ONBOARD WEATHER STATION

A Design Project Report

Presented to the Engineering Division of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering (Electrical)

by

Allison P. Smyth

Project Advisor: Bruce R. Land

Degree Date: May 2010

Abstract

Master of Electrical Engineering Program

Cornell University

Design Project Report

Project Title: Onboard Weather Station

Author: Allison P. Smyth

Abstract:

Every participant in motorsports strives to achieve one goal: optimal vehicle performance to overcome competition. Although this is clear, what may be surprising is the significant impact that changing weather has on the performance of race vehicles. Every race team (sportsman or professional) uses a weather station to predict performance and setup their vehicle. Until recently it was assumed that a static weather reading at the beginning of the race was sufficient to accurately predict performance. However, in 2009, the Smyth family race team found this assumption to be false. For this reason, the Onboard Weather Station, a weather station that is mounted in a race vehicle, was designed, built, and tested. The Onboard Weather Station collects temperature, barometric pressure, and humidity experienced by a vehicle during the course of a race. Data is stored on a secure digital (SD) card, and can also be transferred to a PC wirelessly while the car is in motion. This is advantageous to race teams with multiple vehicles, allowing them to have immediate access to weather data and make changes to their vehicle(s) waiting to compete. The device was designed as a series of modules to make it adaptable to many types of vehicles and racers. These modules are interfaced using a controller area network (CAN) bus. At the current time, the Onboard Weather Station is completely functional and ready for on-car testing.

Report Approved by

Project Advisor:____ Date: ___________

Executive Summary

In this project, a weather station that can be mounted on a race car to monitor air conditions as the vehicle moves down the track (Onboard Weather Station) was designed, built, and tested. Specifications for the weather station were provided by Portatree Timing Systems, Inc., the project sponsor. All design requirements specified prior to March 2010 were met, resulting in the Onboard Weather Station project being deemed a success. Although all goals for the project were completed, the project sponsor plans minor improvements to the product in order to put it into production by late August.

The Onboard Weather Station is composed of two types of modules: main and external. Each system has 1 main module, which acts as the brain of the system. It is designed to organize, store, and transfer data to a computer upon request by the user. Up to 4 external modules, responsible for collecting weather data, can be connected to the main module. The focus of this project was the design and construction of the hardware and firmware of the main and external modules. Despite this, software was also developed to verify the hardware and firmware design.

The main module hardware interfaces an AT90CAN64 microcontroller from Atmel with a secure digital (SD) card, Flash memory, USB transceiver, accelerometer, XBee-PRO wireless transceiver, and a controller area network (CAN) bus. The external modules connect an ATTINY88 microcontroller with a capacitive humidity sensor, 4 analog pressure sensors, and an analog temperature sensor. A separate CAN controller chip (MCP2510) is also interfaced to the microcontroller for communication with the main module. All embedded code for the modules was written in Assembly in order to meet the strict timing requirements of the system.

The system begins operation when the vehicle starts. The main module waits for a manual or automatic trigger to begin data collection. Once triggered, the device begins polling the external modules at a specified sample rate. Data is stored on the SD card or in Flash if the SD card is not present. It can also be wirelessly transferred to a PC running the software. At the end of the race, collected data can be downloaded to the weather software wirelessly or through a USB port. The weather software graphs the data for easy viewing by users. Furthermore, it provides setup and test screens for the device allowing users to easily change system parameters.

The results from the weather station will be used to improve performance prediction for race cars. Prior to this project, it was believed that air entering a car engine during a race had static temperature, humidity, and pressure; therefore, a weather reading made immediately prior to a run would be sufficient to predict engine performance. Preliminary results from the Onboard Weather Station have proved this assumption to be false. The data shows that there is a large gap between air seen by a race vehicle and weather conditions prior to a run. Future data from the Onboard Weather Station provides an opportunity to increase the accuracy of performance predictions (the desire of every drag racer), making this product more valuable than current weather stations on the market.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

In every type of racing, weather is a critical factor that determines the performance of race vehicles. Professional race teams rely on weather data to make adjustments to their vehicles and obtain the maximum performance possible in upcoming runs. Sportsmen (non-professional) racers use weather data to predict the performance of their vehicle in order to be competitive during a race. In drag racing, cars are particularly sensitive to weather as they are designed to run for short periods of time at maximum power. Consequently, all drag racers (sportsman and professional) use weather stations to predict or tune their vehicle. Over the past few years, many weather stations with prediction algorithms have become available to judge vehicle performance based on weather data; however, at times, these weather stations produce drastically different results than what actually occurs during a run.

Many suggestions have been made for theoretical and practical differences in performance prediction; however, the most reasonable cause for the disparity is that cars are experiencing different weather conditions during a run than racers are using to predict performance. This reasoning appears valid as traditional weather stations, which are mounted on race trailers, are unable to monitor the air changes over the actual race surface. Since asphalt is known to store and radiate heat, the layer of air directly over the track will have different conditions than the air seen by a traditional weather station. Even after sun set, a race surface continues to radiate heat; thereby, expanding the gap between the conditions on the track and the air at the weather station. If cars are actually taking in air from the surface of the track, the difference between predicted and actual performance may be explained.

For this reason, a data acquisition system that collects weather data while a vehicle is making a run would be extremely beneficial to all types of racers. Such a device would allow racers to determine the condition of the actual air entering the intake of the engine, which would allow them to make more accurate performance predictions or adjustments to their race vehicle. Although many data acquisition systems and weather stations exist on the market, there are currently no devices that combine the functionality of these systems. Hence, the Onboard Weather Station was conceived.

The Onboard Weather Station is a weather data acquisition system that can be mounted on race cars and collect weather data as vehicles move down the track. It was designed for Portatree Timing Systems Inc., who plans to begin manufacturing this product by late August 2010. The Onboard Weather Station behaves similarly to a data acquisition system in that it is composed of a main control unit and external data collection modules. The main control unit is placed in the driver's compartment for user interaction and data storage while up to 4 external modules are mounted throughout the car to collect weather data. The communication protocol between the modules allows the main module to handle information from different types of external modules. This provides Portatree with the opportunity to interface additional data collection units in the future. The basis for communication among the modules is a controller area network (CAN) bus.

Upon trigger, the main module of the weather station enters race mode where it collects and records data at a specified sample rate. Data is stored to a secure digital (SD) card, which provides the system with the capacity to store multiple runs of data. The capacity of an SD card also gives the system the ability to be used in motorsport events that require long periods of data storage. Backup data storage is provided through Flash when the SD card is not present. During data collection, the Onboard Weather Station can also send data to a PC wirelessly. This feature allows multi-car teams to access data immediately after teammates have run, permitting them to make improvements to their second car as quickly as possible.

After a race, data can be downloaded to custom weather software that was designed specifically to interact with the Onboard Weather Station. The software is able to extract, organize, and graph data from the main module of the weather station using the USB connection on the main module board. If wireless communication is initialized, the data can also be transferred wirelessly to the weather software. Data stored on an SD card can be read directly by the software using a computer SD card reader. The weather software is also used to update weather station parameters such as sampling rate, record trigger type, number of external modules, and wireless data transmission controls. Calibration of all weather sensors are performed with the weather software.

Weather data collected by the Onboard Weather Station includes temperature, humidity, barometric pressure, and wind pressure, which are used to calculate water grains, vapor pressure, and corrected altitude. Every weather value affects engine performance in a different way, which is why the final performance prediction is based on the calculated values. The weather software performs all calculations in the system, and in the future, will use the calculations to predict engine performance.

As a result of this project, both the main and external modules of the Onboard Weather Station have been designed, built, and tested. Both modules were found to meet all design specifications set forth by the project sponsor. The weather software was also started in order to test the main and external modules. Although complete software operation was not the focus of this project, it is achievable in the near future due to the basic software created for this project. Preliminary results collected from a weather station prototype confirm that weather conditions deviate from those observed by traditional weather stations. In the future, work must be done to interpret how these deviations affect engine performance in order to achieve better accuracy in performance predictions.

2 Literature Review

Prior to designing the Onboard Weather Station, research was conducted in order to determine if an identical product existed. Although no device was found that performed the exact operation of the Onboard Weather Station, products in related markets were identified. All products deemed similar were examined in order to identify necessary features for a successful weather station or data acquisition system. The important features found were presented to the lead design team of Portatree Timing Systems who used this research to develop reasonable specifications and design requirements for the Onboard Weather Station.

The initial research conducted involved a search of the United States Patent and Trademark Office database in order to determine if any restrictions exist in the design of weather stations or data acquisition systems. Only one patent was found relating to race car weather collection; number 5509295, *Weather Station Device*, which was issued to Fred J. Bartoli, an engineer of Altronics Corporation. This device is a handheld weather station that collects weather data in order to calculate corrected altitude (or density altitude as it is referred in the patent). It can be seen in Figure 1. The Altronics patent does not affect the design of the Onboard Weather Station as the claim is for a standalone module that performs all weather collection, calculation, and analysis without the need of an external computer. Furthermore, the device is classified as a weather station rather than a data acquisition system, further distinguishing it from the Onboard Weather Station.

Figure 1: Altronics Weather Station

A patent search for race car data acquisition systems or general data acquisition systems did not return relevant hits. From this research, it was determined that by designing the Onboard Weather Station as a data acquisition system (as was desired by the project sponsor), patent infringement on the Altronics system would not occur.

After concluding that the Onboard Weather Station would not violate patents, research was conducted in order to determine the general features of weather stations on the market. The three most popular weather stations were identified and examined in order to determine the most significant features of each. The weather stations observed included PerformAire, Snap-In Weather, and RaceAir Pro sold by Altronics, Portatree, and Computech, respectively. PerformAire can be seen in Figure 1 while Snap-In Weather and RaceAir Pro can be seen in Figure 2.

a.) RaceAir Pro by Computech **b.) Snap-In Weather by Portatree**

Figure 2: a.) RaceAir Pro and b.) Snap-In Weather

The Altronics weather station is designed specifically for drag racers. It is able to store up to 400 runs of data including temperature, barometric pressure, relative humidity, wind speed, wind direction, and oxygen level in order to calculate the air density ratio, vapor pressure, dew point, and density altitude. By adding the PC software, additional runs can be stored in a weather log database and graphing features are also enabled. All Altronics weather stations provide elapsed time (performance) predictions for racers, which they determine based on density altitude calculations from the collected data. Altronics systems have accuracy of 1° Fahrenheit, 0.1 inHg, and better than 10% for temperature, barometric pressure, and relative humidity, respectively. They fully compensate for component variations based on temperature.

Computech weather stations are designed for all types of motorsports and could be considered an all around racing tool rather than simply a weather station. Like the Altronics weather station, the Computech system is able to collect temperature, humidity, and barometric pressure to an accuracy of 1.2°, 3%, and 0.05inHg, respectively. From these readings, the weather station calculates vapor pressure, air density ratios, and density altitude. When the Computech PC software is added to the system, graphing and logging features are also enabled. Like the Altronics system, the Computech system provides elapsed time prediction for drag racers; however, Computech gives users the option of predicting based on density altitude or a horse power correction factor.

The Portatree Snap-In weather station is currently manufactured and sold by the project sponsor. It is designed to collect temperature, humidity, and barometric pressure, which it uses to calculate corrected altitude (density altitude). The system has an accuracy of 0.1°F, 0.01inHg, and 1% for temperature, barometric pressure, and humidity, respectively. Unlike other weather stations, the Portatree system allows the user to select a sample rate of 3, 5, 6, or 10 minutes. It further allows users to correlate run times to weather data collected for up to 150 runs. As with the other weather stations, Snap-In weather performs elapsed time predictions based on corrected altitude. From the connection with Portatree, it was learned that horse power correction will soon be featured in all their weather products, since tests have shown that greater prediction accuracy can be achieved by compensating for this factor. As with the other weather stations, when PC software is added, additional runs can be stored, graphs provided, and wind data analyzed.

From the weather station research, it was determined that the most important factors for performance prediction are temperature, humidity, and barometric pressure, which are used to calculate corrected altitude. To achieve the necessary precision for a corrected altitude prediction, accuracies of 0.1°F, 0.1inHg, and 10% RH for temperature, barometric pressure, and relative humidity, respectively, are required; however, greater accuracy produces better results. From the Altronics system, it was suggested that temperature compensation is also necessary for a truly accurate system. All weather stations incorporate graphing of weather data, while 2 provide horse power correction and air density ratios. Wind speed and direction are also important in all weather stations reviewed. These features were all noted as important for the Onboard Weather Station system for it to be competitive with current weather stations.

Data acquisition systems were also examined in order to determine common sampling times, triggers, and durations as well as user interaction features. Currently, the most popular data acquisition system on the market is manufactured by Racepak. Another well-designed system is that which was developed by Corsa. The two systems can be seen in Figure 3.

a.) Racepak

b.) Corsa

Figure 3: a.) Racepak and b.) Corsa Data Acquisition Systems

The Racepak data acquisition system records engine rotations per minute (RPM), driveshaft RPM, battery voltage, and g-force with a 2-axis accelerometer. External sensor packs can be added to monitor exhaust gas temperatures (thermocouples), cylinder head temperatures, air-fuel ratios, and oil pressure. The device is triggered off a 12-volt event after which it stores data to an SD card at 100 samples per second (10ms sampling period). The system mounted in the race vehicle has no user interface; therefore, the racer is left to assume that the device is operating. Racepak's Data Link software is used to view the data, which is read directly off the SD card after a run.

The Corsa data acquisition system has an internal real time clock (RTC) as well as RPM, accelerometer, thermocouple, and air/fuel ratio modules which are interfaced to the main system with a controller area network (CAN) bus. Data recording is triggered from a mechanical switch or a channel level after which it records 26 minutes of data to Compact Flash. A special Flash reader is necessary to transfer the data from the Corsa system to a PC. Alternatively, a wireless connection can be established for data transfer. The Corsa system is able to sample at 1 to 50 samples/second giving it a maximum sample rate of 20ms. The system runs off of the car ignition system and uses LEDs for driver status indicators. As with the Racepak system, data is solely viewable through provided software.

From research into data acquisition systems, it was determined that the standard way to view collected data was through downloading it to a PC for viewing. Few user interface features are provided on the module located in the vehicle, and when present, they are in the form of LEDs. Both systems observed could be triggered through an external event, and the standard sample period was 10 to 20ms. Both devices provide large amounts of data storage; however, the Racepak SD card approach appeared to provide easier data access to the user. The wireless download provided by Corsa improved access to the data collected by their system. The Corsa system appeared more adaptable to different types of racers and race vehicles than the Racepak system. This was concluded, since more features are selectable and the number of sensor packs that can be added to the Corsa system is nearly limitless due to the CAN bus.

Toward the end of research into relevant systems to the design of the Onboard Weather Station, it became clear that both weather station and data acquisition system manufacturers recognize the correlation between run information and weather data. All weather software reviewed allowed users to associate run data to a weather sample, while the data logging software observed provided a location for a weather entry related to the logged run. It appears that the main reason that these companies have not taken the next step into adding data logging or weather collection to their systems is the belief that weather conditions are unchanging during the short time period of a race. This belief provides great opportunity for the Onboard Weather Station, which is designed to prove that this assumption is false.

