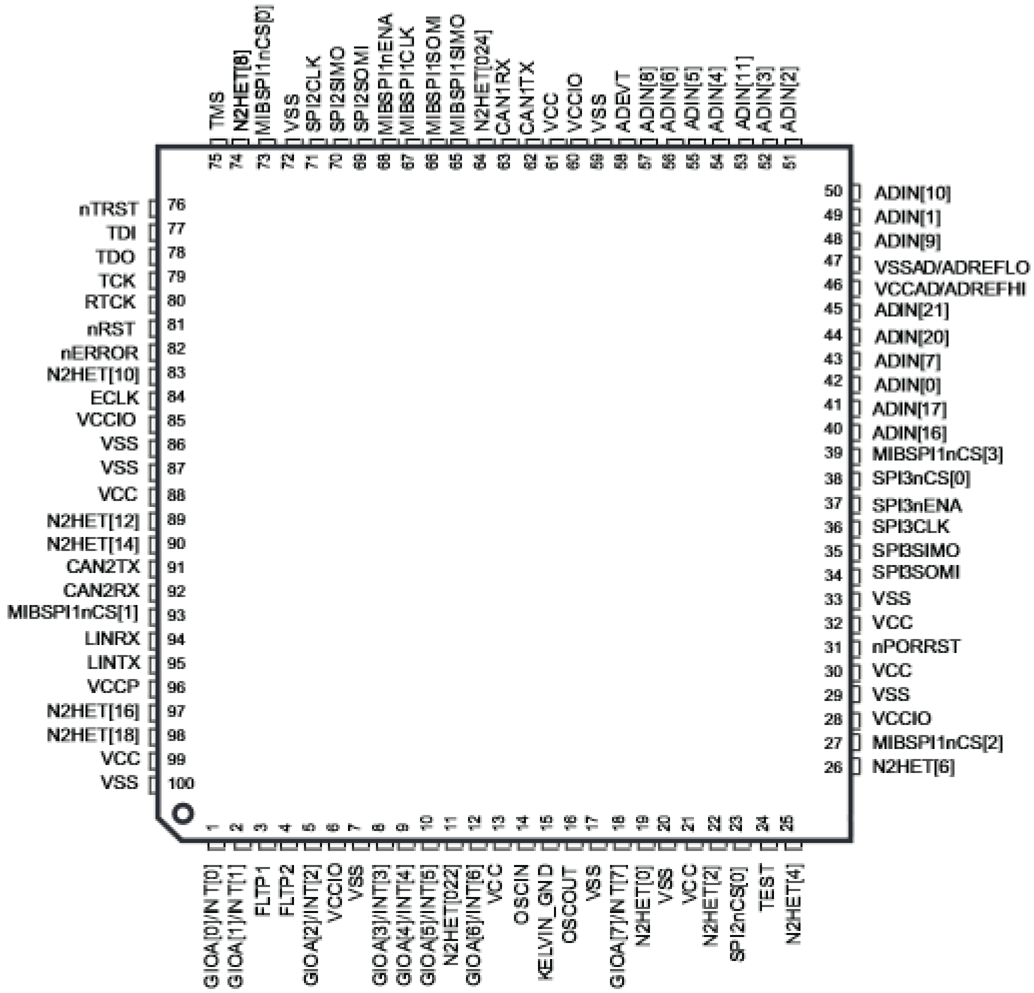


Figure 7: Pin Layout of the TMS570LS0432 MCU

Table 2: Multiplexed pin assignments and control registers for the TMS570LS0432 MCU

100PZ Pin	Default Function	Control 1	Option2	Control 2	Option 3	Control 3
1	GIOA[0]	PINMMR0[8]	SPI3nCS[3]	PINMMR0[9]	-	-
2	GIOA[1]	PINMMR1[0]	SPI3nCS[2]	PINMMR1[1]	-	-
5	GIOA[2]	PINMMR1[8]	SPI3nCS[1]	PINMMR1[9]	-	-
8	GIOA[3]	PINMMR1[16]	SPI2nCS[3]	PINMMR1[17]	-	-
9	GIOA[4]	PINMMR1[24]	SPI2nCS[2]	PINMMR1[25]	-	-
10	GIOA[5]	PINMMR2[0]	EXTCLKIN	PINMMR2[1]	-	-
12	GIOA[6]	PINMMR2[8]	SPI2nCS[1]	PINMMR2[9]	N2HET[31]	PINMMR2[10]
18	GIOA[7]	PINMMR2[16]	N2HET[29]	PINMMR2[17]	-	-
93	MIBSPI1nCS[1]	PINMMR6[8]	EQEPS	PINMMR6[9]	N2HET[17]	PINMMR6[10]
27	MIBSPI1nCS[2]	PINMMR3[0]	N2HET[20]	PINMMR3[1]	N2HET[19]	PINMMR3[2]
39	MIBSPI1nCS[3]	PINMMR4[8]	N2HET[26]	PINMMR4[9]	-	-
68	MIBSPI1nENA	PINMMR5[8]	N2HET[23]	PINMMR5[9]	N2HET[30]	PINMMR5[10]
36	SPI3CLK	PINMMR3[16]	EQEPA	PINMMR3[17]	-	-
38	SPI3nCS[0]	PINMMR4[0]	EQEPI	PINMMR4[1]	-	-
37	SPI3nENA	PINMMR3[24]	EQEPB	PINMMR3[25]	-	-
58	ADEVT	PINMMR4[16]	N2HET[28]	PINMMR4[17]	-	-

Many needed pins such as all MibSPI pins and a number of timer and GPIO pins are already broken out on the standard 40-pin Launchpad headers, but the optional 60pin header will need to be installed in order to access the secondary SPI pins (which are also multiplexed on-chip and will require some configuration to enable).

3.1.2 Sensor Network

The sensors, each driven by their own discrete microcontroller, collect data about the vehicle and transmit that data over via a bus to the main microcontroller. Each sensor requires its own power, and each has its own required input voltage. **Figure 8** shows a general pin-level diagram for devices using the I^2C interface. While each analog sensor may have a different number of analog outputs, **Figure 9** shows a general pin-level diagram of the connections for each sensor module that applies to all of the analog-output sensors. The sensor network will be broken down in detail in this section.

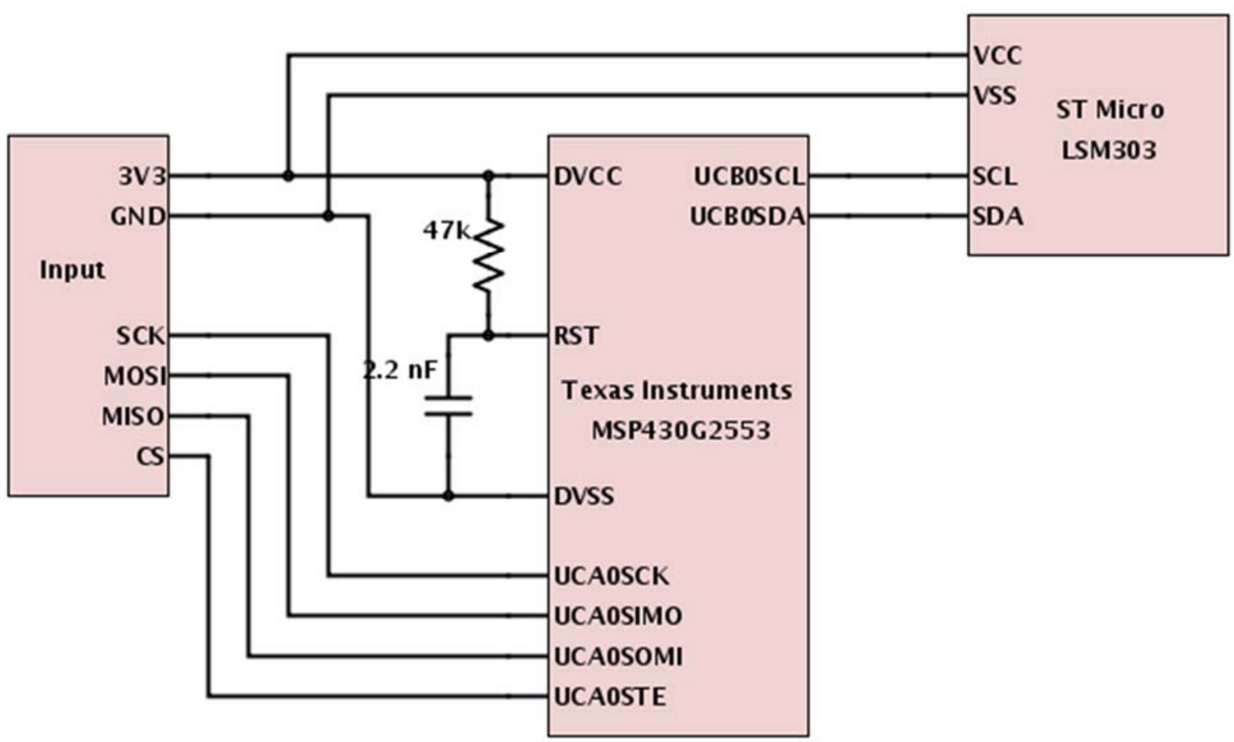


Figure 8: Pin-Level diagram for digital I^2C connection

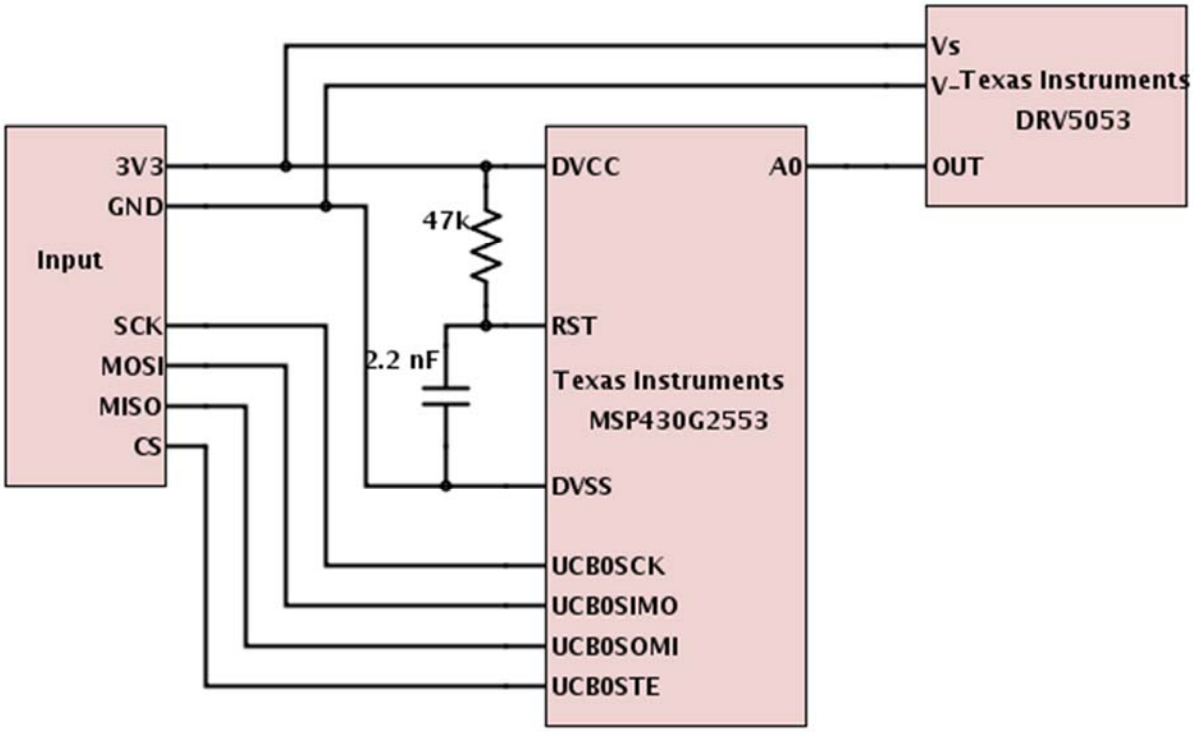


Figure 9: Pin-Level diagram for an analog device

The code in **figure 10** shows how to set up the Hercules' SPI interface to collect sensor data, and **figure 11** shows the slave code running on the MSP430. The full code is available on the team GitHub, accessible from the team website.

```
spiInit();      //Initialize SPI

spiDAT1_t dataConfig1;              //Set up our SPI configuration
dataConfig1.CSNR = SPI_CS_1;            //Using chip select #1 (mibSPI1CS[1])
dataConfig1.CS_HOLD = 0;                //Don't hold CS between transfers
dataConfig1.DFSEL = SPI_FMT_0;          //Select data format 0
dataConfig1.WDEL = 0;                   //No extra delays

uint16_t src[2] = {0x9E,0x50};      //Dummy data to send to slave
uint16_t dst[2];                    //Receive buffer
uint16_t pot = 0;                   //Potentiometer value

while(1){
    spiTransmitAndReceiveData(spiREG1, &dataConfig1,2,&src[0],&dst[0]);     //Send and receive data
    pot = (dst[0]<<8)+dst[1];                                               //Rebuild pot value
}
```

Figure 10: Hercules SPI interface setup

```c
void SPI_Init( void )
{
    WDTCTL = WDTPW | WDTHOLD;
    P1SEL |= BIT0 | BIT4 | BIT5 | BIT6 | BIT7;
    P1SEL2 |= BIT4 | BIT5 | BIT6 | BIT7;

    UCB0CTL1 |= UCSWRST;
    UCB0CTL0 = UCCKPL | UCMSB | UCSYNC | UCMODE_2;
    UCB0CTL1 &= ~UCSWRST;
    IE2 |= UCB0RXIE;
    __bis_SR_register(GIE);
}

int main( void )
{
    SPI_Init();

    while ( 1 );
    return (0);
}

#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR_HOOK (void)
{
    while (!(IFG2 & UCB0RXIFG)) {};
    data = UCB0RXBUF;
    while (!(IFG2 & UCB0TXIFG)) {};
    UCB0TXBUF = SLV_data++;
    __delay_cycles(50);     //change this to something way lower
}
```

Figure 11: MSP430 SPI slave mode setup

3.1.2.1 Data Transmission Bus

The data transmission bus is used to move data around the vehicle, and primarily allows the main MCU to retrieve and collect data from each sensor. The same bus type may also be used to display and record the sensor data.

Table 3: Criteria/Justification for data transmission bus selection

Criteria	**Justification**
Simple Frame Structure	A simple frame structure will simplify code and reduce unnecessary data being transmitted on the bus.
Low Pin Requirement	Conserving pins on the main MCU will allow future developers to add more to the system at a later time.
Onboard Hardware	Hardware or hybrid implementations should be available on-chip and at a low cost. Because there are many devices on the bus, adding hardware can become very costly.

A number of different serial communications standards were considered, including I^2C and CANBus, but the team chose SPI due to its simplicity in slave selection and its full duplex operation. While I^2C was a main contender, its use of an addressed frame was determined to be unnecessary and costly in both code and in data transmissions (SPI's higher pin requirement was decidedly less costly than the use of an addressed frame). Additionally, CANBus requires external hardware and is very feature-heavy and could not be fully or efficiently utilized by the current system. 4-wire SPI is also supported natively in SD cards which greatly simplifies data storage.

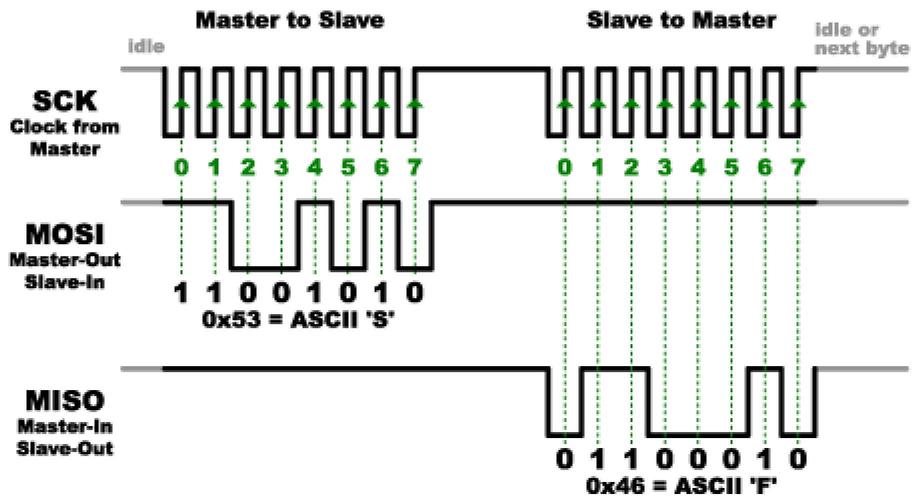


Figure 12: SPI timing diagram
From learn.sparkfun.com

3.1.2.2 Sensor Microcontrollers

In deciding how to deal with the sensor outputs, the design team decided on using a data bus to reduce the number of pins used on the main microcontroller. In order to accomplish this, the analog outputs of the sensors must first be converted to their respective digital representations. For this reason, the design team decided to connect each sensor to its own discrete microcontroller, which will convert their analog outputs to digital representations, change those digital representations into the desired units, and transmit the data to the main microcontroller. In choosing this design paradigm, some of the calculations that would have been done by the main microcontroller can be offloaded onto the sensor microcontrollers.

When deciding on which microcontroller to use for this purpose, the design team was looking for a controller that was low-cost, through-hole mounted, had a minimal number of pins, and had enough flash memory to achieve the task for which it will be used.

Table 4: Criteria/Justification for sensor MCU selection

Criteria	Justification
Low-cost	Since there will be several of these microcontrollers, their cost should be kept as low as possible.
Through-hole Mounting	Through-hole mounting was preferred to make the milling of the PCBs in-house as easy as possible, reducing the risk of having to

	outsource custom PCB milling (see **6.4.2**)
Minimal Pin Count	Minimal pin count was desired again to make the milling of the PCBs in-house as easy as possible, as well as to avoid wasted features.
Flash Memory	The team wanted to be sure that the microcontroller selected had enough flash memory to hold the code that it would be programmed with to avoid wasting money.

The DAQ will utilize Texas Instruments MSP430G2553 microcontrollers for all sensors. It has dual onboard USCIs (one SPI/UART and one SPI/I^2C), and plenty of GPIO and analog inputs.

In choosing to use discrete MCUs, the team will have printed circuit boards milled for each sensor and MCU. See the risk analysis section for details on this process

3.1.2.3 Accelerometer Module

The accelerometer module is to measure (most importantly) the linear acceleration of the vehicle, allowing the Baja team to tune the vehicle to their liking for the specific terrain/event. If the Baja team desires, the accelerometer module may also be able to measure vibrations. The accelerometer module will execute the control flow shown in **figure 13.** When looking for an accelerometer, the design team considered the characteristics outlined in **Table 3.**

Table 5: Criteria/Justification for accelerometer selection

Criteria	**Justification**
Mounting Style	Through-hole mounting was preferred. As with the other PCB-mounted components, using surface mount components was not viable for the in-house PCB milling. See **6.4.2** for associated risks.
Bandwidth	The bandwidth determines how often an accurate acceleration measurement can be taken. The bandwidth should be high enough such that the acceleration can be accurately represented without being
Sensitivity	The sensitivity of an accelerometer determines range of accurate acceleration measurement. Given that the vehicle will likely not reach anywhere near 1g, accelerometers with excessively high sensitivities were ignored, as they are generally more expensive with less sensitive accelerometers.
Cost	Accelerometers can be very expensive and have a wide range of price points. Given the fairly slim budget of the project, the chosen accelerometer was fairly low-cost.

The ST Micro LSM303 is a digital accelerometer and magnetometer package which offers 3-axis acceleration measurement +/- 2g (selectable) with 16 bit resolution. It also offers an onboard 3-axis magnetometer also with 16 bit resolution. The accelerometer will be connected to a discrete MSP430G2553 microcontroller following the basic control flow shown in **figure 13**. The acceleration and compass values will then be sent to the main microcontroller, where it will be written to the log file and sent to the pit unit for display. An option will also be available for the linear acceleration to be displayed on the on-vehicle display, that way the Baja team can measure the vehicle's linear acceleration without having the pit unit present, allowing them to tune the vehicle to maximize acceleration for that portion of the competition.

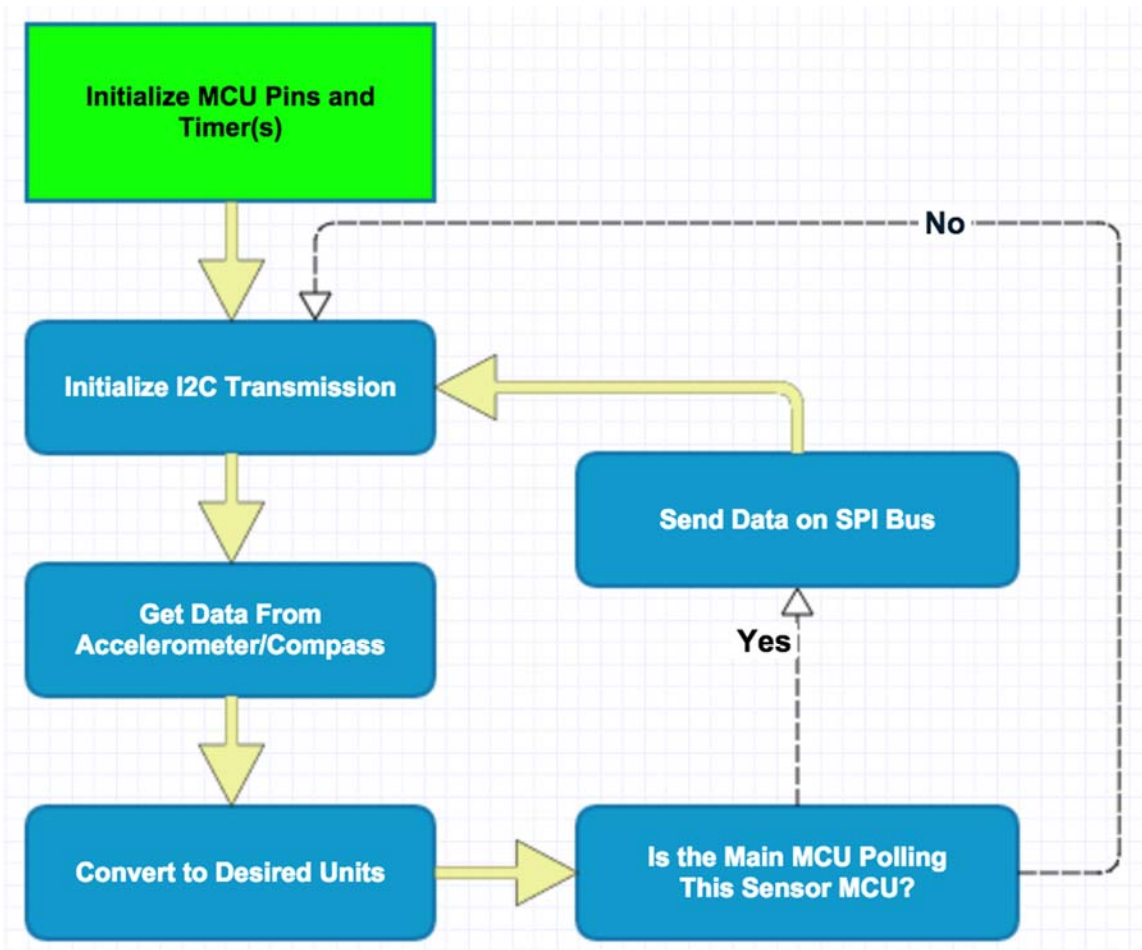


Figure 13: Accelerometer MCU control flow

3.1.2.4 Hall Effect Sensor Module

Hall effect sensors increase their voltage output in the presence of a magnetic field. The Hall effect sensor will be used to measure the speed of the vehicle by affixing a permanent magnet to the drivetrain of the vehicle, and using the sensor to read when the magnet passes by a certain point.

The Hall effect sensor MCU will execute the control flow found below in **figure 14**. Once calculated, the speed of the vehicle will be displayed on the on-vehicle display.