The final system researched for the development of the Onboard Weather Station is the fuel injection system on current passenger vehicles. This system is designed to adjust the air-to-fuel ratio of a vehicle based on the mass of air entering the in-take of the vehicle, the load on the engine, and the engine temperature. It was interesting to learn that fuel-injection systems do not take into account temperature, humidity, and barometric pressure when determining air-to-fuel ratios. Therefore, information from the Onboard Weather Station may be able to improve the performance of these systems in the future by providing the engine control unit with more information regarding the air that is entering the motor. In this way, it could possibly improve the engine efficiency of passenger vehicles.

The final references reviewed for the design of the Onboard Weather Station were the rulebooks for the National Hot Rod Association (NHRA) and International Hot Rod Association (IHRA), the two largest sanctioning bodies for drag racing which is the intended market for the initial Onboard Weather Station product. Both sanctioning bodies had similar rules regarding the use of data recorders on race cars. Neither sanctioning body had restrictions on weather stations. The relevant rules are designed to ensure that cheating will not occur during a race. Data recorders mounted on race cars are not allowed to display collected data to the driver during the race, or connect to the timing system of the race track. Systems may only display data after a pass or run is complete, and devices that assist in determining track location are prohibited. The transmission of data or information to any remote location during the run is also prohibited; however, this restriction is contestable if it can be proven that the data cannot be used to determine the outcome of the run. Data recorders must not activate any function on the vehicle, and data collection devices cannot be activated by the Christmas Tree (system of lights used to start a race). All systems must be approved by the sanctioning bodies prior to use during a race.

From research into related industries and products, it was determined that a device that performs the exact operation of the Onboard Weather Station does not exist. Furthermore, it was determined that the design for the Onboard Weather Station would not result in patent infringement from the one relevant patent related to weather data collection. Desirable features for the Onboard Weather Station were identified through examining the most popular weather stations and data acquisition systems currently on the market. From this research, the data the system should collect, frequency at which the system should collect it, and information the system should provide to the user were identified. Finally, it was determined that a weather station data recorder would be a legal installment on any race vehicle.

3 Design Requirements

The design requirements for the Onboard Weather Station were specified by the project sponsor at the inception of the project. Much of the design criteria resulted from the prior art research described in Section 2. Due to the novelty of such a device, some requirements or specifications were modified over the course of the project. This section will be broken into 3 subsections in order to describe the design criteria for the project. The first section will address the general design requirements while the second will address the technical design requirements. The final section will summarize the design requirements for the project.

3.1 General Requirements

The purpose of this project is to design a weather station that can be mounted on race cars to collect weather data as the car moves down the track. Since the intent of the sponsor is to manufacture and sell the device, all aspects of design are required to conform to manufacturability as well as customer satisfaction.

The most important requirement for the system is to be able to collect and store weather data from the air-intake of a race car. The system must have the ability to record a full day of race data without requiring a user download. The device should be able to connect to a PC for data extraction, viewing, and system parameter updates. The PC software must be able to calculate values related to vehicle performance based on the data collected from the weather station. From these values, performance predictions should be computed for the user.

After viewing the Corsa system, it was determined that a wireless connection to a computer should also be included. Such an option would be beneficial to racers who choose to mount their system in vehicle locations that are difficult to access. The project sponsor augmented this requirement to include that the system should be able to wirelessly transfer data to the software as the race vehicle is moving down the track. They intend this feature to benefit multi-car teams.

For reasons of customer satisfaction, the device must be adaptable to many types of users and race cars. In order to do this a modular design was emphasized by the project sponsor. The intent of modularity is to allow the same weather station to service low budget sportsman racers and high budget professional teams. A modular design will also improve manufacturability, as it is easiest to manufacture a system a single way rather than making slight modifications to every device that undergoes the assembly process. In a modular system, multiple devices are initially designed. Each device is then manufactured without alternation from other devices of the same type. During shipping, the modules desired by the user can be selected from a shelf, interfaced with a simple connection, and sent to the user with little hassle to the manufacturer. Therefore, modular design is the key to manufacturability, and an important requirement for the project sponsor.

The sponsor specified that wireless should not be used to transfer data between interacting modules in the weather station. A race car is a high noise environment, which may inhibit error-free data communication especially for modules placed in the engine compartment of the vehicle. Since errors in transferring data between modules could result in inaccurate data being recorded, the sponsor decided that wireless transfer between modules should not be considered. Although error-free data transfer is also important for wireless communication as the vehicle moves down the track, this is an optional feature that can be added on request. Therefore, it does not affect the base design of the weather station, and if found to be inaccurate, would not require a complete system redesign. For this reason, wireless communication to a PC was a requirement, while wireless communication between modules was prohibited.

The device should be packaged in a durable case to prevent damage to the electrical circuitry in case of vehicle impact. Furthermore, it should be resistant to water, oil, and various cleaners that may be present in a racing environment. In general, any casing should protect the delicate sensor circuitry against the vehicle environment without impacting the device's ability to monitor air conditions.

All modules developed should have a user interface. Although some data acquisition systems reviewed provided no interaction with the driver while in the race vehicle, the project sponsor preferred a design that would inform the user when the device was operating properly or when error conditions occurred.

The final high-level specifications for the system include that the device should be easy to use and should not be detrimental to vehicle performance. If the device interferes with vehicle operation, a racer will not consider it. If the device is complicated to use, racers will often choose to avoid it, even if it may help predict performance. Most racers are mechanically-minded individuals, and tend not to understand the details of electrical systems. Therefore, for them to adapt a new electrical based data collection system, the device including the software would have to be easy to understand and must not require large amounts of user input for functionality. Since the device must also be adaptable to different types of racers, a balance must be created between ease of use and changeable features.

3.2 Technical Requirements

In order to match the abilities of other weather stations on the market, the system was required to sample temperature, humidity, and barometric pressure. Furthermore, to have enough accuracy to properly predict performance, measurement precision greater than 0.1°F, 0.1inHg, and 10% RH was necessary from the sensors selected. In order to match the abilities of more advanced weather stations, the system was also required to include sensors for monitoring wind pressure.

Since the system was also designed as a data acquisition system, it was decided that common measurements from traditional data acquisition systems should be included. The sponsor selected that 4 rotation per minute (RPM) inputs as well as a 3-axis g-force sensor capable of at least ±7g readings (force exerted by professional drag race vehicles on launch) should be included in the initial system. The voltage of the ignition system should also be monitored by the Onboard Weather Station. It was expected that this data would allow the user to more easily correlate the weather data to specific points of the race. The sponsor also expressed a desire for the device to be able to interface with other sensors in the future to make it a more complete data acquisition system.

Originally, the system was required to have a time stamp associated with each sample stored in the weather station. This time stamp would be provided by a real time clock that would operate with a precision of 1 second. This requirement was modified during the design process as it was realized that due to the regularity of sampling periods, only one timestamp is required at the beginning of the run for the time of all other data samples to be known. In the future, it may be decided to increase the accuracy of the real time clock to 0.01 seconds (as the sampling period is on this level of granularity); however, the project sponsor decided that for the current time 1 second granularity would be suitable.

The requirement specifying that the system should record a full day of race data was interpreted as 8 runs of data. This figure was reached by adding 2 runs to the maximum number achievable in a single race day at a competition drag race event with 128 cars in each of 5 classes. This number also appeared sufficient as most individuals performing tests on their race car do not make more than 4 to 6 runs to avoid adding wear to the engine. One run of data was determined to require at most 1 minute and 30 seconds of data storage time. This was conservatively determined based on assuming the driver would want to capture data from the burnout (tire warming procedure) as well as shutdown. Therefore, the device must be able to store 12 minutes worth of data between downloads.

The connection with the PC to download or update data must be through USB, since at race tracks most individuals carry laptops rather than desktop computers. USB ports are more common on laptops than RS-232 serial communication ports; therefore, USB is the most suitable choice for PC communication. Since the Onboard Weather Station is designated as a peripheral device to the PC, it must contain a Type-B USB connector and be able to interface to a PC with a standard Type-A to Type B USB cable. High-speed communication is not necessary with the PC. The only requirement is that all data is transferred without error.

The wireless connection between the PC and the computer must have a range of at least ½ mile. This specification was determined based on the fact that drag strips have a ¼ mile length with a ¼ mile shut down. It is desirable that a user at the starting line of the drag strip who is waiting to run be able to collect data from a teammate at the end of the track. The wireless device must have sufficient speed to transfer a data packet within a sampling period. If the wireless alternative selected is able to meet all these criteria it will also be able to send data to a PC after a race is complete as a method of data download.

For the device to be accepted by the two largest drag racing sanctioning bodies, it must be proven that the data wirelessly transmitted to a PC cannot be used until after a race has ended. In order to do this, the weather software must be setup to block user interaction until the race-time wireless transmission is complete. Furthermore, it must be possible to ensure that no other individual can receive the wireless signal transmitted by the device. In order to do this, addressing will be used, which will make it nearly impossible for signal interception to occur. Before manufacture, this feature will have to undergo final approval from the two sanctioning bodies.

In order for the device to be modular, the design should be broken into components. This requirement was made after the start of the project, as the sponsor moved toward making the device closer to a data acquisition system than a weather station. In data acquisition systems, there is a main module for organizing, storing, and transferring data as well as sensor modules (or sensor packs as they are denoted by Corsa) for collecting data. Therefore, the sponsor required the Onboard Weather Station to be setup in this way. In order to interface multiple sensor modules to the main module without wireless communication (a specification from 3.1), a communication bus will be needed. The project sponsor specified that up to 4 external modules should be able to interface on the bus without impacting system operation.

The design for manufacturability resulted in further implications for overall system design. All components chosen were required to be standard parts rather than custom made devices. A manufacturer check was required for every component selected in order to determine the likelihood of the part becoming obsolete in the near future. Furthermore, research was required to determine if direct replacements existed for all chosen components. If a part is unique, it is a bad choice for a system intended for production. If the component were to become obsolete, without an alternate part the entire system would require redesign. This is costly in a production environment.

The device was required to run off of batteries as well as a 10 to 30 volt signal from the vehicle ignition system. This range was specified as some vehicles use 12 volt while others use 24 volt ignition. The remaining buffer was specified by the project sponsor to protect the weather station from electrical surge on startup. Although current was initially considered, it was eventually determined that current draw of the weather station would not be a significant aspect of the design. The cooling fan of the engine draws 20 amps during operation, which will dominate any current the Onboard Weather Station will draw from the system.

To further support different types of racers and race cars, the device was setup to have several adaptable features (similar to the Corsa data acquisition system). The sponsor specified 3 sample periods for the device: 20, 50, and 100ms. They also specified 3 trigger sources for automatic triggering based on race conditions: transbrake release (transmission brake), two-step press (engine revolution limiter), or RPM level exceeding a user-defined threshold. Furthermore, it should have a positive action on/off switch for manual enabling of record.

In order to prevent the device from obstructing user operation, the board size for the weather station was specified to be 4”x6” by the sponsor. The external sensor modules for monitoring air in-take were limited to 2 ½” x 4.” If possible this size will be reduced in the future to further ensure that no obstruction to air intake occurs. The sponsor specified that the main device should be encased in an extruded aluminum enclosure. Such an enclosure would protect the device from race environment conditions that are destructive to electrical circuit operation.

Another technical requirement for the system was that LEDs be used for user interaction on all modules designed for the system. LEDs were chosen due to NHRA and IHRA rules, which specify that the device cannot provide data to the user during a race. LEDs would lead to the least questions in this respect. Furthermore, LEDs allow the driver to quickly view the device to know if it is operating properly. In a racing environment, where decisions must be made quickly, a simple interface of this type is most appropriate. Finally, LEDs are small, which would allow them to be added to the modules designed without significantly increasing the board space required.

The final requirement for the system was that all embedded code be written in Assembly and all software in Delphi. These specifications were made for several reasons. Assembly was required as the data collection modules must conform to very strict timing requirements. In order to be absolutely positive that all timing requirements are met, Assembly is necessary. Another reason that Assembly and Delphi were required is that the project sponsor maintains all their software and embedded code in these two languages. Therefore, they have the most onsite support for Assembly and Delphi and prefer to keep their product lines in consistent languages.

3.3 Summary

From the detailed analysis in the last 2 sections, it was determined that the following are the main design requirements for the Onboard Weather Station.

- The system shall collect temperature, humidity, and barometric pressure with accuracy of 0.1°F, 0.1inHg, and 10% RH, respectively
- The system shall collect 3-axis g-force data up to ±7g
- The system shall record data from 4 RPM inputs
- The system shall record the voltage level of the race car ignition system
- The system shall be able to monitor other data in the future
- The system shall include a real time clock accurate to 1 second
- The system shall record 12 minutes of data between downloads
- The system shall connect to a PC through a standard Type-A to Type-B USB cable
- The system shall wirelessly transfer data up to ½ mile
- The system shall be able to wirelessly transfer 1 packet of data in a sample period
- The system shall use wireless addressing

- The system shall have a single data storage module interfaced to up to 4 external data collection modules
- The main and external modules shall not communicate wirelessly
- All components selected for the device shall be standard production line parts with direct, drop-in replacements available
- The system shall be powered from batteries or a 10-30 volt vehicle ignition system
- The system shall support sampling periods of 20, 50, and 100ms
- Data recording shall be triggered through a mechanical switch or automatic race condition
- The race conditions to trigger automatic data recording shall include transbrake release, two-step press, or RPM exceeding a user specified threshold
- Automatic race conditions shall endure a user specified period for data recording to begin
- The main board size shall be 4”x6”
- The external sensor board size shall be less than 2 ½”x4”
- The modules shall be enclosed in extruded aluminum cases
- The weather station and software shall require few inputs from the user for off the shelf use
- All modules shall use LEDs for user interaction
- The main and external modules shall be programmed in Assembly

Although the software was not the focus of this project, design requirements were created in order to have guidelines in which to form the test version. These are listed below.

- The software shall graph data collected by the user
- The software shall calculate corrected altitude, horse power correction factor, water grains, and vapor pressure
- The software shall make performance predictions based on calculated values
- The software shall allow users to update system parameters
- The software shall test data collection modules
- The software shall calibrate data collection modules
- The software shall extract data from the weather station
- The software shall store extracted data to a file
- The software shall erase weather station memory
- The software shall block data received wirelessly until transmission has ended
- The software shall be written in Delphi

4 Tradeoff Analysis

From the design requirements recognized in Section 3, it was decided that the work for the project should be conducted in two stages: main module design and external module design. This section will be divided accordingly in order to discuss the range of solutions considered for the subsystems of each of the modules. It should be noted that a large amount of tradeoff analysis was not required in this project due to the strict set of requirements provided by the project sponsor. The analysis that did occur will be included in detail in the following sections.

4.1 Main Module Tradeoffs

From the system requirements identified in Section 3, a general block diagram was created for the main module of the Onboard Weather Station. This can be seen in Figure 4. The tradeoffs considered for this module will be described based on the module subsystem to which they apply. Only subsystems in which actual design alternatives were considered will be included in this section.