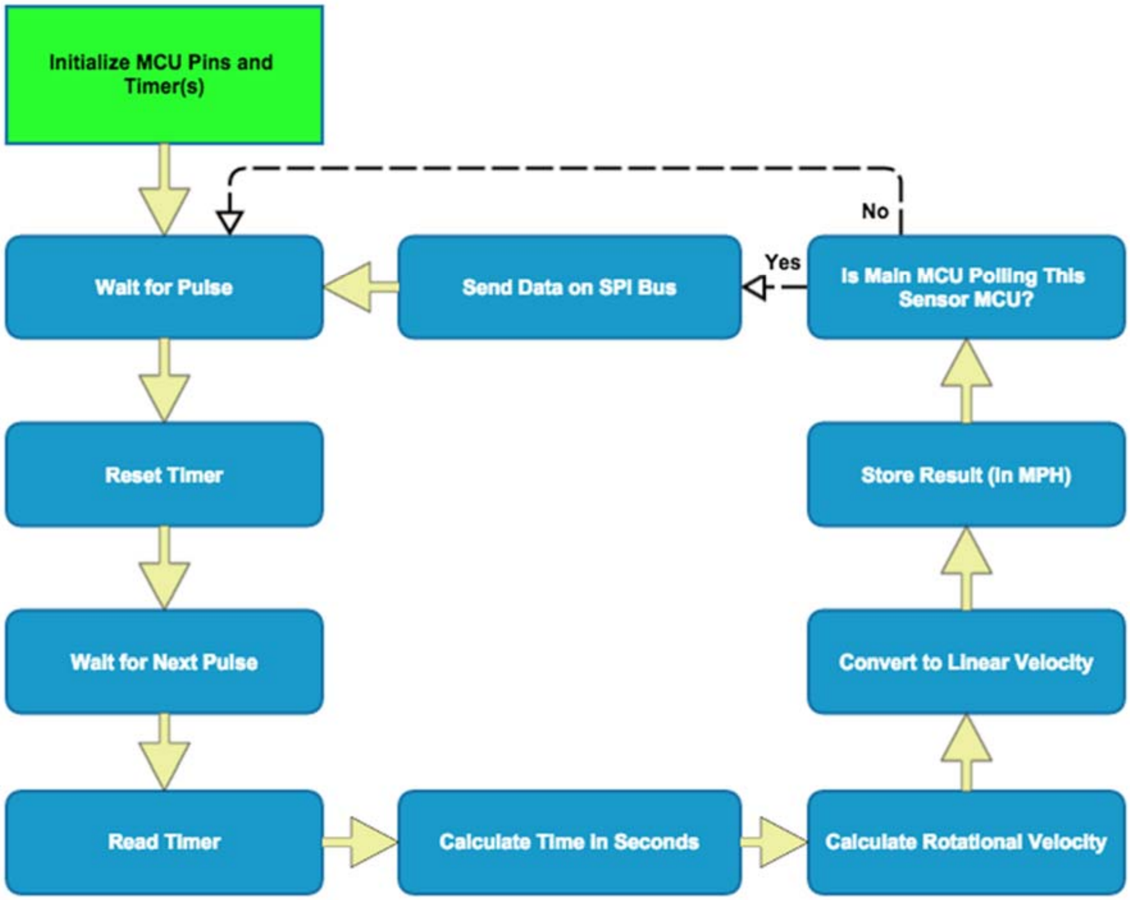


Figure 14: Hall effect sensor MCU control flow

When considering different Hall effect sensors, the design team considered different sensors based on the criteria outlined in **Table 6** below.

Table 6: Criteria/Justification for hall effect sensor selection

Criteria	**Justification**
Through-hole or Frame Mounting	Like the other sensors and sensor MCUs considered, through-hole mounting was desired in order to simplify the milling of custom PCBs due to the risk outlined in **6.4.2**
Analog Output	Some components that are called “Hall effect sensors” are actually switches, which are switched “on” with the presence of a magnetic field, and switched “off” again when the presence of a magnetic field is detected again. The design team desired a Hall effect sensor that output an analog voltage whenever a magnetic field was detected.

Cost	Hall effect sensors can be very expensive depending on how robust (durable) they are as well as the type of output they produce. Industrial Hall effect sensors can cost up to $40/unit.

3.1.2.5 Tire Pressure Monitoring System

When researching the TPMS for the vehicle, the design team found that there were no viable marketed sensors for ATV tires. Since automobile tires are inflated to around 35 PSI, the team thought that the sensors marketed for automobiles (which are fairly expensive, around $25 per sensor minimum) would not be sensitive enough to measure small pressure drops under 10 PSI, the pressure at which the Baja vehicle's tires are inflated. However, a technician at a Dodge/Chrysler/Jeep/RAM Dealership in South Florida tested a set of spare sensors and found that their readings at 6 PSI matched exactly that of a conventional dial tire gauge. Those sensors were ultimately donated to the team. The sensors, along with basically all other automobile TPMS sensors, are activated using proprietary methods. In order to implement the TPMS, which was originally tabled due to the research mentioned above, the design team will have to reach out to the manufacturers in hopes that they will give the team information regarding the activation of the sensors. If the information is not given by the manufacturers, the team will have to do an extensive amount of tinkering and reverse engineering in order to attempt to get the TPMS working properly. This subsystem may still be shelved for future iterations of the DAQ. See **2.5.4.1** for a detailed analysis of the design risk associated with this subsystem.

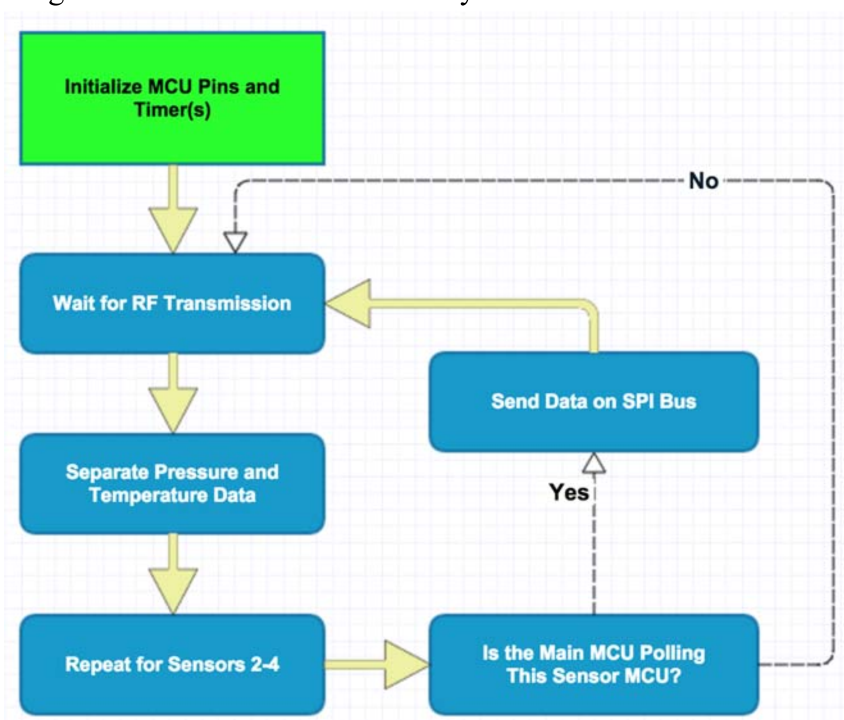


Figure 15: TPMS MCU control flow

3.1.3 Power System

The DAQ system will need its own power source, as per competition rules. The DAQ's power system will consist of two Lithium Iron Phosphate batteries and a circuit to regulate the battery voltage to 3.3 V and 5 V outputs for use by the other subsystems.

Table 7: Power Analysis

Component	Voltage Rail	Active (mA)	Passive (mA)	% active	# of Components	Average Power Draw (mW)
Hercules LaunchPad	5	92		100%	1	445.45
MSP430G2553	3.3	420 uA		100%	4	1.386
LSM303DLHC Accelerometer	3.3	110 uA		100%	1	0.363
Xbee-PRO 900HP	3.3	215	29	30%	1	279.84
DRV5053 Hall Effect Sensor	3.3	2.7		100%	1	8.91
Adafruit LCD Display	5	68		100%	1	340
SD Card Slot	5	66	6.60	30%	1	122
					Total Average Power Draw (mW)	1197.949

The block diagram in **Figure 1** shows the voltages that each subsystem requires, and this system is also shown in the pin level diagram in **Figure 4**.

When considering which battery technology to use, the team considered several factors to make the decision. Since the batteries are the most expensive components of the entire system, it was vital that the team considered all available options and made the right choice first time. The criteria used to evaluate battery technologies is outlined below in **Table 8.**

Table 8: Criteria/Justification for battery selection

Criteria	**Justification**
Safety	Some battery technologies are infamous for having catastrophic failures. For example, Lithium ion and Lithium polymer batteries are subject to explosion or fire if they are overcharged, overdrawn, or connected improperly. Lithium polymer batteries come in pouches, and as such are prone to crushing and puncture, which can lead to a catastrophic failure. Safety is of the utmost importance in the Baja competition. As such, safety factored heavily into the decision.

Price/Capacity	Batteries are very expensive. The batteries will be the most expensive components in the entire system. As such, the $USD/capacity also factored heavily into the decision.
Nominal cell voltage	Nominal cell voltage was a fairly important part of the decision. Some batteries had nominal cell voltages that were far too low or far too high to be viable for the voltage requirements of the components of the system.

The team ultimately decided to use Lithium Iron Phosphate batteries. They come in small form factors, provide a large capacity for a good price, and eliminate all of the safety concerns associated with Lithium ion and Lithium polymer batteries.

3.1.4 Wireless Communication

An RF transceiver pair will be used to create a wireless link between the pit unit and on-vehicle DAQ system. The on-vehicle MCU will send serial data to the transceiver and that data will be repeated on the other end by the paired module and displayed by the pit unit.

The team chose to use the XBee-PRO 900HP due to its long range and low transmit power draw (as compared with its output power). This device operates in the 900 MHz band and offers both serial communication via UART and I/O pin mirroring capabilities. Models are available that offer mesh networking, or simple Point-Multipoint operation. The DAQ system will be using the point-multipoint model and communicating with the XBee via UART on both ends. The XBee can be seen in the pin level diagram in **Figure 4**, which shows all necessary connections to the vehicle MCU.

3.1.5 Display/Storage

Once the main MCU collects and processes information from the sensors, the data is sent through a multi-buffer SPI bus (MibSPI) into a storage unit and an LCD display. An illustration is shown below.

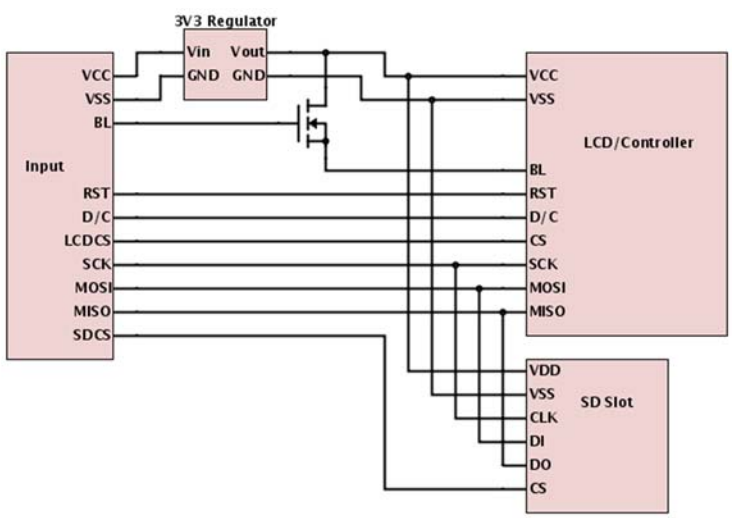


Figure 16: Pin-level diagram for LCD display and SD card

3.1.5.1 Display

The display will be mounted onto the body in the direct line of vision of the driver and will read the speed and acceleration from the main MCU. To meet the needs of this portion of the DAQ, the display will need to have the following characteristics.

- Portable
- LED Backlight
- Low-Priced
- SPI Compatible

The Adafruit TFT LCD 2.2" 18-bit color display and the TFT SPI 2.2" Serial LCD Module Display from Amazon were chosen for consideration. Both screens had the characteristics desired for the on-vehicle display so the deciding factor was the brand as well as the open-source libraries to make writing to the display easier.

The Adafruit LCD display was chosen because of its selection of libraries and examples on the manufacturer's website. This will save time integrating the display into the DAQ by allowing the team to immediately begin interfacing.

The display accepts commands and data over SPI. Writing to the display buffer involves defining a rectangular draw area, then sending 16-bit pixels until drawing is done or until the entire draw area has been filled. Driver code for this display can be found on the team GitHub. The routine in **figure 17** uses a rectangle drawing routine for writing text to the display using a simple 9x8 font.

```
void ili9340DrawChar(uint16_t x, uint16_t y, uint8_t size, char c, uint16_t color){
    uint8_t row,col;
    uint16_t pix, mask, x0, y0, blob;
    for(row = 0; row < 8; ++row)
    {
        pix = console_font_9x8[16*c + 2*row];
        pix = pix<<8;
        pix |= console_font_9x8[16*c + 2*row + 1];
        mask = 0x8000;
        for(col = 0; col < 9;){
            if(pix & mask){
                blob = 0;
                x0 = x+col*size;
                y0 = y+row*size;
                while(pix & mask && col < 9){       //Find "blobs" in the row to optimize drawing
                    ++blob;
                    mask = mask >> 1;
                    ++col;
                }
                ili9340FillRect(x0, y0, blob*size, size, color);
            }else{
                ++col;
                mask = mask >> 1;
            }
        }
    }
}
```

Figure 17: Example code for writing text to the display

3.1.5.2 Storage

Once data has been collected and processed by the MCU, the data will then be written to a storage medium to be used for offline reference. The two storage options considered for this block were USB Mass Storage and Secure Digital (SD).

The benefits and drawbacks of both had to be weighed considerably as they both are well suited to complete this task. Below is a summary of the features of both storage options as they apply to the DAQ storage needs.

Table 9: Criteria/Justification for storage medium selection

Criteria	**USB Mass Storage**	**SD**
Compatibility	Can be used on any modern machine	Many computers do not have built in SD card readers
Portability	Flash drives portable. Longer flash drives can break under stress	Small and portable. Can be easily lost or broken.
Implementation	Requires a host controller	Built in SPI capability

As seen above, one of the downsides to using USB Mass Storage is that host controller is required to communicate between the MCU and the USB device. This could be costly in finances as well as extra development time and complexity. SD supports SPI natively which is also found in hardware on the MCU, greatly easing the hardware development and design.

It is also worth noting the LCD display chosen comes with a built in SD card slot which will save the team some time in development by reducing hardware components, and allowing both the SD card and LCD display to be accessed on one SPI bus using an additional chip select line on the display board.

3.1.6 Central MCU Housing

The center of the on-vehicle system will be housed in a Plano Molding polycarbonate waterproof case. These cases are reasonably priced at $30 and the clamps and waterproof seal will hold up to the harsh operating environment outlined in **Section 1.3**. The housing for the sensor modules has yet to be determined, although smaller versions of this box are being considered as well as custom-made enclosures or generic project boxes. The method for attaching the housings to the vehicle is also to be determined, pending discussion with the SAE Baja team.

When choosing a project box, cost and size were the main factors. Many of the boxes large enough to accommodate the system (at least 10” long and 4” deep) were very expensive, and many boxes in the same price range as the Plano box were far too small. The box will be altered to allow for secure mounting to the vehicle as well as the connection of wires from the DAQ subsystems.

3.2 Pit Unit

The pit unit will be run with a Raspberry Pi Model B+. It will receive transmissions from the main MCU either constantly or whenever the vehicle is in range (see design risk **6.4.2**). The pit unit will run a GUI application that displays all of the pertinent data about the vehicle, which will be received via wireless transmission between the XBee-PRO 900HP modules.

This unit will serve as a supplement to the onboard vehicle data acquisition. As there is minimal design involved, the parts chosen for this module were more straightforward.

3.2.1 Computer

During competition there will be a CPU located in the pit crew station that will receive data wirelessly from the main MCU on the vehicle. This CPU will take the form of a SoC based single board computer to simplify the display of collected data.

The decision to use the Raspberry Pi B+ as the pit crew CPU was made when comparing development boards for the main CPU. The features were impressive but with its optimization for graphic applications, the Raspberry Pi was not a match for what was needed as the DAQ main MCU. Since the pit crew module relies heavily on the display of data to the sideline team members, it was determined that the Raspberry Pi was better suited for this module.

3.2.2 Display

The pit crew will have a display that will output data from the receiving end of the Xbee. During competition, teams are assigned an area on the spot that will serve as their pit crew station. Due to this instability, the display will need to be lightweight and energy efficient for portability. The display will be connected to the CPU by a standard RCA audio/video cable.

4 Test Plan

The test plan devised for the data acquisition system will be following a divide and conquer approach. This means that the data acquisition subsystems will be first tested individually then will be tested again as a whole system. This method identifies faulty modules allows the team to address problems early on as they are found while saving time during the development process.

4.1 Component Test Plan

The first step in the test plan is to evaluate the performance of individual components to determine progression into the next stage. Performance evaluation includes identifying faulty hardware and/or user written software bugs created during initial development.

4.1.1 Accelerometer Module

Initial testing of the accelerometer involves connecting the sensor to a MSP430 Launchpad from TI. After successful connection, the accelerometer will then be turned in the direction which will align one of it's axis to the gravity of earth. Knowing this value to be approximately 9.81 m/s^2 the team can verify the accelerometer module to be functioning by comparing the value given by each axis to the gravity of earth.

4.1.2 Magnetometer

Because both the magnetometer and accelerometer are onboard the LSM303 chip, the testing of the magnetometer will be performed at the same time as the accelerometer test, by pointing the module in different directions and examining the change in the magnetometer output. A stronger magnetic field can also be introduced in order to further test the module.

4.1.2 Hall Effect Sensor Module

A magnet will be attached to a rotating disk. The rotating disk will be set to a slow setting and the rotations will be hand counted. At the same time, the Hall effect sensor will count the number of rotations. The rotations will be compared for accuracy.

4.1.3 Tire Pressure Module

The test plan for this component is to use the sensor to determine the psi of one of the ATV tires on hand. Verify the data using a pressure gage.

4.1.4 Wireless Communication Module

Testing this module involves testing the performance of the hardware out of the box to rule out errors from manufacturing. The XBee Pro modules were each connected to a computer via a USB explorer device. Using the X-CTU software, the XBees were configured and messages sent to one another through a console window. Operational condition was determined by verifying the messages received in one console was the same message sent from the other.

The next form of testing is done by transmitting messages through an XBee from a computer and checking the raspberry pi connected to the receiving XBee to determine whether the signal transmission was a success. The test will be repeated using the Hercules. Lastly, a transmission test between the Pi and Hercules will be performed to ensure proper final operation.

4.1.5 SD card storage

Once the low level driver code is developed, the read and write operations will be tested by using a PC to write test files and to verify that files written back to the card are intact and readable.

4.1.6 Display

Once example code is ported to utilize the Hercules' mibSPI mode, simple shapes such as lines and rectangles can be drawn to the display, confirming its full working condition. More advanced operations such as bitmap and font drawing capabilities are also available.

4.2 System and Integration Test Plan

Once individual system component testing is complete and all the subsystems are deemed to be operational, testing will then move into the integration phase. The system integration phase is separated into two categories of testing; The integration components and system testing.

As part of the integration components testing of the integration phase, aspects of the data acquisition required to combine subsystems will be tested. This includes items such as the SPI communication bus and voltage regulation. Once testing for these integration components are complete, all of the subsystems will be installed on the vehicle and tested as a whole system as part of system testing.

4.2.1 Integration Components

The following are components not directly part of the data acquisition subsystem set but still crucial for subsystem integration.

4.2.1.1 MSP430 ADC

The Hall effect sensor outputs an analog signal which needs to be converted to digital using an ADC. From then on, the data will remain in digital as it is transmitted around the system. For the Hall effect sensor and all other analog devices, the analog signal will be connected to the onboard 10-bit ADC on the MSP430G2553 microcontroller. Testing the built in ADC involves developing baseline code to initialize and read from the ADC, as well as some external hardware to generate a known analog signal.

4.2.1.2 3.3V Regulator

A 3.3V input voltage is required to operate several devices in the DAQ. A switching voltage regulator circuit will be used to provide this voltage from the higher voltage of the battery. To test the voltage regulator, 6-7V supplied to the circuit and the output voltage and the ripple were measured using an oscilloscope.

4.2.1.3 5V Regulator

Several devices also require a 5V input voltage. The testing process for this regulator is the same as the process used for the 3.3V outlined above with the output voltage tested for 5V.

4.2.1.4 SPI

Both the Hercules and MSP430 devices need to communicate using the Serial Peripheral Interface, or SPI. Their respective interfaces can be connected to an oscilloscope and its output viewed in order to determine full working condition. Additionally, slave devices can be connected and the ability to collect and send data can be tested.

4.2.1.5 I^2C

The I^2C serial data bus will be used to collect digital data from the accelerometer. This test will performed by connected the accelerometer to the MSP430 microcontroller. Similar to SPI, I^2C

data transmissions can be examined using an oscilloscope to confirm that the interface is functioning. Additionally, code will be written that collects accelerometer data from the MSP430 and the variables holding this data will be observed to verify functioning data collection.

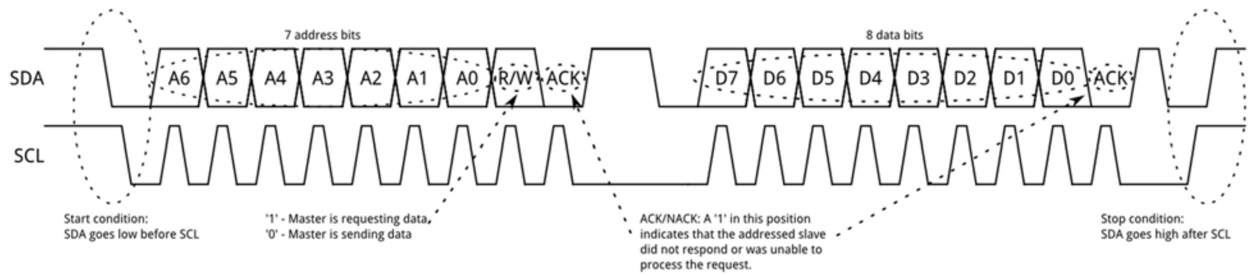


Figure 18: I2C Timing Diagram

From learn.sparkfun.com

4.2.1.6 Battery Life and Power Consumption

The batteries arrived with cell voltage and charge labeled as measured in the factory. To confirm these measurements, a resistive load can be placed across the batteries until they are drained (to around 5.5 volts), and the time recorded to calculate the total charge which can be used with the power analysis table to calculate the total expected battery life of the DAQ system.

4.2.1.7 Project Box

This test is will allow the team to asses the performance of the project enclosure seal when subjected to moisture. The test is performed by placing paper towels inside of the Plano project box and submerging the whole box under a body of water for a small period of time. The box and the paper towels will be examined for any leakage.

4.2.2 System Testing

The following are tests for each DAQ subsystem after vehicle installation.

4.2.2.1 Wireless Communication

Testing for this subsystem begins by manually setting constant sensor values on the Hercules and transmitting these values using the XBee modules to the Raspberry Pi in a lab setting. The data displayed on the Pi will be checked for accuracy.

The next form of testing is done by manually manipulating a sensor on the vehicle and checking an LED light or display on the raspberry pi at a simulated pit crew station to determine whether the signal transmission was a success. The data displayed on the raspberry pi will also be inspected to confirm that the data coming in falls in a realistic range.

4.2.2.2 Accelerometer

Given that preliminary tests are done on the accelerometer module, we can be certain that the module will output correct values when installed on the vehicle. Some sanity checks can be performed to verify that extreme or obviously incorrect values are not being reported to the main MCU.

4.2.2.3 Hall Effect Sensor

The Hall effect sensor module will use the rotations per second value collected from the sensor to calculate the speed of the vehicle in miles per hour. To verify the data calculated by the Hall effect sensor module, a car with an accurate speedometer will run next to the car on a short stretch, and the speed readings will be compared. Any error can be corrected in the MCU’s calculations.

4.2.2.4 Tire Pressure

The purpose of the test is to gauge the performance of the tire pressure sensor once integrated into the rest of the data acquisition system. Before a test drive, the tire pressure for each tire will be recorded using a tire pressure gauge. This value will be compared against the data being displayed for the tire pressure sensors.

4.3 Summary of Test Plan Status

The table below provides a summary of the subsystem testing in progress, the completed system tests, and the systems that still need work. Due to a delay in receiving hardware parts, the development time of the system has been extended into the time originally planned for testing and as a result, testing of the system as a whole will not begin until late February. Individual component testing should not take a significant amount of extra time and any delay will not change the project completion date. See **Appendix A** for more detailed test reporting forms for each of these tests.