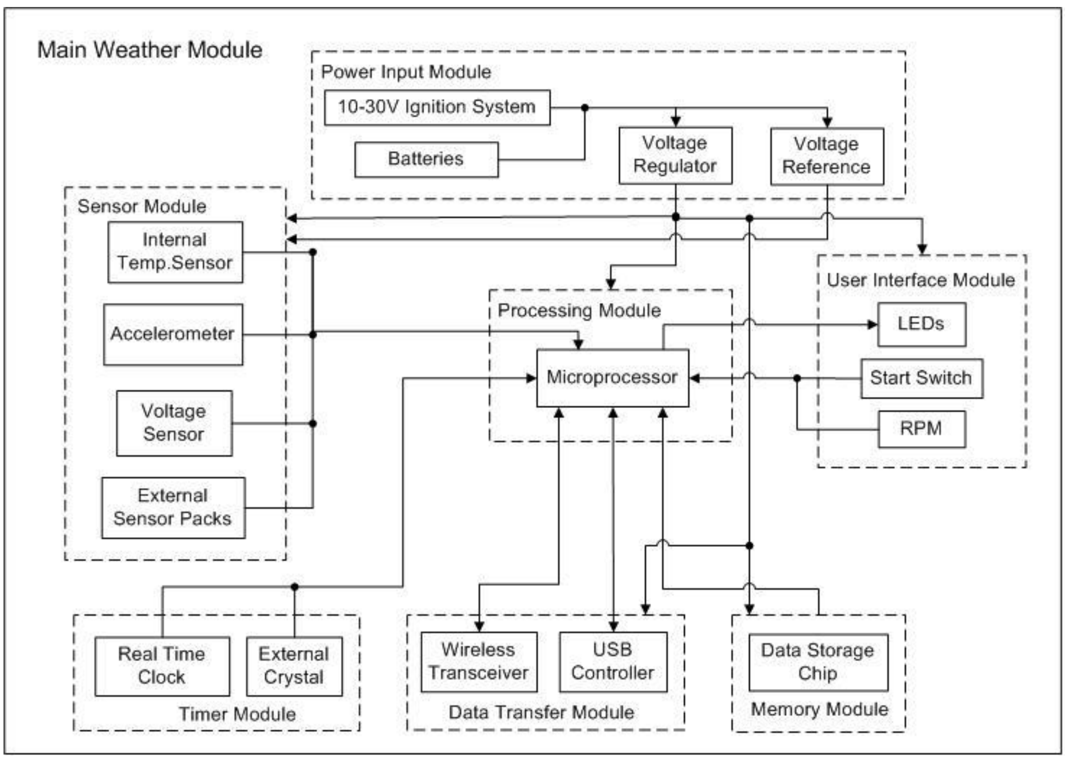


Figure 4: General Block Diagram of Main Module

4.1.1 Power Input Module

The power input module is designed to take in a 10 to 30 volt input from the vehicle ignition system as well as voltage from a battery. It must protect the system from electrical surges that are inherent upon vehicle startup. Since sensors will be present in the system, the power input module is required to provide a precision voltage signal to the sensor module of the circuit as well.

The first aspect of the power input module is the transient suppression circuitry, which ensures that the device is not damaged on system startup. The first device considered for transient suppression was a metal oxide varistor (MOV). The placement and behavior of this device in a circuit is similar to a clamping diode. At low applied voltages, the MOV looks like an open circuit, but when applied voltages exceed the clamping voltage the device becomes a short circuit and protects the components that it shunts. MOVs have nanosecond switching speeds, small size, and can handle current surges up to 100s of amperes. Although, MOVs appeared to be very promising, research revealed that MOVs are prone to catastrophic failure. After a certain number of current surges, MOVs have been seen to catch on fire or explode. Although the probability of this event occurring is small, the associated consequences would result in the ban of the Onboard Weather Station from the racing market. For this reason, the risk was too great, and this solution was not adopted.

Other components identified for circuit protection were resettable fuses (PTC) as well as transient voltage suppression (TVS) diodes. A PTC resettable fuse acts similar to a thermistor and is placed in series with the power entering the circuit. As more current is drawn through the device, its temperature increases due to a small internal resistance. At a certain trip current (temperature level), the device will rapidly enter shutdown (electrical resistance will increase several orders of magnitude). When the current drops below the trip level, the PTC will begin to cool, allowing the resistance to once again drop to its original low value. No reports of catastrophic failure were found for PTC resettable fuses; however, it was recognized that such a device would only protect again transient current, not voltage.

Transient voltage suppression diodes are similar to normal diodes; however, they are designed to handle higher power. Therefore, they can be used to clamp voltage surges in circuits. In circuit schematics, such a device is denoted as a zener diode. It was found that TVS diodes are common protection devices in automotive systems, and as with PTC resettable fuses, catastrophic failure has not been reported. However, TVS diodes will not protect a circuit against current surges.

An online resource was found that suggested the use of TVS diodes combined with PTC resettable fuses for circuit protection. Example circuits were provided as well as design criteria needed for selecting appropriate TVS diodes and PTC fuses. This website was linked to an automotive design page, further emphasizing that this solution is common to electrical design for vehicle systems. Therefore, it was decided that a combination of TVS diode and PTC resettable fuse would be appropriate to handle current and voltage transients that may occur in the circuit.

The next area of design involved the selection of voltage regulators for the system. All regulators are required to handle the current and voltage needed for subsystems while also conforming to the design specification of supporting a 10 to 30 volt input voltage. Before choosing components for the voltage regulation circuitry, the voltage level and precision of the needed regulators was identified by observing the different sensors and components selected for the complete main module system. It was quickly recognized that due to the sensors selected and speed needed for the system clock, both 3 and 5 volt operation would be required. Furthermore, since analog sensors were present on the board a 3 and 5 volt voltage reference would be needed for proper operation.

Current calculations were performed for each of the regulators and references needed based on the components they would support. From these calculations and the specifications previously identified, a Digi-Key search was conducted in order to find relevant parts. Many parts with standard packages were identified; therefore, the final component selection was based on lowest price.

One component selected during this analysis that the author proposes to change in the future was the 5 volt regulator (MIC2954). This device is a precision regulator chosen due to the fact that it would eliminate the need for a 5 volt reference on the board. However, the current that this component is able to handle is close to the exact needs of the Onboard Weather Station. Therefore, after a long period of time, the 5 volt regulator becomes warm. The author has already selected a replacement component that is a more standard part, which also makes it more suited to the design requirements of the project (KA7805ETU). Furthermore, a 5 volt reference has been selected (MAX6035), which will also need to be added to the board. These components combined are less expensive than the MIC2954 making the decision to switch reasonable.

The final aspect of the power input circuitry is the battery backup circuit. This circuit was omitted from the original design and board construction; however, design alternatives have been considered. A method considered initially was a large capacitor to provide safe shut down of the system. It was eventually decided that this method would not guarantee enough power to the system while the SD card finished operations. Furthermore, when the RTC specification was added to the system, this solution failed to meet design requirements.

Many approaches using diodes, batteries, and additional regulators in various arrangements were considered for this circuit. It was eventually decided that the best approach would be to use 4 AA batteries to power the system with 2 zener diodes (one for the 5 volt supply and another for the 3 volt supply). This setup would most effectively ensure proper voltage to the different circuits of the system, which is why it was selected.

4.1.2 Sensor Module

Although the sensors to include in the system were specified by the project sponsor, the placement of sensors in the system was omitted. Therefore, the design of the sensor circuitry involved decisions as to which sensors should be placed on the main module of the weather station versus the external module of the weather station.

Initially, the system layout called for all sensors to be placed on the main module and additional temperature, humidity, and barometric pressure sensors to be placed on the external weather modules of the system. This design was modified, since it was determined that the location of the main module in the race vehicle would not result in weather sensors collecting any information relative to vehicle performance, which is the main goal of the system. Furthermore, removing these sensors from the main module of the system would reduce the price of the main module, which is always a concern in system development.

Despite removing the weather sensors, g-force, ignition system voltage, and vehicle RPM were still monitored on the main module board. The g-force sensor was mounted in the main module due to the fact that a central vehicle location is necessary for accurate g-force measurements. Therefore, g-force sensors on the external modules, which are placed in various points of the vehicle, would not provide the customer with useful performance information. It was also decided that RPM monitoring should occur on the main module board. This decision was also made due to the central location of the main module, which would make it easy to route cables from different RPM locations. Furthermore, to prevent airflow disturbances, the external weather collection modules were required to be as small as possible with few wires protruding. RPM monitoring would result in extra cables being connected to these modules, which was unwanted; further resulting in RPM being placed on the main module.

Another design decision made for the sensor subsystem involved choosing a g-force sensor (accelerometer) for operation. From the design requirements, the only specification was for the device to have 3-axes and support g-force levels of at least ±7g. Although several accelerometers were examined, the device that stood out beyond all alternatives was the MMA7331LT manufactured by Freescale. This device is a 3-axis accelerometer that allows g-force collection of ±4 or ±12g; thereby, meeting both system specifications. Furthermore, the Freescale component is priced at $2.32, which is $10 lower than alternative accelerometers that also meet device specifications. Since the Freescale component was found to have reasonable accuracy and meet all design specifications, it was chosen for the design of the Onboard Weather Station.

A final design decision that was made for the sensor subsystem of the main module was how to connect the external data collection units to the main board. In the original design for the Onboard Weather Station, sensors were mounted on external boards and power, ground, and a data signal was run to the main module microcontroller for analysis. This setup has many restrictions. The sensors on the external modules cannot return analog data signals due to possible noise on the wires running through the vehicle back to the main module. Furthermore, heavily shielded cable must be used to avoid transients on the power lines running to the external boards. This setup also restricts future expansion of the data collection system. If a design of this type was used, the exact signal type and behavior of all sensors would need to be known for the main module to be designed. Any future addition of external sensor modules would require redesign of the main module as well. This was not desired; therefore, this method of design was rejected.

It was therefore decided that each external sensor module should have its own microcontroller to organize data collection. Data would be returned to the main module for storage using a bus setup. Many bus designs were considered for the Onboard Weather Station. The first examined was the actuator and sensor (AS) interface. This bus design is similar to the setup described in the previous paragraph; therefore, it was rejected with little additional research. The IEEE 1451 Bus protocol was also reviewed; however, this setup is specifically designed for a smart transducer interface to a microcontroller.

Finally, controller area network (CAN), local interconnect network (LIN), and FlexRay bus systems were reviewed. These systems are all designed for automotive applications making them ideal for the Onboard Weather Station design. FlexRay is a new automotive network communication protocol that is still in development by the FlexRay Consortium. It is faster and more reliable than CAN (the current standard), but it is also more expensive. Although, FlexRay is the fastest and most reliable of the bus systems observed, the cost of implementing FlexRay was unreasonable for this project. Furthermore, FlexRay has not yet been adapted to microcontroller projects; therefore, FlexRay development in a microcontroller environment would be difficult at the current time.

The LIN bus is also used with automotive networks in order to integrate intelligent sensors or actuators. It is generally implemented as a sub-network of a CAN bus, and is the cheapest of the bus design alternatives. A LIN bus is designed to have one master with up to 16 slaves. Collision detection is not present; therefore, all messages must be initiated by a master and result in one slave replying. LIN is also a slow network implementation as it can only reach speeds of 19.2kbit/s for a bus length of 40 meters. Although LIN is a viable alternative, it was determined that the bus speed and send-reply protocol would be too slow for Onboard Weather Station applications. For this reason a CAN bus was chosen for the final design.

A CAN bus is specifically designed to allow microcontrollers to communicate with each other without a central master coordinating computer. It supports bit rates of 1Mbit/s for network lengths up to 40 meters. Furthermore, since CAN was the first bus protocol introduced to the market, it is widely supported by microcontroller manufacturers. Therefore, design time for a CAN system would meet the constraints for this project. Added benefit from a CAN bus was the fact that power and ground could be provided through the CAN connection. Therefore, external sensor modules would require a single cable for power, ground, and sensor data transmission further allowing them to be compact and meet the size design requirement.

4.1.3 Processing Module

The main design decision required for the processing module was the choice of an appropriate microcontroller. All microcontrollers considered were manufactured by Atmel, since the project sponsor had all needed programming tools and software for Atmel systems. Before designing the processing module, all other subsystems of the main module were created. This was done in order to determine the necessary features a microcontroller would need in order to interact with the remaining subsystems of the device. Several features were identified that could be handled by either specialized microcontrollers or external transceiver chips. For this reason, decisions had to be made in order to determine which solution would best meet the requirements of the design.

The two subsystems that could be interfaced to the processing module through specialized microcontrollers or separate transceiver chips were the CAN bus and the USB to PC connection. A USB microcontroller was immediately eliminated from design decisions. This was done due to past experience with a USB microcontroller manufactured by Atmel. USB is a complicated standard with many requirements for correct device operation. While the Atmel microcontroller has the ability to properly implement the standard, many problems are encountered without a specific setup provided by Atmel. Furthermore, many other users of Atmel USB microcontrollers found that the system would not properly interface to a computer unless the Atmel USB development software was installed. A final concern was the poor documentation of Atmel USB microcontrollers. Therefore, due to the poor documentation, design errata, and interface restrictions, an Atmel USB microcontroller was not chosen for the system design.

A CAN microcontroller was eventually chosen due to the fact that CAN transceiver chips require a serial peripheral interface (SPI) to communicate with a microcontroller. Several other timing critical subsystems required the use of SPI or UART (which can be modified to behave like SPI); therefore, it was decided that interfacing a CAN controller through SPI would put system timing requirements at risk, which was not acceptable based on the design requirements. Furthermore, documentation and CAN libraries in C were available for Atmel microcontrollers, which would aid the design work. A final benefit of a CAN microcontroller is the fact that if the CAN interface of the microcontroller is found to function sub-optimally, a UART port could be converted to salvage CAN operation. However, if a microcontroller without CAN capabilities was chosen, the only design alternative would be the SPI or UART connection. Therefore, this flexibility also made the CAN microcontroller a good decision.

After all interface features were identified, a Digi-Key search was conducted in order to identify potential solutions. Several Atmel CAN microcontrollers were found belonging to two main product lines (AT89 and AT90). From initial research into the two lines, it appeared that the AT89 was going obsolete. Very few of these components were in stock, and all distributors had a long wait time. Therefore, the final decision was for the AT90CAN64 microcontroller.

4.1.4 Data Transfer Module

Since a USB microcontroller was not selected, a USB transceiver chip was a necessary component for the system. Research was not conducted into possible design alternatives for this chip, as the project sponsor had previously used a CP2102, which provides UART to USB conversions. Therefore, development boards and Assembly interface code for this device were available as was knowledge of design quirks. For this reason, the CP2102 was identified as a reasonable choice for the data transfer module design.

The second component of the data transfer module is the wireless transceiver. Several wireless transceivers were researched for this module including the Radiotronix RCT-433 transmitters and RCR-433 receivers, the Digi XBee and XBee-PROs, and the EmbedRF wireless device. Bluetooth modules were also examined for comparison including the Embedded Blue Transceiver, Parallax Bluetooth Module, and Parani-ESD by Sena. Although Bluetooth would allow the system to transmit data to some computers and phones without an adaptor, it was eventually eliminated due to its high price, complexity for microcontroller operations, and limited range. Furthermore, Bluetooth power consumption is higher than other alternatives.

Most of the remaining devices identified also had limited range, which would not meet the design requirement of ½ mile communication. For this reason, the only module acceptable for operation was the XBee-PRO. XBee devices support many wireless communication stacks; however, after research it was determined that the simple 802.15.4 protocol would best suit the needs of the project. An XBee-PRO has a line of sight range of 1 mile and is FCC certified. Therefore, it requires no additional approval to be used in the Onboard Weather Station application. Furthermore, it supports a data rate that was found to meet the design requirement of sending 1 data packet every sample period. For this reason, the XBee-PRO was the best choice for the wireless portion of the data transfer module.

4.1.5 Memory Module

The final subsystem of the main module is the memory module. Due to the rapid adaptation of secure digital (SD) cards in computers and microcontroller applications, it was quickly decided that an SD card would be the best solution for data storage. SD cards of every size meet the system requirement of storing 12 minutes of data. Furthermore, it was determined that an SD card would be a good solution due to the difficulty in finding an EEPROM or Flash chip with the necessary capacity for the system. Such a chip would be a rare component that would not have many potential replacement parts; therefore, it would not meet the design requirement of manufacturability.

A possible alternative identified was Compact Flash as it is used in the Corsa data acquisition system; however, this device requires a special adaptor for extracting data to a computer. Furthermore, Compact Flash does not have an easy interface to microcontrollers like SD cards, and requires wear balancing (writing to different locations equally) to prevent the device from wearing out after a few write/erase cycles. For these reasons, an SD card was seen as the best design alternative.

Since SD cards can be removed from the system, backup data storage was required to allow some run data to be preserved if users forgot to reinsert the SD device. It was determined that the backup data system should be able to store 1 run of data. The weather station would warn the user that data had been stored to this location such that the SD card could be returned to the device for subsequent runs. EEPROM and Flash chips were reviewed in order to find a device with sufficient memory to store 1 run of data from the Onboard Weather Station. Initial research showed that EEPROM chips did not have sufficient capacity. Therefore, Flash chips were examined. Several Flash chips were available; however, the product line with the largest capacity that was carried by a majority of vendors was the AT45DB series by Atmel. The AT45DB161 was chosen since it would be able to store 4.096 minutes of data, which is more than sufficient for capturing 1 run of information.

The final decision regarding the memory module was whether to store data using a FAT32 file system or through raw data storage. Both techniques are supported by SD cards. Initial research showed that the FAT32 file system would allow the SD card to be read on a computer with no additional software. However, write times for FAT32 are longer than writing raw data to the card. Exact write times for FAT32 filing could not be found in the SD card standard or on microcontroller SD card forums; therefore, it was decided that raw data writing would be used to ensure the device would meet timing requirements.

Although this decision is justified, it may be beneficial to switch to a FAT32 file system in the future. In order for this to occur, verifications would need to be made to ensure the device could meet timing requirements. FAT32 would allow the system to better organize run data for easy data extraction by the weather station software. Although it is possible for the weather station to extract data in the raw file format, further processing is needed in order to organize it into runs. Technically this is not a problem, since the computer processing is not time critical as it occurs after a race; however, the author believes FAT32 filing would provide an overall cleaner solution that should at least be examined in the future.

4.2 External Module Tradeoffs

The system requirements from Section 3 also resulted in the creation of a general block diagram for the external module of the Onboard Weather Station. This can be seen in Figure 5. This section will elaborate the tradeoffs made for the external data collection module. As in section 4.1, tradeoffs are organized based on the subsystem of the external data collection module for which they were made. Subsystems where tradeoff analysis was not performed will be omitted from this section.

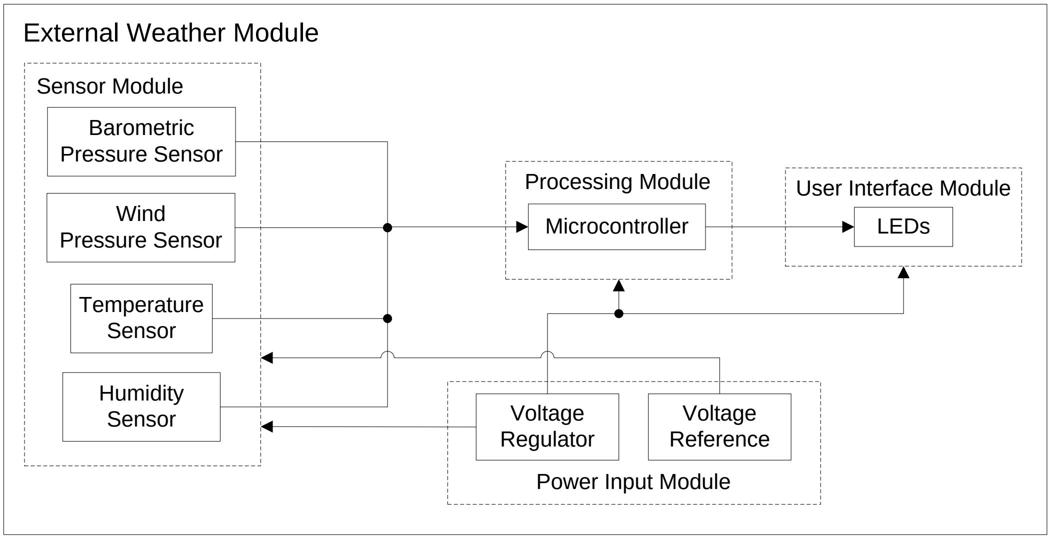


Figure 5: General Block Diagram of External Module

4.2.1 Power Input Module

As in the main module, the power circuitry of the external data collection module was also required to suppress transients as well as supply the needed voltage and current to the system. Due to the use of analog sensors, precision voltage levels were also needed for the external module circuitry. Though power is supplied to the external module from the main module through the CAN bus, it is unregulated in order to prevent voltage dropping below a required level during transfer. Therefore, the power entering the external module is provided directly from the ignition system as with the main module. For this reason, transient suppression is still required. As with the main module, a PTC resettable fuse and a TVS diode were used for current and voltage transient suppression. The rationale for these components is provided in Section 4.1.1.

All external module subsystems were able to operate off of a 5 volt supply. Furthermore, due to the simple design of the system, it consumed minimal current. The overall result was a precision voltage regulator being chosen to supply power to the system (MIC2954). This device eliminates the need of a voltage reference as the voltage regulator can also supply precision voltage levels. Unlike the main module, the MIC2954 is well suited to the operation of the external module. A single component solution reduces the required board space for layout improving the device's ability to meet the size design requirement. Furthermore, since the external module draws much less current under full load, a replacement precision voltage regulator can be found if the MIC2954 was to go obsolete.

4.2.2 Sensor Module

The main aspect of design for the sensor module involved selecting the sensors for collecting temperature, humidity, barometric pressure, and wind pressure. Several sensors were considered for collecting temperature. The final design choice came between the LM34CAZ, an analog temperature sensor, and the TMP36, a digital sensor. Both components had been used by the project sponsor in the past; therefore, tradeoff analysis was performed by comparing results from actual tests. It was decided that the LM34CAZ was a better selection as the output of the sensor was not influenced by factors such as inaccurate clock speed or resistor values. Furthermore, the LM34CAZ has greater accuracy than the TMP36 even disregarding external factors. The final reason that the LM34CAZ was chosen involved the fact that it was manufactured by National Semiconductor rather than Analog Devices (the manufacturer of the TMP36). Analog Devices is known to have long lead times even for standard components, which could be detrimental in a manufacturing environment. Therefore, in all aspects of design the LM34CAZ is a better choice.

Several humidity sensors were also considered for the system, but the final decision came between the HIH-4031, an analog sensor by Honeywell, and the HS1101LF, a digital sensor by Measurement Specialties and Humirel. As with the temperature sensors, the project sponsor also had experience with both of these humidity sensors and was able to provide insight into their operation and accuracy. The sensor chosen was the HS1101LF by Humirel due to slightly better accuracy and easier calibration. It was found that the Honeywell sensor had larger variation due to changes in temperature than the Humirel sensor. Although, the Humirel sensor is slightly more difficult to use due to its digital interface, it is a more standard component than the Honeywell alternative. Therefore, if it were to go obsolete, a replacement sensor would be easier to find. For these reasons, the HS1101LF humidity sensor was chosen.

The final sensors selected were pressure sensors. Although several sensors were reviewed for capturing pressure data, the selection was quickly narrowed to Freescale components. Freescale provides the largest range of pressure sensors all with the same basic interface. This setup is extremely suited for system development, as it is known that if one Freescale sensor goes obsolete, another exists with a similar interface. Furthermore, the Freescale pressure sensors were reasonably priced and could collect data to the desired accuracy. For this reason, Freescale pressure sensors were selected for both barometric pressure and wind pressure sensing.

Due to the setup of Freescale product lines, it was possible to use the same basic sensor for barometric pressure and wind pressure sensing. The main difference is the packaging of the two sensors: one provides an open surface and the other a closed tube for directed measurements. The ability to use the same sensor as the basis for both measurements reduced the development time of the system, and also provided a good argument for using Freescale pressure sensors.

4.2.3 Processing Module

The final subsystem of the external module that required a design decision was the processing module. As in the main module, this subsystem was to be composed of an Atmel microcontroller. In order to choose a proper device, the components selected for all other subsystems were considered. As in the main module, the primary decision that was required for the external module was whether to use a specialized microcontroller that supports CAN. After reviewing the CAN microcontrollers provided by Atmel, it was determined that a CAN microcontroller would provide too many features for the operations needed in the external module. Although more features would not cause problems, it would add complexity to the solution as well as increase the price of the external data collection modules. Therefore, it was decided that a CAN transceiver chip would be used and a standard Atmel processor would be identified.

The features required by the microcontroller in order to interface with all subsystems of the external module included an analog to digital converter (ADC), serial peripheral interface (SPI), system timers, and external interrupts. Nearly all Atmel processors contain these features; therefore, the search was limited to processors of lower cost. All results were in the ATTINY and ATMEGA product lines. The final processor chosen was the ATTINY88. Although the ATTINY48 would also meet the requirements for processing in the external sensor module, it was found that the random access memory (RAM) of this microcontroller was rather low, which could cause problems when expanding the external data collection modules in the future. In order to prevent a processor change from being necessary during expansion, the larger version of the processor was selected. Furthermore, the price difference between the two processors was negligible, making the final choice a good design decision.

A CAN controller was chosen based on forums related to CAN bus embedded applications as well as research through various distributors. It was found that the main manufacturer of CAN controllers is Microchip Technology. Microchip produces two types of CAN controllers, the MCP2510 which implements CAN V2.0A and CAN V2.0B as well as the MCP2515, which only implements CAN V2.0B. The MCP2510 was chosen to give the system flexibility when implementing the CAN bus. The AT90CAN64 microcontroller chosen for the main module processing unit can also implement both CAN standards; therefore, it was recognized that testing could be done by designing with the MCP2510 in order to determine which CAN type works best.

If CAN V2.0B is found to work equal to or better than CAN V2.0A, the MCP2515 could replace the MCP2510 in the future. This component is less expensive than the MCP2510, and also has the same footprint and pin out. Therefore, swapping these components would simply involve changing the embedded interface. From the datasheets of the two devices, it appears that the interfaces are also similar. Overall, the decision to use the MCP2510 is reasonable for the current system setup and also provides potential for reducing the cost of the system in the future.

5 Design and Implementation

As with the tradeoff analysis this section will be divided into main module design and external module design. A section will also be included regarding the software design. Schematics for the main module and external module can be seen in Appendix A and B, respectively. Furthermore, parts lists for the modules can be reviewed in Appendix E. A user's manual for the software is provided in Appendix C. Appendix D is reserved for the assembly source code.

5.1 Main Module Design

A detailed block diagram of the main module can be seen in Figure 6. The design will be broken down based on the subsystems that appear in Figure 6. Due to its simplicity, the signal conditioning module will be combined with the sensor module in the system breakdown.

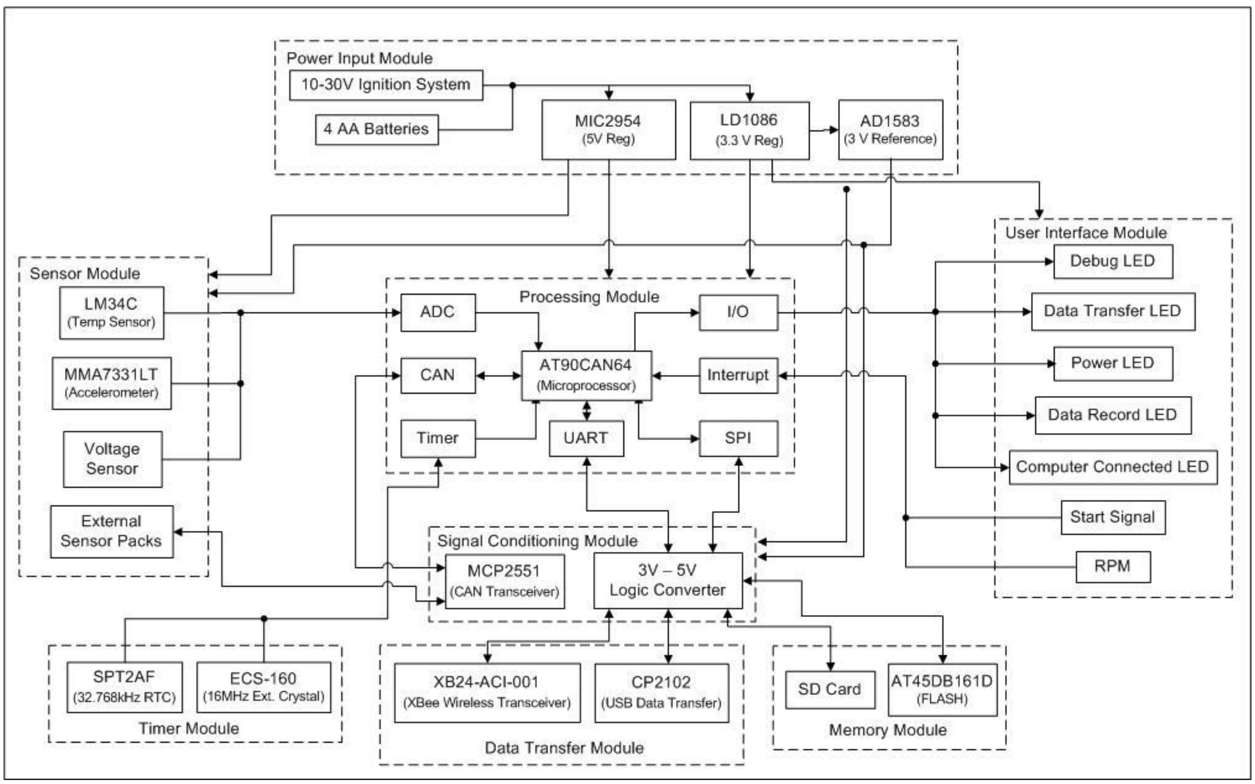


Figure 6: Detailed Main Module Block Diagram

5.1.1 Power Input Module

The power input module is responsible for providing power to all circuits in the main module as well as to circuits in the external modules connected to the system through the CAN bus. Due to the multiple voltage levels needed for correct operation of the main module, the power supply circuitry of the main module is composed of 3 voltage regulators or references. The other main components of the module are the RHEF070, which is a PTC resettable fuse, and the SMDJ30A, a TVS diode. Figure 7 is a representation of the power supply circuitry.