Table 10: Test Plan Status

Title	Test Type	Attempt 1	Tester	Result	Attempt 2	Tester2	Result2
Accelerometer Calibration Test	Component	1/9/2015	Chris	Pass			
Hall Effect Sensor Test	Component						
Tire Pressure Sensor Test	Component						
Xbee Baseline Test	Component	12/5/2014	Chris/Dewey	Pass			
Xbee Short Range Test	Component	12/5/2014	Chris/Dewey	Pass			
Xbee Long Range Test	Component	12/5/2014	Chris/Dewey				
Xbee Raspberry Pi - Communication Test	Component	1/16/2015	Hebe	Fail	1/23/2014	Dewey	Pass
Hercules SD Card Test	Component						
Hercules Display Test	Component	1/14/2015	Dewey	Pass			
MSP430 ADC10 Test	Integration	1/25/2015	Chris	Pass			
3.3V Switching Regulator Circuit Test	Integration	2/1/2015	Dewey	Fail			
5V Switching Regulator Circuit Test	Integration	2/1/2015	Dewey	Fail			
Hercules SPI Test	Integration	1/12/2015	Dewey	Pass			
MSP430 SPI Slave Test	Integration	1/21/2015	Chris	Pass			
MSP SPI Slave Mode with ADC10 Test	Integration	1/15/2015	Chris/Dewey	Fail			
MSP430 I2C	Integration						
I2C and SPI MSP430 Test	Integration						
Battery Life Test	Integration						
Plano Waterproof Seal Test	Integration						
MSP430 PCB Test	Integration						
Xbee between Hercules and RPi Test In-Lab	Integration						
Xbee between Hercules and RPi Test On-vehicle	Integration						
Hall Effect Sensor Accuracy Test	Integration						
Tire Pressure Sensor Test	Integration						

5 Spending Report

Senior Design Team E#5 Baja DAQ Spending Report

Order #	Order Location	Item #	Item Description	Price	Quantity	Item Total	Order SubTotal	Tax	S&H	Order Total	Purchase Date
1	Digi	1	XBee-PRO 900HP, P-MP, RPSMA	$39.00	2	$78.00					
		2	2.1dB 900MHz Articulating Dipole Antenna	$9.00	2	$18.00					
							$96.00	*$8.20*	*$13.31*	**$117.51**	11/6/2014
2	Sparkfun	1	XBee Explorer USB	$24.95	1	$24.95					
		2	XBee Breakout Board	$2.95	2	$5.90					
		3	Breakaway Headers - 2.54mm	$1.50	3	$4.50					
		4	10 pin Headers - 2mm	$1.00	4	$4.00					
		5	Raspberry Pi Model B+	$39.95	1	$39.95					
		6	ADXL335 Accelerometer Breakout	$14.95	1	$14.95					
							$94.25			**$94.25**	11/6/2014
3	Mouser	1	Dual Female USB Type A Connector	$1.39	1	$1.39					
		2	ROHM 5 V LDO Voltage Regulator	$1.76	2	$3.52					
		3	ROHM 3.3 V LDO Voltage Regulator	$2.17	2	$4.34					
		4	3M 20 pin DIP Socket	$0.42	8	$3.36					
							$12.61		*$4.99*	**$17.60**	11/6/2014
4	Amazon	1	8GB Micro SD Card	$6.49	2	$12.98					
							$12.98			**$12.98**	11/6/2014
5	Texas Instruments	1	MSP430G2553 Chips	$2.80	5	$14.00					
							$14.00		*$7.00*	**$21.00**	1/2/2015
6	All-Battery.com	1	Tenergy 3.2V 24Ah LiFePO4 Battery	$39.99	2	$79.98					
		2	Tenergy 6.4V 1A 2-Cell Charger	$9.99	1	$9.99					
							$89.97		*$15.18*	**$105.15**	1/9/2015
		3	Special Handling (HazMat Shipping)	$40.00	1	$40.00					
							$40.00			**$40.00**	1/14/2015
7	Amazon	1	Neodymium Magnets, 10pk	$5.99	1	$5.99					
		2	3.5mm to RCA A/V Cable	$4.59	1	$4.59					
		3	12V 7" TFT Display	$26.20	1	$26.20					
		4	Cable Grommets, 30pk	$9.28	1	$9.28					
							$46.06			**$46.06**	1/9/2015
		5	Plano Large Waterproof Case	$50.12	1	$50.12					
							$50.12		*$9.76*	**$59.88**	1/9/2015
		6	Estone Waterproof Enclosure	$3.79	6	$22.74					
							$22.74			**$22.74**	1/9/2015
8	Adafruit	1	Triple-axis Accelerometer+Magnetometer	$14.95	1	$14.95					
		2	2.2" 18-bit color TFT LCD display w/ microSD	$24.95	1	$24.95					
		3	Discount Code	-$3.99	1	-$3.99					
							$35.91		*$10.94*	**$46.85**	1/23/2015
9	Mouser	1	30pin Male Header	$1.47	3	$4.41					
		2	30pin Female Header	$2.13	3	$6.39					
		3	20pin Female Header	$1.53	2	$3.06					
		4	3M 20 pin DIP Socket	$0.58	3	$1.74					
		5	47K Resistor	$0.054	10	$0.54					
		6	2.2 nF Ceramic Capacitor	$0.08	10	$0.80					
		7	100 uF Capacitor	$0.12	10	$1.20					
		8	330 uF Capacitor	$0.18	5	$0.90					
		9	150 uH Inductor	$0.42	5	$2.10					
		10	1N5819 Schottky Diode	$0.13	5	$0.65					
							$21.79		*$4.99*	**$26.78**	1/24/2015
								Project Budget:		$750.00	
								Total Expenditures		$610.80	
								Remaining Budget:		$139.20	

6 Conclusion

The design for this data acquisition system involves sensors boards with microcontrollers placed in various parts around the vehicle. The sensors will send data over an SPI bus to a main MCU for data processing and storage. All the data will be stored on an SD card and sent wirelessly from the vehicle to the pit crew. The speed value will be displayed on an LCD screen within line of sight of the driver. The pit crew system will contain a CPU that listens for the wireless communication and displays that information during race time.

Each module has been carefully chosen to suit the needs of the task it was meant to perform. The portability and energy efficiency are important characteristics for a project of this nature and has been taken into consideration for all hardware chosen for each module.

Project risks have been weighed for the DAQ system with the majority of them stemming from what will be the DAQ systems operating environment. As such, the team is prepared to take precautions in protecting the components against environment wear and tear and will have a plan outlined to handle the consequences of component failure.

Appendix A

Scheduled Test Reporting Form

Test Item: Accelerometer Calibration
Tester Name: Christopher Riker
Test Date: TBD
Test Time: TBD
Test Location: CoE Senior Design Lab

Test No: 1
Test Type: Component Test
Test Result: TBD

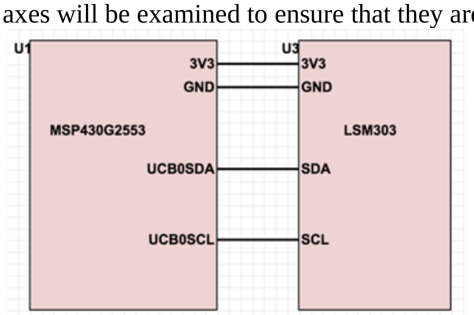
Test Objective:
The objective of this test is to determine whether or not the accelerometer outputs the correct values when a known, constant acceleration is placed on it.

Test Description/Requirements:
Requirements:

1: LSM303 Accelerometer/Magnetometer
2: MSP430G2553 Microcontroller
3: MSP-EX430G2 LaunchPad
4: MSP430G2553 Microcontroller
5: Breadboard

Process:

The LSM303 accelerometer will be connected to the MSP430 LaunchPad, which will be configured for I2C master mode. The accelerometer will be placed in a breadboard and hooked up using the pin diagram found in **figure**. The accelerometer will then be turned such that the gravitational acceleration from the earth is applied to each of its axes. The output of these axes will be examined to ensure that they are equivalent to 9.81 m/s^2.



Anticipated Results:

The LSM303 accelerometer will output 1G of acceleration for each of the axes when they are subjected to the gravitational acceleration from the earth.

Requirement for Success:

The test will be considered a success if the accelerometer accurately measures the acceleration due to the earth's gravitational force.

Actual Results:

TBA

Reason for Failure:

TBA

Recommended Fix:

TBA

Other Comments:

Scheduled Test Reporting Form

Test Item: Hall Effect Sensor Basic Operation
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: CoE Senior Design Lab

Test No: 4
Test Type: Component Test
Test Result: TBD

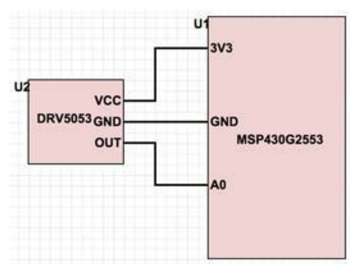
Test Objective:
The objective of this test is to verify that the hall effect sensor is able to detect the presence of a magnet as it passes by the sensor. This is to ensure that the sensor is operational and will be able to be used to detect a magnet passing by that is attached to a wheel or axle.

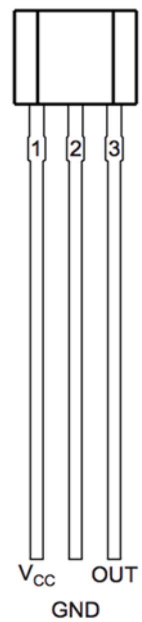
Test Description/Requirements:

Requirements:

1: One DRV5053 analog bipolar hall effect sensor
2: One neodymium magnet
3: One MSP-EXP430G2 LaunchPad development board

Process:
The MSP430 ADC10 module will be configured for use. The hall effect sensor will be connected to the LaunchPad on a breadboard. The VCC pin will be connected to the 3.3V output of the MSP-EXP430G2, and the GND pin will be connected to the ground. The OUT pin will be connected to the analog input port A0 (pin 1.0). Some code will be written to do A/D conversion on the hall effect sensor output continually. With the debugger running, a neodymium magnet will be passed in front of the hall effect sensor. The ACD output will be observed to determine whether or not it rises when the magnet is in front of the sensor.





Anticipated Results:
The voltage output of the hall effect sensor will rise as the magnet moves close to the sensor. The voltage output will peak when the magnet is directly in front of the sensor.

Requirement for Success:
The test will be successful if the voltage output of the sensor peaks when the magnet is directly in front of the sensor, telling the tester(s) that the hall effect sensor is properly detecting a magnetic field.

Actual Results:
TBA

Reason for Failure:
TBA

Recommended Fix:
TBA

Other Comments:

Scheduled Test Reporting Form

Test Item: Tire Pressure Sensor Test
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: SAE Portable

Tester ID No: 5174
Test No: 2.5
Test Type: Component/Integration Test
Test Result: N/A

Test Objective:
The objective of this test is to test the tire pressure sensor to determine its operational condition.

Test Description/Requirements:
Requirements:

1- Schrader C4N3MF9 TPMS Sensors
2- TPMS Sensor readers
3- Pressure Gauge
4-ATV Tire

Process:
The TPMS sensor will be installed into the ATV tire using the resources available online for Tire Pressure Monitoring system installation. After installation, tire pressure will be recorded using a commercial tire pressure gauge. Using the TPMS sensor readers, the value of the tire pressure coming from the Schrader TPMS sensors will be recorded. The two tire pressure values gathered will be compared to determine whether the tire pressure sensor is outputting correct data.

Anticipated Results:
The tire pressure sensors output a value comparable to the one recorded with the pressure gauge. There may be a slight difference due to the varying sensitivity of the two tire pressure measurement tools.

Requirement for Success:
In order to be a success the tire pressure sensor must read a consistent value similar to the one gathered from the tire pressure gauge.

Actual Results:
TBA

Reason for Failure:
TBA

Recommended Fix:
TBA

Other Comments:

Scheduled Test Reporting Form

Test Item: XBee Baseline Wireless Transmission Test
Tester Name: Dewey Williams & Christopher Riker
Test Date: December 10th, 2014
Test Time: 4 PM
Test Location: CoE Senior Design Lab
Test No: 6
Test Type: Component Test
Test Result: Pass

Test Objective:
The objective of this test was to determine whether or not the XBees arrived in an operational condition, and were able to send/receive transmissions.

Test Description/Requirements:
Requirements:

1: Two XBee-PRO 900HP wireless transceivers
2: Two computers with X-CTU installed
3: Two XBee USB Explorers

Process:

The XBee-PRO modules were connected to the computers via the USB explorers. The testers first verified that the XBees were responding to commands by sending the command +++, which puts the XBee in AT mode. A unique network identifier was then configured to prevent interference with other devices in the area. The XBees were then connected to the network using the connect button. The console window of one XBee was used to send packets from one XBee to the other, and vice versa. The testers confirmed that the packets were received and were transmitted without error.

Anticipated Results:

The XBee modules will return "OK" when placed in AT mode. After that, the XBee modules will send and receive packets to one another without error or packet loss.

Requirement for Success:

The test will be a success if the XBee modules both return "OK" when placed in AT mode, and if they both send and receive packets to one another without error or packet loss. Upon success, the XBee modules will be declared operational and no further action will be taken.

Actual Results:

The XBee modules both returned "OK" when plated into AT mode, and the devices were able to exchange packets without error in the transmissions.