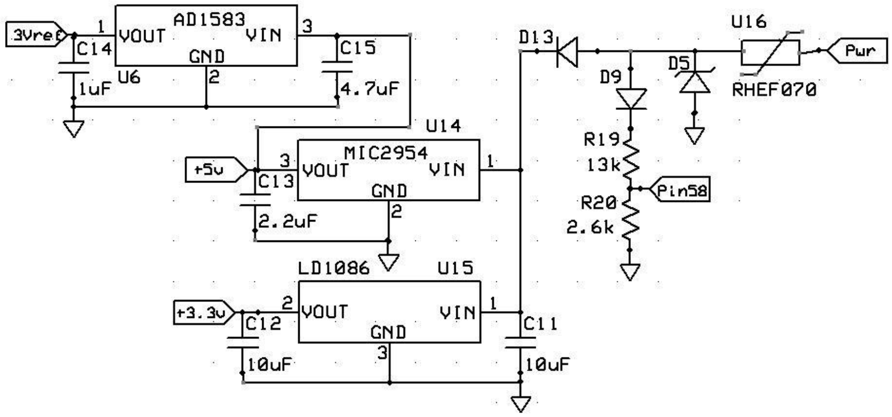


Figure 7: Main Module Power Input Circuit

The power supply module receives power from the ignition system of the vehicle through a barrier strip input. The power entry point is labeled Pwr in Figure 7. After entry the power is passed through a transient suppression circuit. In order to design this circuit, the expected voltage input and current requirements of the system were needed. From the design specifications, it was known that a 10 to 30 volt signal would enter the power supply module from the ignition system of the race vehicle. However, to complete this circuit a current estimate was required. To achieve this estimate, the system was broken down based on components that required a 5 volt supply and those that required a 3 volt supply.

Due to the number and type of components that operated on 3 volts, it was known that a precise voltage would not be possible with the size regulator required. Therefore, the current calculation was further broken down by 3 volt reference. A 5 volt regulator was found that was capable of providing both a precise voltage level and the current required for the 5 volt components; therefore, a 5 volt reference was not needed. The complete calculation table can be seen in Table 1.

Supply	Component	Quantity Running Concurrently	Single Component Current (mA)	Total Current (mA)
5 Volt Regulator:	PC900 (optoisolator)	5	5	25
	LEDs	5	10	50
	MCP2551 (CAN transceiver)	1	75	75
	LM34CAZ (Temp Sensor)	1	0.142	0.142
	AT90CAN64 (Microcontroller)	1	29	29
			5 V Reg. Total:	**179.142**
			Buffered Total:	**268.713**
3 Volt Regulator:	XBee (Wireless Transceiver)	1	215	215

	SD Card	1	200	200
	AT45DB161 (Flash)	1	17	17
	CP2102 (UART - USB Bridge)	1	26	26
			3 V Reg. Total:	**458**
			Buffered Total:	**687**
3 Volt Reference:	MMA7331LT (Accelerometer)	1	0.6	0.6
			3 V Ref. Total:	**0.6**
			Buffered Total:	**0.9**
			System Total:	**637.742**
			Buffered System Total:	**956.613**

Table 1: Main Module Current Consumption

From the current calculation, it was determined that the transient suppression circuit should be able to support 650mA of current in normal operation; however, it should not trip unless current draw exceeds 1 amp. The component chosen, RHE070, has a hold current (normal support current) of 700 mA and a trip current (current that will cause the fuse to trigger) of 1.4 amps. The maximum response time of the device is slow at 3.2 seconds due its size; however, the remaining circuitry was designed to compensate for this potential lag. The TVS diode chosen, SMDJ30A, is able to clamp voltage at 33.3 volts making it ideal for the main module transient suppression circuit.

After passing through the transient suppression components, the power enters the regulator portion of the circuit. Here is it divided between the 3 volt and 5 volt regulators. Power is not directly provided to the 3 volt reference because a 3 volt reference that supports a 10 to 30 volt input signal could not be found. Therefore, the signal entering the 3 volt reference is provided from the output of the 5 volt regulator. All regulator circuits were created based on specifications in the data sheets of the respective components. Therefore, all capacitor values are those suggested by the device manufacturer for stable operation.

The 5 volt regulator is a MIC2954 manufactured by Micrel. This device is a precision voltage regulator, which is equivalent to a high current voltage reference. The MIC2954 is a positive-fixed voltage regulator that is able to support current draw of up to 250mA. The device allows a voltage input up to 30 volts while maintaining a precision of ±0.025V. This precision was found to be sufficient for accurate analog to digital conversions in the microcontroller as well as sensor circuitry.

The 3 volt regulator chosen was a LD1086 manufactured by STMicroelectronics. This device is able to support 1.5 A of current while maintaining a voltage level of 3.3 volts (also acceptable for the 3 volt circuitry). The device is able to operate with an input voltage between 4.9 and 30 volts, which meets design specifications. The precision of the device is ±0.066V which was determined to be too large to ensure accurate accelerometer readings. For this reasons, the AD1583 manufactured by Analog Devices was chosen as a 3V voltage reference. This device has a tolerance of 0.1% making it ideal for the accelerometer circuitry.

Another feature of the power input circuit that can be seen in Figure 7 is the voltage divider that connects to Pin 58 of the microcontroller. This is placed before the voltage regulator circuitry yet after the transient suppression circuit in order to monitor the voltage levels being provided by the ignition system. This signal is passed into the analog to digital converter (ADC) of the microcontroller for monitoring and storing. When the voltage drops below a threshold level, the AT90CAN64 is informed through the ADC conversion result. It is then able to enter a low power mode. The resistors chosen for this divider provide a division by 6 on the input voltage to the system. Therefore, the 10 to 30 volt signal is reduced to 1.66 to 5 volts, which is acceptable to the microcontroller. The resistances were also chosen to be relatively large such to ensure that the voltage divider circuit would not draw excessive current (less than 1 mA).

It should be mentioned that the current requirements for the external modules were not considered in the power input circuit of the main module despite the fact that the external boards are powered from the CAN bus originating on the main module. Consideration for the external modules was not necessary in the main module due to the fact that raw, unregulated power is passed over the CAN bus directly from the barrier strip or input to the main module. This was done since the number of external modules connected to the CAN bus at any point in time is unknown. Therefore, the current requirement of the system as a whole is unknown. These unknowns would provide a large potential current range, which would make it impossible to properly protect the circuit from transients or regulate voltage. For these reasons, it was decided that unregulated voltage should be passed over the CAN bus.

Another benefit of this setup is the fact that the main module power is completely independent of the external module power making the design more modular than would have been achieved by combined power regulation.

The final aspect of the power input circuit of the main module is the battery backup circuit. This circuit was omitted from the prototype created in the project due to time constraints; however, a design was created that appears reasonable for future implementation. This design uses 4 AA batteries as well as 3 volt and 5 volt zener diodes for regulation. The purpose of this circuit is to allow the microcontroller, SD card, and Flash circuits to continue to operate after the ignition system of the vehicle is shut down. The SD card and Flash will be able to finish writes that may be in progress before entering a low power state, while the microcontroller will be allowed to sleep and only awake to maintain its real time clock. The battery backup circuit will be included in the release version of the Onboard Weather Station in order to ensure that memory corruption does not occur.

5.1.2 Sensor Module

The sensor subsystem of the main module is composed of the accelerometer or g-force circuit, the board temperature circuit, the voltage sensing circuit, and CAN bus circuit, which provides access to the external modules of the system. The voltage sensing circuit was described in Section 5.1.1; therefore, further description will be omitted from this section. The RPM circuit will be described in this section despite its original designation as a user interface to the system.

The accelerometer chosen for g-force measurement is a MMA7331LT manufactured by Freescale (Figure 8). A pull-up resistor was used on pin 10 to set the device for ±12g operation (±4g operation is also possible) in order to allow it to meet the ±7g design specification. The MMA7331LT is a 3-axis accelerometer that requires a precise 3 volt input signal for proper operation. It outputs a 0 to 3 volt signal on pins 2 through 4, which represent the x-axis, y-axis, and z-axis g-force levels, respectively. These lines are fed directly into the microcontroller for analog to digital conversion and storage. The sensitivity of the accelerometer is 83.6mV/g. Since the analog to digital converter of the microcontroller has a precision of 4.88mV when using a 5 volt reference, g-force changes of 0.058gs are detectable. This was determined to be more than sufficient by the project sponsor.

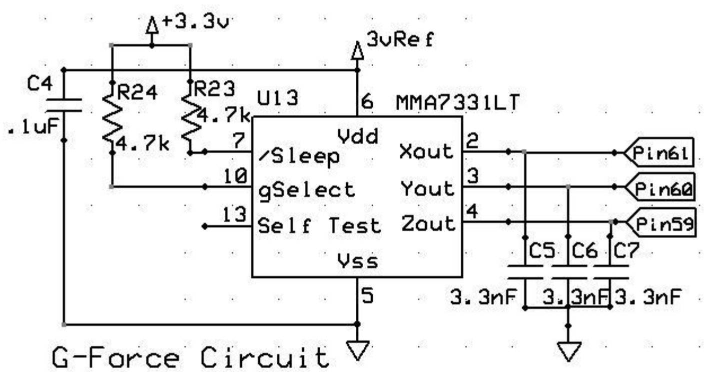


Figure 8: Main Module Accelerometer Circuit

Single-pole switched capacitor filters are included within the accelerometer package; thereby, eliminating the need for external components to set the cut-off frequency of the device. Despite this, the manufacturer suggests that 3.3nF capacitors be used on the outputs of the accelerometer to reduce clock noise from the switched capacitor filters. Decoupling capacitors on the power supply lines are suggested for further noise immunity. All circuit suggestions provided by the manufacturer were included in the Onboard Weather Station system to ensure that the most accuracy possible was achieved.

The temperature circuit of the main module was an addition made following the start of the project. Portatree decided that it may be beneficial to monitor the internal temperature of the system in order to determine if board venting would be needed in the future. Therefore, an LM34CAZ was selected (Figure 9). The LM34CAZ is a Fahrenheit temperature sensor capable of measuring temperature to 1° accuracy. The device has a linear scale resulting in 10mV increase for each degree Fahrenheit, and is guaranteed to operate from -50° to 300°F. Therefore, it would be able to monitor the limited change of board temperature without a problem. Furthermore, since the voltage change for each degree Fahrenheit is greater than 4.88mV, all temperature changes will be recognized by the microcontroller. An actual precision of 0.488°F is achieved.

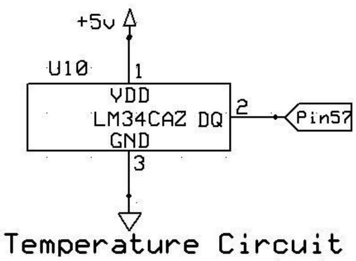


Figure 9: Main Module Temperature Circuit

As can be seen in the figure, the LM34CAZ requires no external components for proper operation. Therefore, once power and ground are supplied, the device will provide analog output signals relating voltage to temperature on pin 2. The result will be passed to pin 57 of the microcontroller, which is connected to an ADC port.

The RPM circuits for the system (Figure 10) are composed of a PC900 optoisolator. RPM signals in a race vehicle are a series of pulses with amplitude of 12 volts whose frequency depends on the revolutions of the motor. The RPM value is also dependent on the number of cylinders as well as the stroke type of the motor; therefore, two vehicles with the same RPM pulse signal could actually have different RPMs. The main focus of the RPM circuit was to detect the frequency of the pulse signal provided to the main module. Using the frequency and information provided by the user, the weather software is able to convert the frequency to an actual RPM value.

An optoisolator was chosen for this circuit in order to isolate the microcontroller input pins from the vehicle power supply. Furthermore, this device was able to reduce the RPM voltage signal from 12 volts to 5 volts without distorting the frequency that the microcontroller was required to measure. The optoisolator selected is a digital output device with a 1µs response time. Therefore, it provided an ideal input signal with little distortion to the signal period. All output signals from the optoisolators were fed into external interrupt pins of the microcontroller. Thus on every rising edge of the RPM signal, the microcontroller would be interrupted in order to take a time measurement of the RPM period. In this way, the RPM signal from the vehicle could be measured.

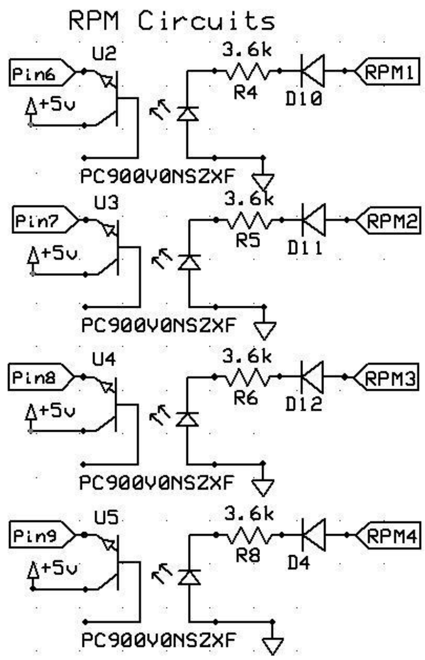


Figure 10: Main Module RPM Circuit

When the circuit was first designed, the input resistors to the optoisolator chosen were too large. This resulted in insufficient current being input to the device, which caused the internal infrared component to fail to light. The overall result was failed operation of RPM signal detection. By reducing the resistor value, the system was found to work properly. Another initial problem with the RPM circuit was the fact that the output from the PC900 optoisolators was a high-impedance active low signal. Therefore, for proper operation, a pull-up resistor is needed on the output line. Luckily, Atmel microcontrollers have internal pull-up resistors, which prevented modifications from being required to the printed circuit board created.

The final circuit in this module is the CAN bus, which is used to connect to the external data collection modules of the system (Figure 11).

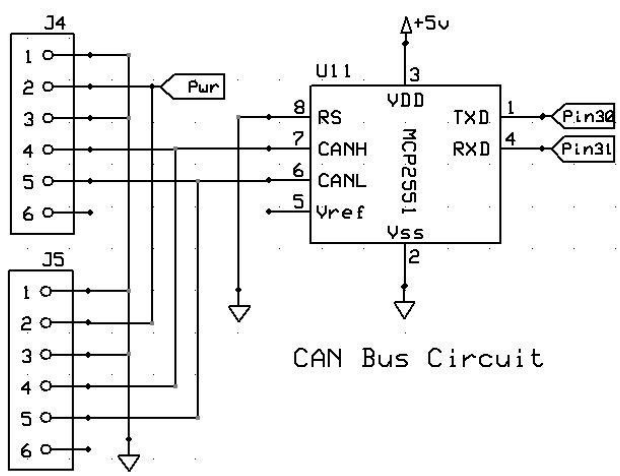


Figure 11: Main Module CAN Circuit

Although a CAN microcontroller was chosen for controlling CAN operations, in order to properly setup CAN bus signals a CAN transceiver is also required. The CAN transceiver chosen is a MCP2551 manufactured by Microchip. The inputs to the CAN transceiver are pins 30 and 31 (the CAN output pins) of the microcontroller. The chip is able to translate CAN transmit and receive signals to differential voltage signals, which are then sent over the CAN bus through the connectors seen in the left of the figure. The CAN transceiver also performs signal detection as specified by the CAN standard.

Two CAN bus connectors were designed into the main module despite the fact that the system can support 4 external modules. This decision was made in order to reduce the number of connectors and space required on the main module printed circuit board. Therefore, rather than all external modules being directly connected to the main module, they can be daisy-chained and access the main module through the CAN bus. In this way, the CAN bus is able to increase the modularity of the system.

Many CAN connectors can be used that comply with the CAN standard; however, the connector that is most common is a DB-9. For this reason, 2 DB-9 connectors were used on the main module. The pin-out for this connector can be seen in Table 2.

Pin Number	Signal Name
1	Reserved
2	CAN_L
3	CAN_GND
4	Reserved
5	CAN_SHLD
6	Ground
7	CAN_H
8	Reserved
9	Power

Table 2: CAN Connector Pin-Out

The differential signals were connected to the transceiver chip while the power and ground signals were connected to the raw power input to the system. The shield connection is also connected to ground; however, it should be noted that this connection should not be connected to the ground of any external modules otherwise the purpose of the shield will be defeated.

5.1.3 Timer Module

The timing circuit for the main module of the Onboard Weather Station consists of two external crystals: one at 16MHz and the second at 32.768 kHz. Both crystals are balanced with the necessary capacitors in order to ensure that they oscillate at the correct frequency. The timing circuitry can be seen in Figure 12.

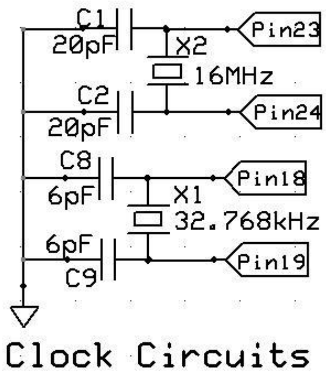


Figure 12: Main Module Timing Circuit

The 16MHz crystal was included in the system in order to have the ability to run the AT90CAN64 microcontroller at its maximum possible frequency. This was necessary in order to ensure that the 20ms timing requirement for the maximum sampling rate supported by the system could be achieved. An external crystal was also used as the system clock due to the fact that Atmel internal clocks are known for their inaccuracies. Since many aspects of the weather station are timing critical or require accurate time measurements (such as the RPM frequency measurement), an external crystal was deemed the best choice. Many 16MHz crystals are available. The ECS-160 was chosen due to its reasonable stability and tolerance (±50ppm and ±30ppm, respectively) as well as its standard package, which would make it easy to replace should it become obsolete.

The 32.768 kHz crystal was selected to perform timing for the real time clock in the weather station. A frequency of 32.768 kHz is ideal for the system as Atmel timers are specially designed with prescalers (clock dividers) that can divide the 32.768 kHz frequency such that an 8-bit register overflow results in the timing of a 1 second interval. Since interrupts are also provided for 8-bit timer register overflows, a real time clock can be created with relatively little processing. The crystal chosen is a SPT2AF manufactured by Seiko. This component was chosen since Portatree uses this crystal on another device they manufacture.

5.1.4 Data Transfer Module

The data transfer module of the system is composed of two devices: the CP2102 for USB transfer (Figure 13) and the XBee-PRO for wireless transfer (Figure 14). The CP2102 is a UART to USB bridge manufactured by Silicon Laboratories. The device connects to the universal asynchronous receive transmit (UART) of the microcontroller through pins 27 and 28. The UART signals are converted to conform to the USB 1.1 standard and transmitted through a Type B USB connector to a PC. Originally a development board for this component was purchased from SparkFun Electronics in order to test device operation. Although helpful in learning about the CP2102, the circuit from the development board could not be used in the Onboard Weather Station as it powers the system it is connected to through the USB port of the computer.

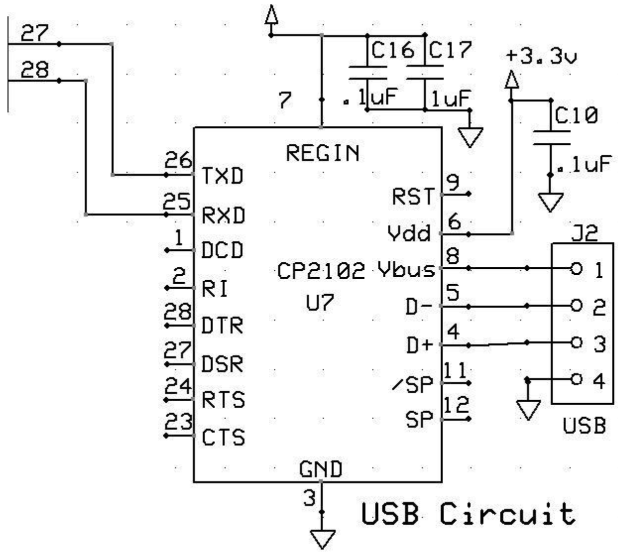


Figure 13: Main Module USB Circuit

Other than the signal lines, the CP2102 only requires power and ground for proper operation. Decoupling capacitors are included in the circuit as specified for correct operation in the manufacturer's datasheet. As can be seen, the CP2102 requires 3 volts for operation. The device is also able to support 5 volt power input due to an internal voltage regulator; however, it was decided to power the device from the 3 volt supply due to current limits on the 5 volt regulator. A USB port consists of 2 data lines, a voltage line, and a ground line. USB data transfer is a complicated procedure of alternating the signals on the 2 data lines. An explanation of this will not be given, since the USB conversion chip is able to handle all details of communication.

The XBee-PRO wireless transceiver also interfaces to the microcontroller through UART. Since the AT90CAN64 is equipped with 2 UART connections, device multiplexing is not necessary. For proper communication, the XBee must only connect to the transmit and receive UART lines of the microcontroller. Handshaking control signals are not necessary. The only remaining signals required for proper operation are the 3 volt supply line and ground.

The XBee-PRO uses the 802.15.4 communication protocol in order to transmit data up to 1 mile. It has a wireless transfer speed of 250kbps and a maximum UART speed of 115.2kbps. The device operates at a wireless frequency of 2.4GHz, which is part of the industrial, scientific, and medical (ISM) radio frequency (RF) band. Other common devices that use this band are cordless phones and microwave ovens. The XBee uses error checking and addressing in order to ensure robust wireless transfer. Retry and acknowledgment are included in every transmission. Furthermore, the device supports AES 128-bit encryption. The device is FCC approved, further making it an ideal drop-in solution for the Onboard Weather Station.

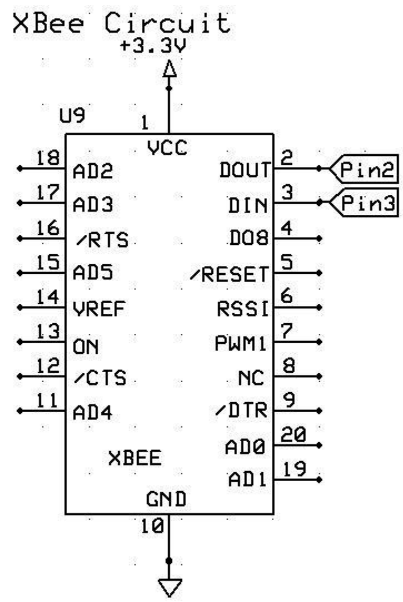


Figure 14: Main Module XBee-PRO Circuit

5.1.5 Memory Module

The memory module of the system is composed of an AT45DB16 Flash (Figure 15) as well as an SD card circuit (Figure 16). The main form of memory used in the system is the SD card, the smallest of which (512MByte) is able to record 1048 minutes (equivalent to 17 days) of data with a 20ms sampling period. Therefore, this device more than meets the system requirement of storing 12 minutes of data. The Flash chip is included in case the user removes and forgets to reinsert the SD card between rounds. The Flash controller is able to store 16MBit of data, which is equivalent to 4096 pages of 528 bytes. This translates to 4.096 minutes of data storage time, which will allow 2 runs (assuming 1 min and 30 sec runs) to be saved.

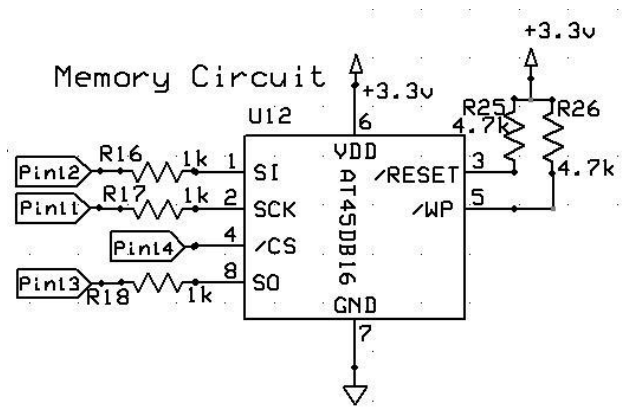


Figure 15: Main Module Flash Circuit

The AT45DB16 is an integrated Flash chip manufactured by Atmel. The interface to the device is a serial peripheral interface (SPI) making communication with the microcontroller straightforward. The resistors seen on the SPI lines are included to prevent contention between the Flash chip and the Atmel programmer, which also uses SPI. The 1kΩ resistors give precedence to the Atmel programmer as specified in the microcontroller datasheet. Since the reset and write protect features of the chip are not necessary for system operation, they were tied high through pull-up resistors. Unlike some memory devices, the chip select of the AT45DB16 Flash is needed for proper device operation. Therefore, this line is connected to the microcontroller as well. The chip select line is also necessary due to the fact that multiple modules interface with the microcontroller through its single SPI connection.

The AT45DB16 supports a 66MHz SPI frequency, which is run at 8MHz (maximum frequency supported by the microcontroller) allowing data to be rapidly transmitted to the device. Furthermore, it has a minimum duration of 100,000 program/erase cycles per page. The main concern in selecting a component of this type other than capacity is write speed as the device must be able to match the pace of the sample period. Unfortunately, in Flash prior to a device write a page erase must occur. Therefore, although it can write data to memory in 6ms, it takes 40ms for the combined page erase and write. It was found that the SD card had similar limitations.

Since the erase and write procedures of both the SD card and Flash are self-timed, a solution was achievable. Data from 3 sample periods are recorded to a buffer within the microcontroller resulting in 60ms of data being buffered (assuming the smallest sample period). At this point, the data is written to the selected memory. The system is designed such that 3 sample periods of data corresponds to 512 bytes of data, which fits within the page of both the SD card and Flash devices. Therefore, only a single page write is required, which can occur in less than 60ms. In this way, the timing requirements for the storage devices are met.

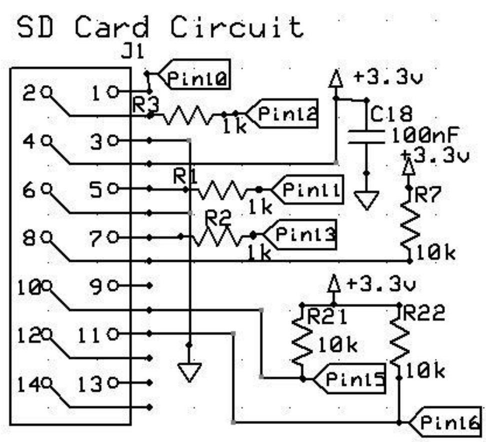


Figure 16: Main Module SD Card Circuit

The SD card also communicates with the microcontroller through the SPI connection. Unlike the Flash chip, the chip select of the SD card is not needed to control read and write operations. It was connected to the microcontroller in this implementation due to the fact that other devices also require the SPI for communicating with the microcontroller. As with the Flash, the SD card is setup as a slave to the microcontroller system. Therefore, the microcontroller provides the device with a clock signal and input data. The SD card provides a single return data signal to the microcontroller. Although SPI is not the main interface to an SD card, it can easily be setup and is still able to operate up to 25MHz, which is beyond the speed the microcontroller can communicate. The number of program/erase cycles for an SD card is manufacturer dependent.

A brief analysis of the rationale for memory requirements will be provided before moving to the next section of the report. In order to determine the amount of memory the system would require, the first consideration that was required was the number of external modules in the system and the amount of data that each of these modules would require. At the current time, the system is designed to support 4 external modules with each module recording 32 bytes of data. The 32 byte number was achieved by looking at the number of available input/output ports on the ATTINY88 after all necessary control inputs and outputs are reserved. It was found that 8 analog inputs and 8 digital inputs would be available for data collection. An Atmel ADC has 10-bit accuracy; therefore, 2 bytes are required to store results from any analog input. Furthermore, the largest timer provided by Atmel has a 16-bit granularity. Therefore, 2 bytes is also a reasonable size estimate for digital input data. The overall result is 32 bytes of data.

It is recognized that this estimate is high, since the current external modules only require 12 bytes of data storage; however, this adaptability was desired by the project sponsor. In the future, this could be reduced to 16 bytes per external module, which would allow 8 modules to be included in the system. This setup would most likely also reduce the amount of unused memory storage space.

After considering space required by the external modules, main module storage space was also considered. The main module is required to store 4 RPM readings of 2 bytes each, 1 voltage reading at 2 bytes, 3 accelerometer readings at 2 bytes each, and a temperature reading also at 2 bytes. The overall requirement for the main module is 18 bytes of storage. Table 3 summarizes the findings.

Data Source	**Number of Bytes to Save**	**Number of Devices**	**Total Space Required (Bytes)**
Temperature Sensor	2	1	2
RPM Inputs	2	4	8
Voltage	2	1	2
Accelerometer	2	3	6
External Modules	32	4	128
		Total:	**146**

Table 3: Memory Calculation Table

Therefore, 146 bytes of data are required for each sampling period. With a page size of 512 bytes, this corresponds to 3.5 samples per page. Since it would be difficult to store half of a sample to a page, it was decided that 3 samples should be stored per page. This would also leave buffer space in case extra sensors or data from the main module is to be stored in the future.

The time capacity calculations (amount of run time a particular memory solution can store) are completed by determining the number of sampling periods a particular memory device can support between downloads. This number is them multiplied by the sample period to get a time estimate. These estimates may be slightly high as additional overhead occurs due to run start markers (1 page of data); however, they do give a general idea of what is to be expected from each type of memory.

5.1.6 User Interface Module

The user interface module is composed of a series of LEDs (Figure 17) as well as the two-step input circuit (Figure 18). The LEDs are super-bright, directed LEDs manufactured by Bivar Corporation. They were chosen due to their ability to aptly catch the driver's attention should an error occur prior to a run. One green, 1 red, and 3 yellow LEDs are used in the circuit. The green LED is a power LED that is also used to signal when the computer is connected to the microcontroller. The red LED is lit when the device is recording data. The yellow LEDs are used for error codes.

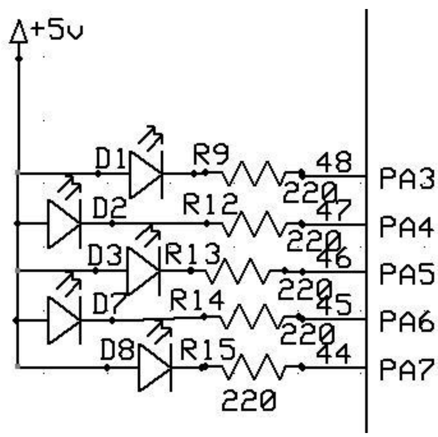


Figure 17: Main Module LED Circuit

The LED circuit is setup to draw approximately 10mA of current, allowing them to be safely driven from any digital output port of the microcontroller. They are setup to be active low in the system, thus requiring the microcontroller to sink current.