Reason for Failure:

Test passed

Recommended Fix:

N/A

Other Comments:

Scheduled Test Reporting Form

Test Item: XBee Short Range Test
Tester Name: Christopher Riker & Dewey Williams
Test Date: 12/5/2014
Test Time: 4 PM
Test Location: Innovation Park

Test No: 7
Test Type: Component Test
Test Result: Pass

Test Objective:
The objective of this test was to test whether or not the XBee-PRO 900HP modules would send and receive errorless packets at a distance of 0.25 miles (non-LOS). This was a preliminary test for a longer range test.

Test Description/Requirements:
Requirements:

1: Two XBee-PRO 900HP wireless transceivers
2: Two XBee USB explorers
3: Computers with X-CTU installed

Process:
Since the XBee modules were already configured with a unique network identifier, they required no setup other than to be connected to the tester's computers. The testers found 0.25 miles of space in the Innovation Park area. The space found did not allow the testers to get a line of sight between the XBee modules. Once the testers were in their respective spots, X-CTU was used to send packets from one XBee module to the other. Both long and short messages were sent to test whether or not large packets of information could be transmitted over this range. After testing transmission, the testers compared their X-CTU consoles to ensure that the data was transmitted without error. The range of the testing area was measured with an automobile odometer to be 0.25 miles.

Anticipated Results:
The XBee modules were expected to be able to send and receive packets at this range with little to no error.

Requirement for Success:
The test was considered a success if the testers were able to send packets to each other through X-CTU and those packets were received without packet loss or error.

Actual Results:
The XBee modules were able to send and receive packets to and from one another without error.

Reason for Failure:
TBA

Recommended Fix:
TBA

Other Comments:
There was one small anomaly in one of the XBee module's console windows that was not associated with any message that was sent by the other. It is unknown what the cause of this anomaly was.

Scheduled Test Reporting Form

Test Item:	XBee Long Range Test	
Tester Name:	TBD	
Test Date:	TBD	Test No: 8
Test Time:	TBD	Test Type: Component Test
Test Location:	TBD	Test Result:

Test Objective:
The objective of this test is to test the maximum non-line-of-sight range of the XBee-PRO 900HP modules. This will help the design team determine how often the pit will lose communication with the vehicle.

Test Description/Requirements:
Requirements:

1: Two XBee-PRO wireless transceivers
2: Two XBee USB explorers
3: Two computers equipped with X-CTU

Process:

The XBee will be connected to the computers, as done before in other tests. The testers will travel to the St. Marks trail. One tester will remain at the trailhead, while the other follows the trail in an automobile to the 1 mile marker. If successful, the tester will then move to the 2 mile marker, and so on. The vehicle's odometer will be used to get a close estimate as to the range achieved by the modules.

Anticipated Results:

The team desires a 3-mile maximum range, as this is the worst case possible in competition conditions. Since there is not much documentation available for non-line-of-sight ranges for the XBee-PRO 900HP, there is no quantified expectation of maximum range.

Requirement for Success:

The test will be considered a success if the XBee modules can send and receive errorless transmissions up to a range of 2-3 miles.

Actual Results:

TBD

Reason for Failure:

TBD

Recommended Fix:

TBD

Other Comments:

Scheduled Test Reporting Form

<u>Test Item</u>: Xbee - Raspberry Pi Communication
<u>Tester Name</u>: Dewey Williams
<u>Test Date</u>: 1/23/2015
<u>Test Time</u>: 8 PM
<u>Test Location</u>: CoE Senior Design Lab

<u>Test No</u>: 9
<u>Test Type</u>: Component Test
<u>Test Result</u>: Pass

<u>Test Objective</u>:
The objective of this test is to connect an XBee module to the Raspberry Pi so that transmissions may be read by the operating system.

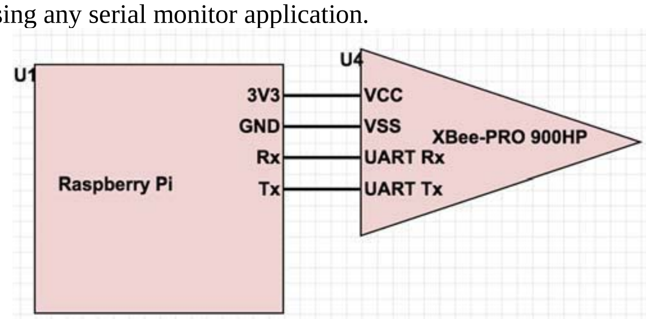
<u>Test Description/Requirements</u>:
Requirements:

XBee Module
Raspberry Pi running Raspbian
XBee Explorer
A host to send test messages

Process:

First, attach an XBee module to a working host (such as a PC or a microcontroller). Next, connect another XBee to the XBee Explorer and connect it to the Raspberry Pi via USB. Use a serial monitor application to attach to the Explorer's port (by default, the explorer will appear as "/dev/ttyUSB0"), and verify that test transmissions are properly sent and received.

Another method is to use the Pi's UART to connect to the XBee. Use of the UART by applications requires disabling some default configurations in Raspbian (more information on this can be found online). The UART is located by default at /dev/ttyAMA0. By connecting the XBee to the Tx and Rx pins on the Pi's expansion header, the XBee can be reached using any serial monitor application.



<u>Anticipated Results</u>:

The XBee will connect to the Raspberry Pi and respond to commands and transmitted data.

<u>Requirement for Success</u>:

The test is a success if the Raspberry Pi is able to send and receive messages via the XBee at the OS level. This is required in order to later develop the GUI application to display data from the vehicle.

<u>Actual Results</u>:

The Pi was able to receive messages from another XBee attached to a PC running X-CTU. The USB connection was tested using a serial monitor application, and UART was tested using a short C program based on example code found online. The Pi was also able to send commands to the XBee (such as +++ for AT mode), as well as send messages to the other host.

Scheduled Test Reporting Form

Test Item: Hercules SD Card R/W
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: College of Engineering Lab

Test No: 10
Test Type: Component Test
Test Result: N/A

Test Objective:
This test will establish the capability of the Hercules MCU to read and write data to/from a microSD card. It will require the development of new code based on an existing file system framework.

Test Description/Requirements:
Requirements:
TI Hercules TMS570LS04x MCU
microSD card and slot

Process:
Connect the pins of the microSD card to the SPI1/mibSPI interface of the Hercules. Using FATFS to handle high level file operations, add the necessary low level driver code to send commands and data to the SD card. Verify that data can be read effectively by using a debug tool or another data output medium to display the collected data. Verify that data can be written effectively by performing a write operation using the Hercules and checking the data using a PC or another known-good card reading device. Circular reads and writes using the Hercules alone may also be tested to check the integrity of arbitrary data written to the card.

Anticipated Results:
Pending software and driver code development, SD functionality should be achieved using the above method.

Requirement for Success:
In order for this test to be a success, the final code should be able to read files created on a PC, and should be able to write files to be read with a PC using a FAT file system. The files should be of arbitrary length and content as this code may be used in different DAQ system operations in the future.

Actual Results:
TBA

Reason for Failure:
TBA

Recommended Fix:
TBA

Other Comments:

Scheduled Test Reporting Form

Test Item:	Hercules + ILI9340 Display Test	
Tester Name:	Dewey Williams	
Test Date:	1/14/2015	Test No: 11
Test Time:	1 PM	Test Type: Test
Test Location:	CoE Senior Design Lab	Test Result: Pass

Test Objective:
The objective of this test is to develop code for the Hercules that will successfully initialize and write pixel data to a ILI9340-driven LCD display.

Test Description/Requirements:
Requirements:

TI Hercules TMS570LS04x with mibSPI capabilities
ILI9340-driven display (We used a 2.2” TFT LCD Display from Adafruit)

Process:

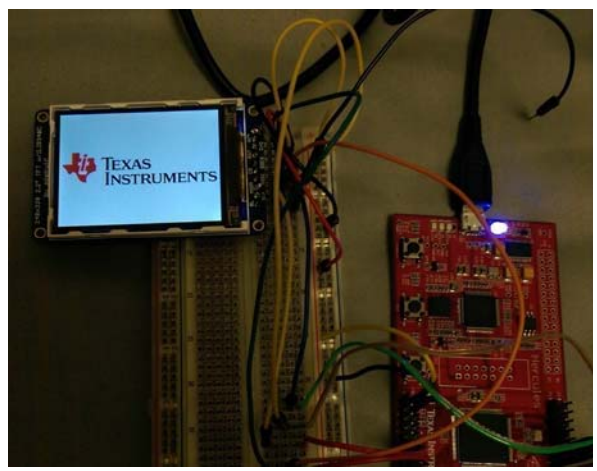
Following the ILI9340 datasheet and example code from Adafruit (written for the ATMega328), develop code to properly reset and initialize the display. Use the SPI1/mibSPI interface to attach to the data and clock lines, and two GPIO pins to connect to the D/C and RST lines on the display (additionally, the backlight can be PWM driven if desired). Configure the SPI interface and transfer groups as desired using HalCoGen (at least one transfer group must be a single 8 bit word as in traditional SPI for sending commands), and verify that the display turns on in an initialized state. Use the datasheet and example code to insert pixel data into the display buffer for output on the display.

Anticipated Results:

The display will be properly reset and initialized on startup of the Hercules, and pixel data will be sent and displayed on the screen.

Requirement for Success:

In order to be a success, the driver code should be able to reset and initialize the display, send commands to manipulate screen output, and send arbitrary pixel data to be displayed.



Actual Results:

The Adafruit example code was partially ported to be used on the Hercules by simply replacing AVR-specific function calls with those provided by HalCoGen. To verify operation, simple lines were drawn and the “rotate” command was sent to the display. Additionally, a full resolution bitmap was uploaded to the Hercules and drawn on the display using these techniques. The test was a success.

Reason for Failure:

N/A

Recommended Fix:

N/A

Other Comments:

The developed code was further optimized to utilize the mibSPI more effectively by prebuffering pixel data into 24 16-bit buffers, which greatly increased the performance of drawing large shapes and bitmaps. Additionally, text writing capability was added for writing data or text to the display. Development of this code is ongoing.

Scheduled Test Reporting Form

Test Item: MSP430 ADC10
Tester Name: Christopher Riker
Test Date: 1/25/15
Test Time: 5 PM
Test Location: CoE Senior Design Lab

Test No: 12
Test Type: Integration Test
Test Result: Pass

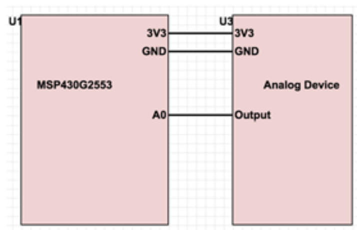
Test Objective:
The objective of this test is to demonstrate the ADC functions of the MSP430G2553 and to generate baseline code to be used in sensor applications.

Test Description/Requirements:
Requirements:

MSP430G2553 microcontroller
Flash programmer and debugger (such as an MSP-EXP430G2 Launchpad)
Analog device (such as an analog hall effect or potentiometer)

Process:

Create a program using example code and other resources which will receive data from the ADC and store it to a variable at some constant interval. Use a debugging utility to view the contents of variables and registers to verify that the collected information is correct.



Anticipated Results:

Data stored in the variable will reflect the position of the potentiometer or the position of the accelerometer.

Requirement for Success:

In order for this test to be a success, the MCU must be able to activate and retrieve data from the ADC, and store that data for later use. The ADC should be able to capture the full range of the attached analog device with a reasonable degree of accuracy.

Actual Results:

The MSP430 was connected to a potentiometer on ADC channel 0 (pin 1.0), and was programmed to continually activate and retrieve data from the ADC. The potentiometer was moved to various positions and the debugger was used to verify the data gathered from the ADC. For each position, the captured data was accurate within a reasonable margin, so the test is considered a success.

Reason for Failure:

N/A

Recommended Fix:

N/A

Other Comments:

Code samples for this test are available in the full report. This code will be assembled into an easy to use driver library to and included in the software for each analog sensor MCU.

Scheduled Test Reporting Form

Test Item: 3.3V Switching Regulator Circuit
Tester Name: Dewey
Test Date: 2/1/2015 Test ID No: 13
Test Time: 8 PM Test Type: Integration Test
Test Location: CoE Senior Design Lab Test Result: Failed

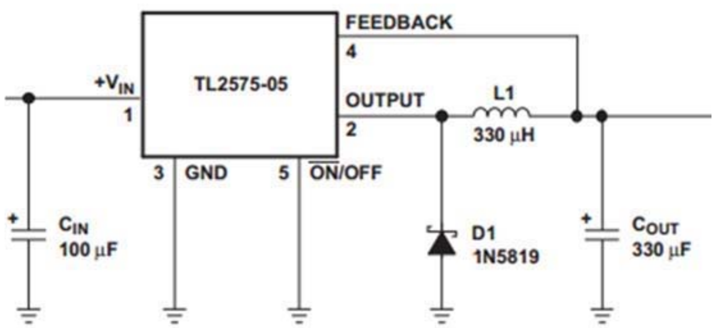
Test Objective:
The objective of this test was to determine whether or not the 3.3V switching regulators supplied the correct voltage with a minimum amount of ripple.