The two-step input circuit uses a PC900 (also used in the RPM input circuits). Once again, the desire was to isolate the vehicle power from the microcontroller input pins. Furthermore, the component is able to reduce the input voltage from 12 to 5 volts, which is a reasonable level for a microcontroller. As with the RPM circuit, the internal pull-up resistors of the microcontroller are enabled on the line connecting to the PC900 two-step input in order to ensure proper device operation. The PC900 was chosen for this circuit for similar reasons as the RPM input circuit (fast response time, isolation, and digital output). Therefore, the details will not be repeated in this section.

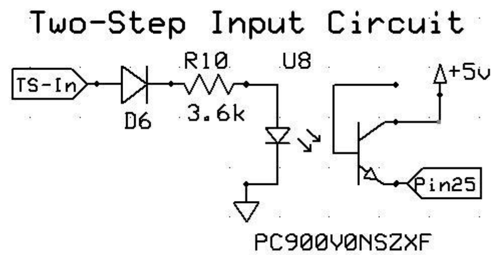


Figure 18: Main Module Two-Step Input Circuit

Another portion of the user input module that was not captured in either schematic image is the manual switch to trigger data record. Most switches interfaced to race cars are designed to be connected to the vehicle ground on close. Therefore, the circuit for the manual switch consists of a diode, to protect the system against a reverse voltage connection, which is directly connected to pin 5 (a digital input pin) of the microcontroller. The pull-up resistor of the microcontroller is enabled to ensure proper operation when no switch is connected and when the switch is not in the closed position.

5.1.7 Processing Module - Embedded Code

The processing module is composed of an AT90CAN64 microcontroller manufactured by Atmel. The device has an 8-bit core with 64Kbyte program memory, 2Kbyte EEPROM, and 4Kbyte RAM. It can operate at 16MHz, which was deemed adequate for meeting timing requirements. Furthermore, the device has CAN, SPI, and UART connections as well as timers, external interrupts, and ADC peripherals. As with most microcontrollers, it has digital input/output pins for general operation. From this analysis, it was determined that the device would be able to interface with all subsystems of the main module allowing proper system operation. The pin-out with labels corresponding to the systems connected to each pin can be seen in Figure 19.

Two important connections that were not described in previous sections include the joint test action group (JTAG) connector, which is used for programming and debugging of the main module of the Onboard Weather Station as well as the in-system programmer (ISP). The JTAG connection is included in the Onboard Weather Station design in order to allow the embedded code to be debugged through JTAG's boundary scan feature. A boundary scan is able to review all registers that are not part of the internal core of the processor. Breakpoints can be inserted in the embedded code in order to halt the processor when a certain boundary condition is found. At this point, all external registers of the system can be reviewed. In this way, the JTAG boundary scan is an invaluable resource for system development.

The ISP connection is included due to the fact that a pin required for JTAG programming is also needed for analog to digital conversion of the temperature sensor signal. After reviewing the Atmel datasheet, it was determined that JTAG pins should not be multi-tasked to other system operations. Therefore, to ensure proper programming and analog to digital conversion in the final system, the ISP connector must be used. The ISP connector interfaces to the microcontroller through SPI in order to allow programming. Although SPI is used to connect several other subsystems to the microcontroller, the chip selects prevent contention from occurring. The ISP connection is given precedence as specified in the Atmel datasheet in order to ensure proper programming. Since the ISP connection will only be used to program the boards during production, the connection will not occur during normal device operation; thereby, eliminating the risk of resource contention.

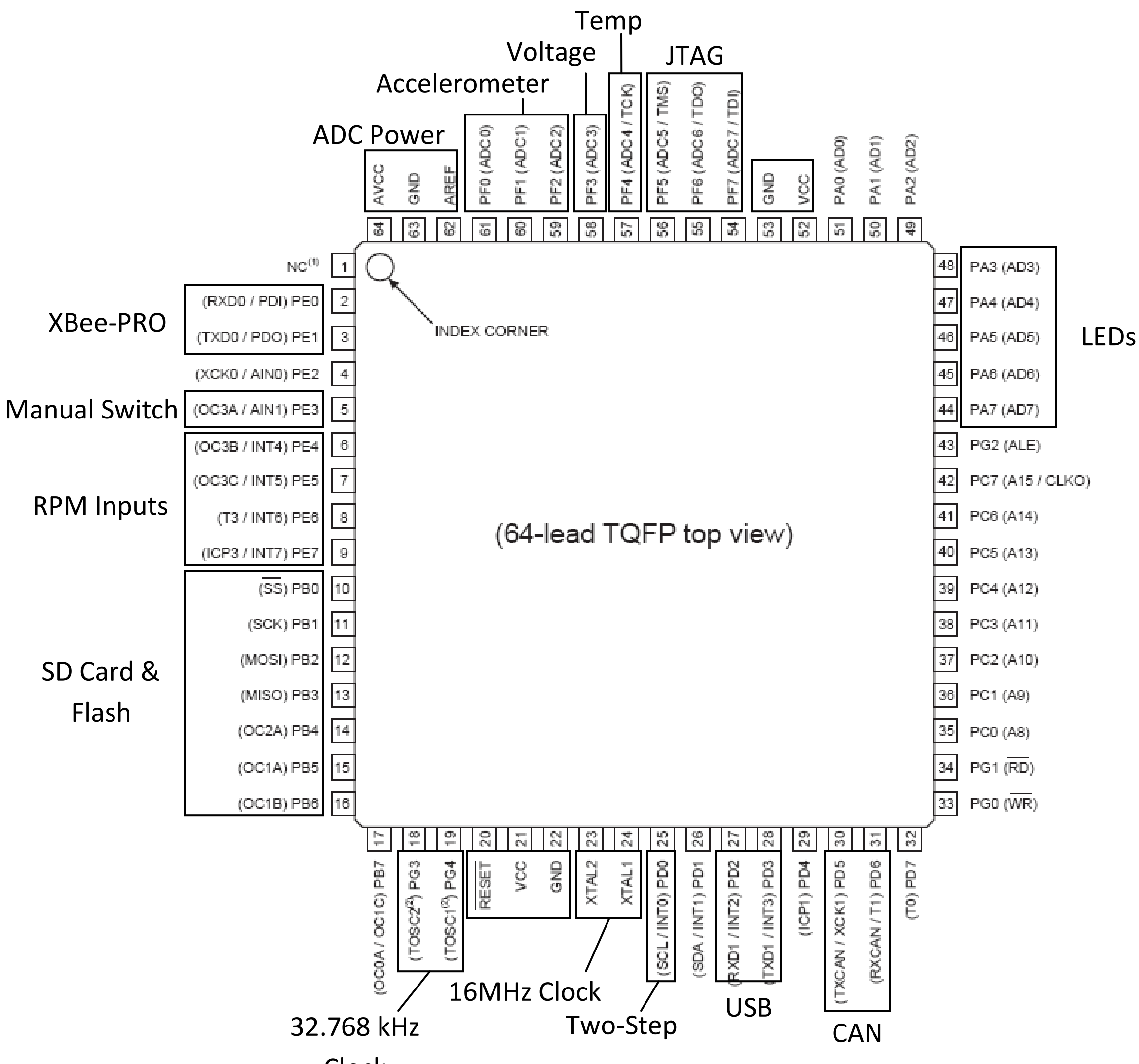


Figure 19: AT90CAN64 Pin-Out

The final circuit related to the processing module is a low-pass filter connected to the analog voltage supply of the processor. The analog voltage supply is a power input line to the processor that is specifically used by the analog to digital circuitry. For this reason, additional noise immunity is needed in order to guarantee accurate conversions. The filter implemented was suggested by Atmel for optimal performance of the ADC circuitry. It is composed of a 10µH inductor connected to 5V power and a 0.1µF capacitor connected to ground.

An important aspect of the processing module is timing of functions within the sampling loop. Therefore, prior to beginning the firmware, a list of operations that must occur in the sample loop was created in order to determine if the minimum sample period (20ms) could be achieved. The first operation considered was the analog to digital conversions made within the main module. From the AT90CAN64 data sheet, it was found that 13 ADC clock cycles are required for a conversion. This is equivalent to 104μs per conversion since a 125 kHz ADC clock is used. In the main module, 5 conversions are necessary, which accumulates to 520μs of conversion time.

The next operation observed is the CAN transmission. The CAN bus is operated at its maximum speed of 1Mbps (mega bit per second). Since the maximum data frame size of a CAN transmission is 8 bytes, 4 transmissions are required per module to transfer 32 bytes. Even though the external weather modules do not collect 32 bytes worth of data, in order to create a general protocol, it is necessary that the system be setup as though 32 bytes are transferred by each module. A CAN data frame with 8 bytes of data is 111 bits long based on the CAN protocol. In order for 4 CAN data transmissions to occur, 4 start transmissions are required from the main module. These are also 111 bits long. Therefore, a total of 2220 bits are required for the complete CAN transmission. At 1 Mbps this corresponds to 2.117ms of transmission time.

Data storage must also occur within the 20ms sample period; however, since this operation is self-timed by the Flash and SD card devices all that must be considered is the SPI transmission time. The SPI is setup to operate at 8MHz. In the Onboard Weather Station system, 522 bytes of data must be transmitted over SPI to either Flash or the SD card in a sample period. The 522 bytes includes data to write as well as the write commands. At 8-bits per byte, this corresponds to 4176 bits of data, resulting in a transmission time of time of 522μs.

The final operation to be performed during the sample period is XBee transmission. As with the memory devices, the only timing that must be considered is data transfer time as the wireless module handles the details of transmission once data is received. The microcontroller is not required to wait for the transmission to complete. No commands are required by the XBee module in order to initiate transmission; therefore, only the data to be sent must be transferred. The XBee module is connected to the microcontroller through the UART interface, which is setup to operate at the maximum speed supported by XBees, 115.2kbaud.

Two considerations were made for data transfer to the XBee transceivers: sending 512 bytes and sending 146 bytes. A data send of 512 bytes corresponds to transferring 3 sample periods of data; therefore, it would only be required every third sampling period. Transfer of 146 bytes would be required once per sampling period. In UART, a start, stop, and parity bit is appended to each byte resulting in a total transmission of 1606 bits for a single sampling period of data and 5632 bits for 3 sample periods. It was found that a data send of 5632 bits would require 48.9ms while a transmission of 1606 requires 13.95ms. Therefore, it is not possible to transfer 3 sample periods of data to the XBees in a single sample period; transmission must occur each sample period when it is enabled.

Table 4 shows a summary of the timing requirements. It was found that all operations could be completed in 17.169ms. Therefore, in theory timing should be met.

Device	Time Required (ms)
ADC	0.520
CAN	2.177
SPI	0.522
UART	13.95
Total:	**17.169**

Table 4: Processing Module Timing Calculation

The embedded code for the processing module is written in Atmel Assembly as specified in the project requirements. Four main files compose the assembly program: Interrupts.asm, Common.asm, Computer.asm, and MainWeatherModule.asm. Interrupts.asm contains all interrupt subroutines for the system. Common.asm contains subroutines commonly used by the system. Computer.asm contains the code to operate when a PC is connected to the Onboard Weather Station, and MainWeatherModule.asm contains the code to run the general operation of the weather station. A flow chart for MainWeatherModule.asm can be seen in Figure 20. Only the main program and computer interface will be described in the next few pages as the remaining functions can be found in other designs and were only used for system support.

The main operation of the program is as follows. On hard reset, the device initializes the timers, ADC, RTC, RPM interrupts, UART connections to the XBee and CP2102, manual input switch, two-step input, LEDs, and the SPI connection to the SD card and Flash. Run parameters are also requested from the EEPROM and stored in RAM variables for quick, easy access. At this point, the program reaches the power recovery point, or the entry point to the system when recovering from low-power mode (system startup under normal conditions).

The device checks to see if XBee wireless transfer is enabled. If this is true it proceeds to initialize the XBee. Otherwise, it skips to initialize the attached memory device. In order to initialize the XBee, it is necessary to ensure that the UART baud rate is set to 9600, as this is the startup transmission rate of the XBee modules. This is done by setting the UBRR0H and UBRR0L registers of the AT90CAN64 microcontroller. Necessary precautions were taken to ensure low error for the selected baud rate based on equations in the Atmel datasheet.

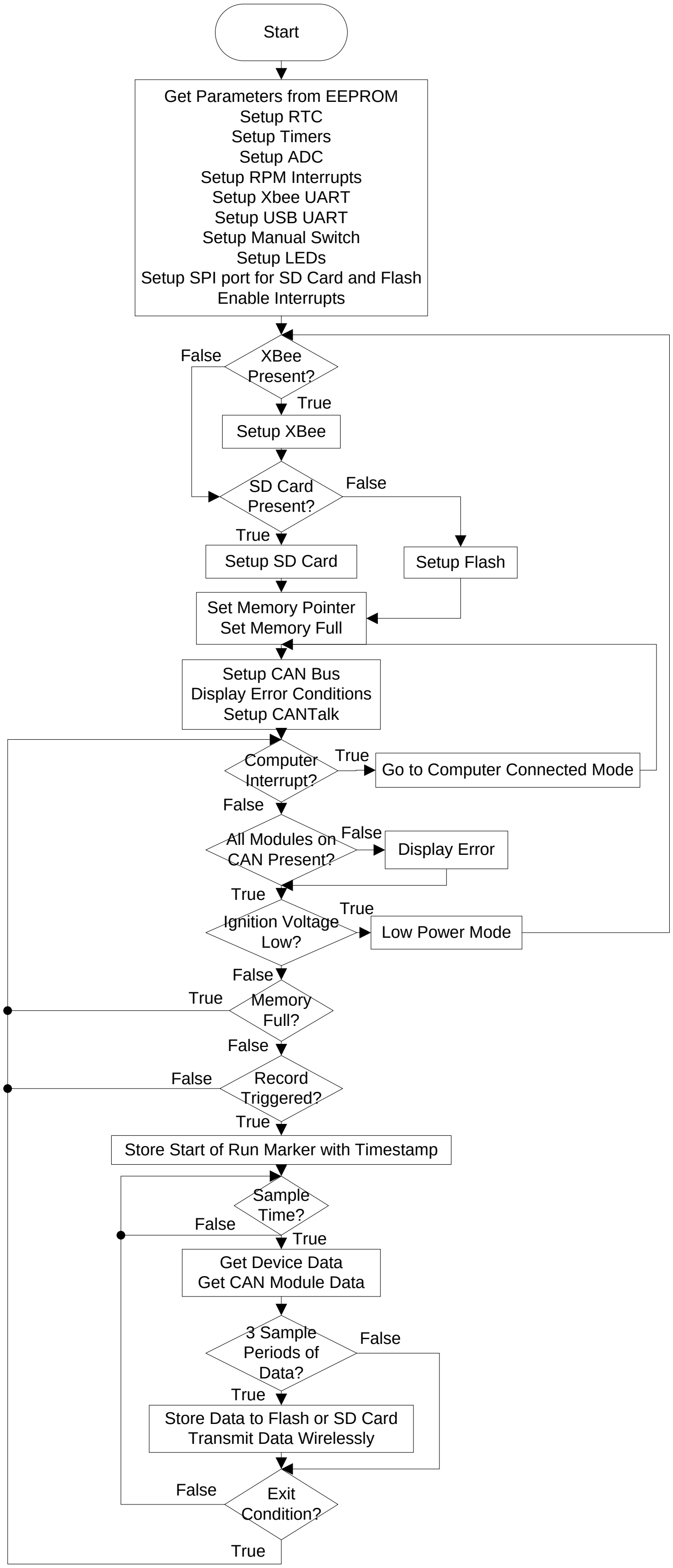


Figure 20: Main Weather Module Flowchart

After the baud rate is set, control commands are sent to the XBee in order to enter system setup mode. At this point, the destination address for wireless transmission is set as well as the new baud rate to allow 115.2kbaud transmission with the device. Once complete, the UART baud rate for XBee communication is also raised to 115.2kbaud. If an error is received during any point of XBee setup, an error flag is set to inform the user that wireless transfer will not occur during this system run cycle.

At this point, the SD card connector is checked in order to determine if an SD card is present. If the connector is found to be empty, Flash is initialized. Otherwise, the SD card initialization sequence is continued. The external write protect switch of the SD card is checked next. If it is found to be set, the Flash is initialized and a warning light is set for the user. If the SD card is found to be present and not write protected, SD card initialization occurs.

In order for the microcontroller to initialize the SD card, the SPI interface must be set to a clock speed lower than 400 kHz as specified by the SD card standard. In the embedded program written to interface with the SD card, the clock speed is reduced to 250 kHz before sending the initialization command sequence to the SD card. This sequence also calculates the size and type of the SD card connected in order to determine the last memory address available as well as the memory pointer increment to use for data storage.

If the SD card initializes properly, a flag is set to inform the system that the SD card should be used for data storage. At this point the memory pointer for the SD card is set to reference the next available memory location on the SD card. In the original version of the code, this was done by a loop through SD card memory in order to find a blank location. After initial testing, it was realized that such a loop could potentially take 139 minutes to initialize (over 2 hours) assuming a 2GB SD card that is almost full. Since this is not practical for the Onboard Weather Station system, a new setup was created.

At the current time, the first page of the SD card is loaded with 'SDPTR' to denote that it is currently being used with the Onboard Weather Station system. Immediately following this string is the 4-byte memory pointer to the next free location of the SD card. If an SD card does not have this string, it is known that it has never been used in the Onboard Weather Station. A card erase is performed, and 'SDPTR' is saved to the device along with the address of the first SD card memory location.

After the address is received, it is compared with the maximum address for the SD card. If found to be lower than the maximum address, the SD card full flag is cleared and the system proceeds to initialize the CAN bus; otherwise, the SD card full flag is set. If an error occurs during any point of initialization, an SD card failure LED is lit, and the device proceeds to initialize the on-board Flash.

Flash initialization is simpler than that for the SD card as it is a standard SPI peripheral. Furthermore, since the Flash component cannot be removed from the board, it is possible to maintain a copy of the Flash memory pointer in EEPROM and RAM. At the end of every data record cycle, a new memory pointer is written to EEPROM in case a hard reset occurs. Otherwise, the pointer can be accessed in RAM. Therefore, upon entering the Flash initialization portion of the program, the Flash memory pointer is already easily accessible. Before proceeding to CAN initialization, the Flash memory pointer is checked in order to ensure that it is below the maximum Flash memory address. If this is found to be false, the Flash full flag is set and an error LED is lit.

Before initializing the CAN, a 1 second delay occurs in order to guarantee that every CAN module connected to the bus has initialized and is ready to receive CAN packets. At this point, the timing parameters for the AT90CAN64 CAN bus are setup along with a transmission register. The AT90CAN64 has 15 message objects which can be used for transmitting or receiving data on the CAN bus. Each message object can be setup uniquely for different CAN bus functionality. In the Onboard Weather Station, the first message object is used strictly for transmission to the connected modules. The remaining message objects are used to receive CAN bus replies.

In order to setup the receive message objects of the controller, it is necessary to know the addresses of all CAN bus modules connected to the system. Since hard coding addresses into the device would restrict the design to accepting communication from 14 addresses, a cycling procedure was created in order to isolate addresses with external modules attached. This procedure loops through all possible CAN addresses and waits 1ms for a response. If no response is received, it is known that no module exists at that address. The system completes this procedure until all modules connected to the bus are identified. It then compares this module count to the number of modules expected. If fewer modules are found than expected an error LED is lit.

At this point, the addresses of all modules connected to the CAN bus are known. Therefore, the system is able to setup the message objects of the CAN controller such that it only communicates with the modules initialized. After the modules connected via the CAN bus have been identified, the system also attempts to receive calibration data from each module. Calibration data parameters are unique to each data collection module and are needed to accurately translate the raw sensor data to actual values. Since a valid way to calibrate the system has not yet been identified, the calibration portion of the code is not yet complete.

After the CAN bus has been initialized, all error codes are displayed to the user for an additional period of time before being cleared. The system is then able to enter what is known as "Pit Road Mode", which is intended to be used when the car is being driven to the starting line of the track or when the car is being maintained. In pit road mode, UART receive interrupts are enabled in order to allow the system to quickly detect if a computer is connected. Furthermore, a feature called CANTalk is initialized. CANTalk is a communication protocol developed by the author in order to ensure that the CAN modules do not lose communication with the main module before the race begins. The final system initialized is the automatic trigger for data record. If an RPM input is selected as an automatic trigger in the system, the necessary timers must be started in order to monitor RPM.

At this point, a loop is entered in order to monitor several system conditions until a change is detected. The first check is to determine if a computer has been connected. If this is found to be true, the computer connected function is called. Next, the CANTalk time period is checked. If a certain time period has passed, a data frame is sent to all CAN modules in order to determine if they are still present on the bus. If the modules fail to reply within 1ms, an error counter is incremented for that module. Otherwise the error counter is cleared. If 3 errors occur for any module on the CAN bus, an LED is set to inform the user that one CAN module has lost communication.

After CANTalk is complete, the ignition voltage is checked. If the system has lost power, sleep mode is entered. In sleep mode all systems are disabled except for the ADC and RTC interrupt. The system wakes up for every RTC timer register overflow (once per second) in order to update the time registers. At this point, it checks to see if voltage has been returned to the system. If this is true then the system jumps to the software reset portion of the code (marked previously). Otherwise, it returns to sleep until the next timer interrupt. Despite the interrupts every second, putting the processor into power save mode significantly reduces power consumption (16µA vs. 29mA in normal operation) allowing the system batteries to last for longer periods of time.

If the voltage check shows that the vehicle is still running, a check of the memory full flags is completed. If the memory selected for recording data is full, the system is not allowed to enter data record mode. Therefore, the program will loop to the beginning of “Pit Road Mode”. Otherwise, the system proceeds to check the run triggers. Since several sources are able to trigger a run in the Onboard Weather Station system, the system must first distinguish which source to check. This is done by examining the run time parameters available in RAM.

The manual switch is checked by simply reviewing the status of the input pin to which it is connected. If the pin is found to be low, it is known that record mode should be entered. In order to account for debounce time, the switch must be closed for 1 second before the system will enter “Race Mode” where all data recording occurs. The two-step trigger is also checked by reviewing the state of an input-pin, since the two-step is similar to a switch. However, since the two-step is classified as a race-time or automatic trigger, the way in which it is handled during “Race Mode” is different than the manual switch. Furthermore, the two-step does not have a 1 second debounce.

RPM is also able to trigger the weather station by exceeding a certain threshold. Since RPM values are recorded as times in the weather station rather than actual RPMs, the threshold comparison must check to see if the recorded time is lower than the threshold time (smaller times correspond to higher frequency and therefore larger RPM). If this is true, “Race Mode” or data recording mode is entered.

When “Race Mode” is entered, the program begins by writing a start block to the selected memory device. This block includes a start string to denote a new sample cycle has begun, the start time (century, year, month, day, hours, minutes, and seconds), and calibration data for all modules connected to the CAN bus during this sampling cycle.

Before entering the sampling loop, all timer interrupts relevant to race data collection are started. Furthermore, the UART receive interrupts setup for “Pit Road Mode” are disabled, since a computer should not be able to interrupt the system while in race operation. The system then loops until the sample period has passed.

The first check performed after this period is to determine if the automatic trigger condition has been prematurely removed from the system. If this is true a special exit sequence is called. Otherwise, the system checks to see if the duration period for the automatic trigger has ended. If this is true, a flag is set to skip this check in subsequent sample periods. The automatic trigger duration is a run time parameter that can be set by the user. Data will only be recorded if the automatic trigger remains present for the specified duration.

At this point, an empty data frame is sent to all CAN modules in order to signal them to send sample data to the main module. While waiting for the data to return, the system samples the ADC in order to receive the x, y, and z-axis accelerometer data as well as the temperature and voltage values of the system. RPM values are also recorded for each RPM input. All data values are stored in a data buffer that is capable of storing 512 bytes.

Next the CAN message objects are checked in order to determine if data has been received. After the first 8 bytes are recorded for every module on the CAN bus, a second transmission is sent to trigger the send of the next 8 bytes. This is continued until all 32 bytes have been received from the external modules. Data is stored in the buffer in order of module.

After all data has been collected, a counter is reviewed in order to determine if 3 sample periods have passed since the last data store. If this is true, data is sent to the selected memory for writing. Otherwise, the XBee enabled flag is checked. The memory pointer is incremented and checked to ensure that the selected memory device is not full. If the device is full, data record mode is exited. When data is written to memory, the counter of sample periods is cleared.

If the XBee wireless transceiver is enabled and initialized properly, the system continues sample loop operation by performing wireless transfer of the data collected in the sample period. Otherwise, it ends the sample loop by checking to see if the race duration has ended (a parameter to exit the sample loop when data record is started by an automatic trigger) or if the manual switch is no longer closed (open condition must be present for a 1 second debounce period). Either condition has the potential to end the race depending on the type of record triggering enabled. A final condition that results in "Race Mode" exit is ignition system voltage dropping below 10 volts. If no exit condition occurs, the system returns to the beginning of the sampling loop.

Two exit modes exist for "Race Mode". The first was mentioned previously and occurs when the automatic trigger signal is lost before the duration is complete. If this is found, the system retracts the memory pointer such that it points to the memory location prior to the start of the data record session. Therefore, it is as if the session never occurred. The memory pointers stored on the SD card or EEPROM are updated appropriately.

In normal race exit, the system first checks to determine if there is buffered data that should be written to memory. If this is true, the data is written and memory pointers updated. All timers that are only required for sampling are disabled, as well as the RPM interrupts not needed for data record triggering. The record LED is disabled, and the system reenters "Pit Road Mode."

The other function of the main module that should be described is the computer interface code. Upon entering the computer connected function, the system disables UART receive interrupts and RPM interrupts. It enables the 0.01ms timer to be used for timeouts. At this point it sends the version number and version date of the embedded code to the weather software. In order to proceed, the weather software must send the device a character to signify the next operation the device should perform.

It should be noted that the computer connection code is designed such that a computer can send and receive data from the weather station through wireless communication or USB transfer. This is selected based on the method the computer uses to send the initial connection code to the weather station. Some commands cannot be completed if wireless is selected as the connection method. These commands all correspond to requesting specific data about the wireless module attached to the weather station. This will be elaborated on in the next few paragraphs.

The functions the weather station supports and that can be requested by the computer are as follows: SD card data send, Flash data send, SD card erase, Flash erase, SD card empty check, Flash empty check, timeout prevention send, real time clock setup, parameter setup, parameter send, system test, system calibration, module count send, XBee serial number send, and computer mode exit. If a timeout is exceeded or computer exit character received, computer mode is exited and the system returns to "Pit Road Mode."

The XBee serial number request cannot be performed if the method of communication is wireless due to the fact that the XBee device must enter control mode in order to access its serial number. The serial number of the XBee device is also its wireless address. Therefore, this number is needed in order for the wireless device controlled by the weather station software to communicate with the wireless module attached to the weather station.

The SD card and Flash reads are currently setup to send all data present on the device to the weather station software. It should be noted that the request will not be fulfilled if the memory device is empty. Furthermore, if the SD card is not present or not selected as the memory to be used in the circuit, an error code will be returned to the weather software.

The SD card and Flash erase functions erase all memory in the selected device. For an SD card erase, if the SD card is not present or not selected as the memory to be used in the circuit, an error code will be returned to the weather software. After memory erase, the memory pointers for the device erased are updated. If the selected device is the SD card, a new memory pointer is recorded in the first memory position of the card. If Flash was erased, the update occurs to the Flash pointer EEPROM variable.

The memory check requests are designed to look at the memory pointer for either the SD card or Flash. If the pointer references the first location of memory of the selected device, it is assumed that the device to be checked is empty. Otherwise, the device is assumed to have data. One of these conditions is returned to the weather station software. An error will be returned by the SD card memory check function, if the device is not present or not initialized.

The RTC setup function is designed to update all variables related to the real time clock with new values sent from the weather station software. Before updating RTC parameters, a check is performed to ensure that communication between the weather software and weather station has not timed out. Otherwise, a timeout character could be recorded in an RTC variable, which would cause the entire system to malfunction.

The set parameters function is designed to receive new run time parameters for the weather station. It is setup similar to the RTC function in that it waits for data from the weather station software, checks to ensure that it is not a timeout character, and then stores the received data in the proper locations. The main difference is that the set parameters function is not designed to update all parameters simultaneously. The weather station software must specify which parameter to update after the set parameter command has been sent. The parameters that can be updated include the sampling period, wireless transfer on/off, wireless destination address, record duration when using automatic triggering, start trigger type, method of automatic triggering, threshold for RPM trigger, duration of automatic trigger, and number of CAN modules to expect on the bus. The get parameters function is identical to set parameters; however, instead of waiting for data to update the system variables, the device returns the current setup of the system.

The module count send function is setup to inform the weather software of the actual number of external modules connected to the main module through the CAN bus. It should be noted that this is different than the expected number of modules, which is a system parameter that can be set.

The final function implemented in this revision of the embedded code is the test function. The calibration function has not yet been created due to uncertainty as to the best way to calibrate the external data collection modules. Once entered, the test function waits for an external module to be specified to test. The software guarantees that the module requested is actually available on the CAN bus. At this point, it proceeds to send all sensor data (excluding RPMs) from the main module as well as all sensor data from the selected external module to the software.

For further questions regarding the embedded code please reference Appendix D.

5.1.8 Printed Circuit Board Design – Device Enclosure

Based on the circuits described in the previous sections, a printed circuit board was created for the main module of the weather station. This board has a width of 6 inches and a height of 4 inches as specified in the design requirements. The printed circuit board was designed well; thus, few modifications are required for the final revision. The changes that are required include increasing the size of the footprint of 2 components as well as reversing the direction of a further 2 components. Finally, several connectors need to be re-arranged on the PC board in order to allow them to be more easily accessed by the customer or more easily fit into the case chosen. The following figures show the PC board of the main module as well as the board within the chosen case.

Figure 21: Main Module PC Board


Figure 22: Main Module Extruded Aluminum Case

5.2 External Module Design

A detailed block diagram of the external module can be seen in Figure 23. As with the main module, the design will be described by subsection of the external weather module. Since the signal conditioning module denoted in the diagram influences the operation of the humidity sensor, it will be described in the sensor module section. Furthermore, the data input module (CAN bus) will not be described as the pin-out for the CAN connector has been detailed in Section 5.1.2.