Test Description/Requirements:
Requirements:

1: One TL2575 3.3V switching regulator
2: Oscilloscope
3: Power source
4: One 100uF capacitor
5: One 330uF capacitor
6: One 150uH inductor
7: One 1N5819 Schottky diode

Process:

The TL2575 switching regulator was hooked up into the circuit as shown in **figure (**5V regulator shown, same circuit). A power supply was used to supply the regulator with ~6-7V of input voltage. An oscilloscope was then connected across C_{out} and the output voltage and ripple were observed.



Anticipated Results:

The regulator will regulate the input voltage to 3.3 volts with a ripple of 50mV - 150mV.

Requirement for Success:

The test was a success if the regulator sufficiently regulated the input voltage to 3.3V, and minimized the amount of ripple. The data sheet for the IC specifies an achievable ripple of 50mV to 150mV.

Actual Results:

The voltage regulator sufficiently regulated the voltage input voltage to 3.3V. However, there was a very large amount of ripple (800mV_{PP} - 1V_{PP}).

Reason for Failure:

Insufficient output filtering, improper testing environment

Recommended Fix:

Try to add the second-stage ripple filter, measure voltage and ripple again.

Scheduled Test Reporting Form

Test Item: 5V Switching Regulator Circuit
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: CoE Senior Design Lab

Test No: 15
Test Type: Integration Test
Test Result:

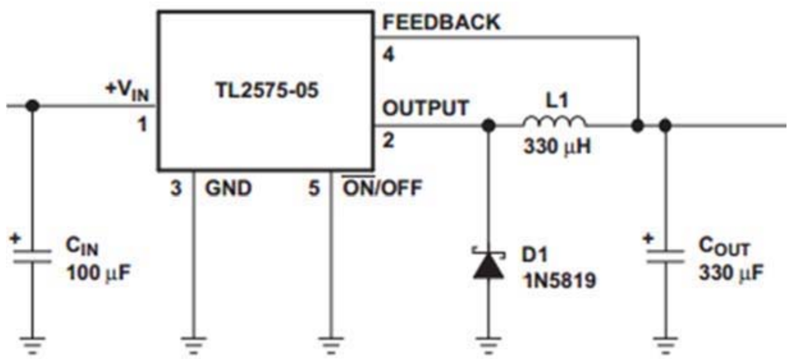
Test Objective:
The objective of this test was to determine whether or not the 5V switching regulators supplied the correct voltage with a minimum amount of ripple.

Test Description/Requirements:
Requirements:

1: One TL2575 5V switching regulator
2: Oscilloscope
3: Power source
4: One 100uF capacitor
5: One 330uF capacitor
6: One 150uH inductor
7: One 1N5819 Schottky diode

Process:

The TL2575 switching regulator will be hooked up into the circuit as shown in **figure (**5V regulator shown, same circuit). A power supply will be used to supply the regulator with ~6-7V of input voltage. An oscilloscope will then be connected across C_{out} and the output voltage and ripple will be observed.



Anticipated Results:

The regulator will regulate the input voltage to 5V with a ripple of 50mV - 150mV.

Requirement for Success:

TBD

Actual Results:

TBD

Reason for Failure:

TBD

Recommended Fix:

TBD

Scheduled Test Reporting Form

Test Item: Hercules SPI
Tester Name: Dewey Williams
Test Date: 1/12/2015
Test Time: 5 PM
Test Location: CoE Senior Design Lab

Test No: 15
Test Type: Integration Test
Test Result: Pass

Test Objective:
The objective of this test is to verify correct operation of the Hercules' SPI interfaces in master mode. Only the SPI1/mibSPI interface will be tested, as the pins necessary for the other interfaces are not available until the additional MCU pin header is populated.

Test Description/Requirements:
Requirements:

TI Hercules TMS570LS04x MCU
Oscilloscope
Known-working SPI slave device

Process:

Use HalCoGen to configure the SPI1 (legacy) interface for testing. Program the Hercules to send some dummy data over SPI, and view the SCK, SIMO, and CS lines with an oscilloscope and verify correct operation. Next, attach a slave device and attempt to communicate. Verify any received data using a debug tool. An oscilloscope may be used to view the SCK, SIMO, SOMI, and CS pins again to verify full duplex operation. Repeat these steps using the mibSPI interface.

Anticipated Results:

The Hercules will be able to send and receive data over SPI using both legacy and mibSPI modes.

Requirement for Success:

The test is a success if the Hercules is able to communicate with a slave device in both legacy and mibSPI modes.

Actual Results:

The Hercules was able to successfully communicate with our ILI9340-driven display using both legacy and mibSPI modes. An oscilloscope was used to examine the behavior of the mibSPI module while transmitting multiple bytes. Correct operation was again verified by connecting the Hercules to a MSP430 configured as an SPI slave and checking the data at both ends using a debug tool.

Reason for Failure:

N/A

Recommended Fix:

N/A

Other Comments:

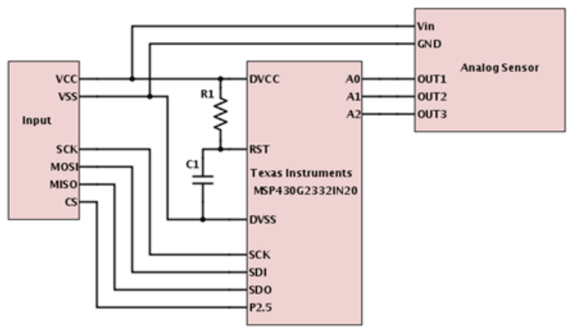
Scheduled Test Reporting Form

Test Item:	MSP430 SPI Slave Mode	
Tester Name:	Christopher Riker	
Test Date:	1/21/15	Test No: 16
Test Time:	5 PM	Test Type: Integration Test
Test Location:	CoE Senior Design Lab	Test Result: Pass

Test Objective:
The objective of this test is to demonstrate the SPI functions of the MSP430's USCI and to generate baseline code to be used in SPI slave sensor applications.

Test Description/Requirements:
Requirements:

MSP430G2553 microcontroller
Flash programmer and debugger (such as an MSP-EXP430G2 Launchpad)
Known-working SPI master device (such as another microcontroller)



Process:
Create a program using example code and other resources which will send SPI data (such as an increasing counter or some constant value) when data is sent by a master device. Connect the SPI and ground pins of each device and view the contents of internal registers and memory using a debugging utility to verify the correct transmission of data. Two similarly programmed MCUs may be connected to different chip selects on the master in order to verify that inactive devices will not interfere with other transmissions on the bus, or that these other transmissions will not affect the inactive slaves. An example pin-level layout can be found below.

Anticipated Results:
Data sent by the slave will be correctly received at the master, and vice versa.

Requirement for Success:
In order for this test to be a success, the slave device must successfully transmit the most current internal data when polled by the master, and it should also successfully receive data from the master, in order to verify full duplex operation. The slave devices must also be completely inactive on the bus unless their chip select (or Slave Talk Enable in the case of the MSP430s) is pulled active in order to prevent interference.

Actual Results:
The MSP430s were programmed to send an incrementing count when polled by the master device (the TI Hercules TMS570LS0432) using UCB0 (pins 1.4-1.7). Each slave was programmed with a different starting count so that the results could be properly identified. After some development and debugging, data was correctly sent and received between the master and selected slave devices, and no interference was detected so this test is considered a success.

Other Comments:
Code samples for this test are available in the full report. This code will be assembled into an easy to use driver library to and included in the software for each sensor MCU.

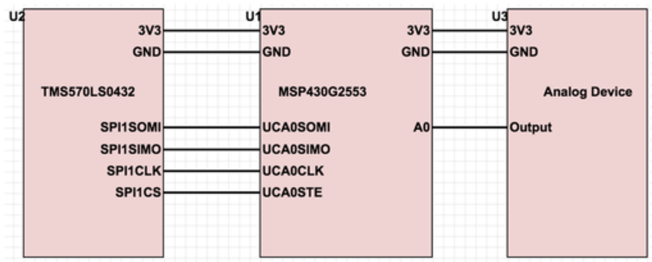
Scheduled Test Reporting Form

Test Item: MSP430G2553 SPI slave mode w/ ADC10 conversion
Tester Name: Dewey Williams & Christopher Riker
Test Date: 1/15/2015
Test Time: 5 PM
Test Location: CoE Senior Design Lab
Test No: 17
Test Type: Integration Test
Test Result: Failure

Test Objective:
The objective of this test is to determine if the MSP430 microcontroller is capable of doing efficient A/D conversion while communicating with a master device over SPI.

Test Description/Requirements:
Requirements:

1: One MSP-EXP430G2 LaunchPad
2: One SPI master device
3: One potentiometer
4: Breadboard
5: Jumper wires



Process:
The TMS570LS04x LaunchPad will be configured in SPI master mode using HalCoGen. The MSP430's ADC10 module will be set up to continually convert on one channel (sequence mode 2). The potentiometer will be connected to VCC and Ground, with the wiper being connected to analog port A0. Code will be written for the TMS570LS04x that continually begins SPI transmissions with the MSP430 and stores the received message in some variable. Both chips will be programmed and the debuggers started. The tester(s) will observe the variable and how it changes when turning the potentiometer knob, noting how quickly the variable updates.

Anticipated Results:
The variable that stores the transmitted data should update in real-time with the turning of the potentiometer knob. Since the hardware USCI module and the ADC module are implemented on-chip separately, no troubles are anticipated by the testers.

Requirement for Success:
The test will be considered successful if the MSP430 is able to efficiently convert analog readings to digital signals, and the master device receives those signals in real-time.

Actual Results:
Initially, the test seemed to be a failure. However, upon checking wire connections, the testers discovered that one of the wires was connected to the wrong slot in the breadboard. When this was corrected, the test did not go as anticipated. The variable took 2-3 transmissions to update when turning the potentiometer wheel.

Reason for Failure:
Upon further consideration, the testers discovered that a line of code in the USCI0RX interrupt vector `__delay_cycles(10000);` may have been the culprit.

Recommended Fix:
The `__delay_cycles()` call is necessary, however 10,000 cycles may be overkill. The number should be reduced drastically, which may lead to a successful future test.

Other Comments:

Scheduled Test Reporting Form

Test Item: MSP430G2553 I^2C (Master mode)
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: CoE Senior Design Lab

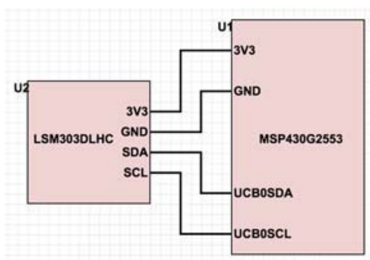
Test No: 18
Test Type: Integration Test
Test Result:

Test Objective:
The objective of this test is to determine whether or not the testers can get the MSP430G2553 microcontroller configured and working in I^2C master mode. This will be used to communicate with the digital accelerometer/magnetometer.

Test Description/Requirements:
Requirements:

1: MSP-EXP430G2 LaunchPad (with an MSP430G2553)
2: LSM303 Accelerometer/Magnetometer
3: Breadboard

Process:
The MSP430G2553 will first be configured as an I2C master device (using Grace or similar). The LSM303 will then be connected to the proper pins of the MSP430 (pin diagram found in **figure**). Code will be written that polls the acceleration values from the LSM303 continually. The variable used to store the accelerometer output will be observed to determine whether or not the MSP430 is receiving communication from the accelerometer.



Anticipated Results:
The MSP430 will be configured correctly such that is able to communicate with the LSM303 and receive accurate readings from it.

Requirement for Success:
The test will be considered a success if the tester is able to get the MSP430 communicating with the LSM303 over I2C.

Actual Results:
TBD

Reason for Failure:
TBD

Recommended Fix:
TBD

Other Comments:

Scheduled Test Reporting Form

<u>Test Item</u>: MSP430 I2C Master Mode w/ SPI Slave Mode
<u>Tester Name</u>: TBD
<u>Test Date</u>: TBD
<u>Test Time</u>: TBD
<u>Test Location</u>: College of Engineering Lab

<u>Test No</u>: 19
<u>Test Type</u>: Integration Test
<u>Test Result</u>:

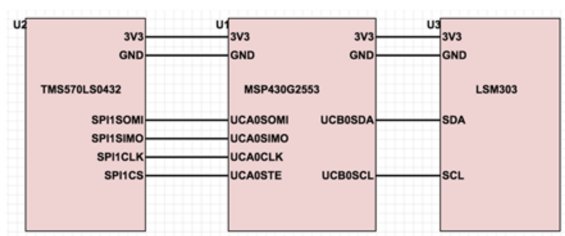
<u>Test Objective</u>:
The objective of this test is to determine whether or not the MSP430G2553 can be used in I^2C master mode and SPI slave mode at the same time. This will be used for the accelerometer module, as the acceleration data must be collected from the digital accelerometer over I^2C and then sent to the Hercules LaunchPad over SPI.

<u>Test Description/Requirements</u>:
Requirements:

1: One TMS570LS04x (Hercules LaunchPad)
2: One MSP-EXP430G2 LaunchPad
3: One LSM303 Accelerometer/Magnetometer
4: Breadboard

Process:

The Hercules LaunchPad will be configured for SPI Master mode (already done). The MSP430 will be configured for SPI Slave mode (already done) using the USCIB0 module. The MSP430 will then be configured for I^2C master mode using the USCIA0 module. The LSM303 accelerometer will then be connected to the MSP430. A pin-level diagram of this connection can be found below.



The MSP430 will be configured to continually poll the accelerometer for acceleration data, with the Hercules requesting SPI transmissions every so often. A variable on the MSP430 that stores the accelerometer data collected over I2C will be watched to ensure that the MSP430 is indeed collecting data from the accelerometer. The data transmitted over SPI will be stored in some variable, and that variable will be watched to determine whether not the MSP430 is successfully transmitting data to the Hercules over SPI.

<u>Anticipated Results</u>:

Since the MSP430 has two separate USCI modules, the design team does not expect any issues stemming from using both of them in tandem.

<u>Requirement for Success</u>:

The test will be considered a success if the MSP430 can collect data from the LSM303 accelerometer over I^2C while sending data to the Hercules over SPI.

<u>Actual Results</u>:

TBA

<u>Reason for Failure</u>:

TBA

<u>Recommended Fix</u>:

TBA

Scheduled Test Reporting Form

Test Item: Battery Life
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: College of Engineering Lab

Test No: 20
Test Type: Integration Test
Test Result:

Test Objective:
The objective of this test is to measure the battery life of the system when it is running in a fully operational state.

Test Description/Requirements:
Requirements:

1: Dummy load (e.g. low value resistor)
2: Timer

Process:

A resistive load will be placed across the batteries until they are drained (to around 5.5 volts), and the time recorded to calculate the total charge which can be used with the power analysis table to calculate the total expected battery life of the DAQ system.

Anticipated Results:

With the chosen batteries, the system should theoretically run for far longer than the required 5-hours (as determined in the project proposal).

Requirement for Success:

The test will be considered successful if the batteries are able to power the whole system for no less than the required 5-hour goal set by the design team.

Actual Results:

TBD

Reason for Failure:

TBD

Recommended Fix:

TBD

Other Comments:

Scheduled Test Reporting Form

Test Item: Plano 147000 Waterproof Case
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: College of Engineering

Test No: 21
Test Type: Integration Test
Test Result:

Test Objective:
The objective of this test is to test whether or not the Plano 147000 waterproof seal will hold up to the environment in which it will be used in.

Test Description/Requirements:
Requirements:

1: Plano 147000 Waterproof Case
2: Water source
3: Paper towels

Process:

The case will be lined with paper towels. This will make it easy to tell whether or not water penetrated the box. The case will then be submerged just under the surface of some water source (location TBD). The box will then be removed, and the paper towels will be examined to determine whether or not water penetrated the box. This test will be sufficient in determining whether or not the box will protect the components inside of it from the elements.

Anticipated Results:

The moisture will be successfully kept outside of the box (as advertised), and the paper towels inside will remain dry.

Requirement for Success:

The test will be considered a success if the box keeps any and all moisture out.

Actual Results:

TBA

Reason for Failure:

TBA

Recommended Fix:

TBA

Other Comments:

Scheduled Test Reporting Form

Test Item: Project Box Waterproof Seal Test
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: TBD

Test No: 2.5
Test Type: Test
Test Result: Pass

Test Objective:
The objective of this test is to determine whether or not the waterproof project boxes obtained for each of the discrete components are actually waterproof.

Test Description/Requirements:
Requirements:

1: Project boxes with seals cut to the correct length
2: Paper towels
3: Water source

Process:
The seals of the small project boxes first need to be cut to the correct length. Once that is done, the boxes will be filled with paper towels, and the tops will be screwed securely on. The boxes, one by one, will be submerged just under the surface of the water source (location TBD). The boxes will then be opened up and the paper towels inside examined to determine whether or not water got into the boxes.

Anticipated Results:
The project boxes will successfully keep water out and the paper towels inside will remain completely dry.

Requirement for Success:
The test will be considered a success if the boxes keep the water out and the paper towels inside remain completely dry.

Actual Results:
TBA

Reason for Failure:
TBA

Recommended Fix:
TBA

Other Comments:

Scheduled Test Reporting Form

<u>Test Item</u>: Sensor Board Testing
<u>Tester Name</u>: TBD
<u>Test Date</u>: TBD
<u>Test Time</u>: TBD
<u>Test Location</u>: TBD

<u>Test No</u>: 22
<u>Test Type</u>: Integration Test
<u>Test Result</u>:

<u>Test Objective</u>:

The objective of this test is to verify the correct working condition of circuit boards designed and milled at the College of Engineering.

<u>Test Description/Requirements</u>:

Requirements:

Milled and populated sensor board
Programmed MSP430G2553 MCU

Process:

Program the MSP430 to toggle each pin in turn (for all pins broken out on the circuit board), and insert into the socket. Apply 3.3v to the designated pin and connect the ground pin to ground. Test each pin on the headers with an oscilloscope or multimeter to verify that the pin is properly connected to the MCU.

<u>Anticipated Results</u>:

The milled circuit board will function as designed, and the MCU will have proper control over each pin.

<u>Requirement for Success</u>:

In order to be a success, the MCU must be able to toggle each header pin on the board. The MCU must also be in a stable state (that is, no strange resets or other odd behavior) to ensure that the chip and board can be safely embedded.

<u>Actual Results</u>:

TBA

<u>Reason for Failure</u>:

TBA

<u>Recommended Fix</u>:

TBA

<u>Other Comments</u>:

Scheduled Test Reporting Form

Test Item: XBee communication between Hercules LaunchPad and Raspberry Pi
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: College of Engineering
Test No: 23
Test Type: Integration Test
Test Result: N/A

Test Objective:
The objective is to test is to determine whether or not the code developed for communication between the XBee modules connected to the Hercules and the Raspberry Pi operates as intended.

Test Description/Requirements:
Requirements:

1: Two XBee-PRO 900HP Module
2: Raspberry Pi running Raspbian
3: TI TMS570LS04x LaunchPad

Process:

Code will be developed for the Hercules LaunchPad and the Raspberry Pi that utilizes UART to send messages back and forth between the two devices via the XBee modules. The Hercules will generate some kind of arbitrary data (e.g. an incrementing variable placed into a string) and send that data to the Raspberry Pi. The Raspberry Pi will then receive that data and echo it to the console. The data will be checked for accuracy and completeness.

The XBee module has already been successfully tested to work with the Raspberry Pi's UART interface (using X-CTU on a laptop to send messages to the Raspberry Pi). The purpose of this test is to ensure that the design team can get the Hercules to do the same.

Anticipated Results:

The team anticipates that the Hercules and Raspberry Pi will be able to effectively communicate (i.e. without erroneous transmissions) with one another with the XBee modules using their respective UART interfaces.

Requirement for Success:

The test will be considered successful if the Hercules and Raspberry Pi are able to communicate with each other via the XBee modules over their respective UART interfaces without erroneous transmissions.

Actual Results:

TBA

Reason for Failure:

TBA

Recommended Fix:

TBA

Other Comments:

Scheduled Test Reporting Form

Test Item: Vehicle-to-Pit Communication
Tester Name: TBD
Test Date: TBD——
Test Time: TBD——
Test Location: College of Engineering

Test No: 24
Test Type: Integration Test
Test Result: N/A

Test Objective:
The objective is to test the functionality of the wireless communication module as part of the data acquisition system when the vehicle is in motion.

Test Description/Requirements:
Requirements:

1- XBee Module
2- Raspberry Pi running Raspbian
3- TI Hercules TMS570LS04x

Process:

The assumption is by the time this test is performed, the XBee is connected to both the Hercules and the Raspberry Pi. The Hercules is installed on the vehicle collecting data from the sensors and the Raspberry Pi is stationed at a simulated pit crew station.

This test is performed by setting sensor variables in the Hercules to constant values. These constant values will be sent to the Raspberry Pi and the data observed.

Anticipated Results:

Thorough testing of the components before vehicle installation gives confidence to successful wireless communication after system integration.

Requirement for Success:

For successful results, the values observed on the Raspberry Pi must be identical to the constant values set to be sent through the Hercules.

Actual Results:

TBA

Reason for Failure:

TBA

Recommended Fix:

TBA

Other Comments:

Scheduled Test Reporting Form

<u>Test Item</u>: Hall Effect Sensor Accuracy Test
<u>Tester Name</u>: TBD
<u>Test Date</u>: TBD
<u>Test Time</u>: TBD
<u>Test Location</u>: College of Engineering Lab

<u>Test No</u>: 25
<u>Test Type</u>: Integration Test
<u>Test Result</u>: N/A

<u>Test Objective</u>:
The objective of this test is to observe the performance of the hall effect module after vehicle installation and verify its accuracy.

<u>Test Description/Requirements</u>:
Requirements:

1: DRV5053 analog bipolar hall effect sensor
2: Four neodymium magnets
3: TI Hercules TMS570LS04x MCU

Process:

The assumption is that all the Hall effect sensor has been installed on the vehicle and the speed data gathered from the sensor is being transmitted to the Hercules and displayed.

The test is performed by driving a second vehicle with a speedometer alongside the Baja car. The speed values on the DAQ display and the second vehicle speedometer will be noted by their respective drivers while the vehicles are moving at a constant speed. Once the vehicle is stopped, the constant value is recorded. The test will be repeated several times for the vehicle moving at different speeds.

<u>Anticipated Results</u>:

Comparing the two values collected from the speedometer and speed sensor module will yield comparable results with minimal percentage error.

<u>Requirement for Success</u>:

To be successful the value of the speed being output from the speed sensor must match the value of the speed output from the speedometer within 5%. This allows room for sensitivity constraints of the sensor while still displaying a speed that is accurate and useful to the driver.

<u>Actual Results</u>:

TBA

<u>Reason for Failure</u>:

TBA

<u>Recommended Fix</u>:

TBA

<u>Other Comments</u>:

Scheduled Test Reporting Form

Test Item: Hall Effect Sensor in Project Box
Tester Name: TBD
Test Date: TBD
Test Time: TBD
Test Location: College of Engineering

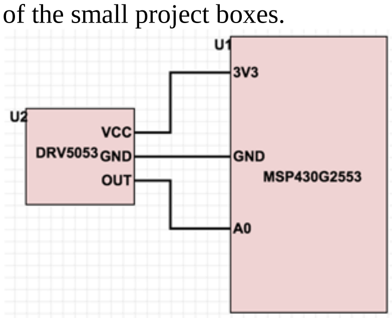
Test No: 2.5
Test Type: Test
Test Result: TBD

Test Objective:
The objective of this test is to determine whether or not the Hall effect sensor will be able to detect pulses from a passing magnet when it is inside the waterproof project box.

Test Description/Requirements:
Requirements:

1: Waterproof project box
2: One DRV50530 Bipolar Analog Hall Effect Sensor
3: One neodymium magnet
4: One MSP-EXP430G2 LaunchPad

Process:
The MSP430 ADC10 will be configured for use, and the Hall effect sensor will be connected to the LaunchPad appropriately, as done in previous tests (see **Figure 1** below). The sensor will be placed on a breadboard behind the lid of the project box. A neodymium magnet will then be passed in front of the lid past the sensor while the MSP430 ACD10 is continually reading and converting the voltage output of the Hall effect sensor. The digital output from the ADC10 will be observed to determine whether or not the Hall effect sensor will be able to detect a passing magnet through the lid or walls of the small project boxes.



Anticipated Results:
Since the neodymium magnets are very strong, the testers expect that the Hall effect sensor will be able to detect the change in the magnetic field as the magnet passes in front of the lid of the project box.

Requirement for Success:
The test will be considered a success if the Hall effect sensor can detect the pulses from the passing magnet effectively enough to be used for speed measurement (e.g. in high-frequency applications).

Actual Results:
TBD

Reason for Failure:
TBD

Recommended Fix:
TBD

Other Comments: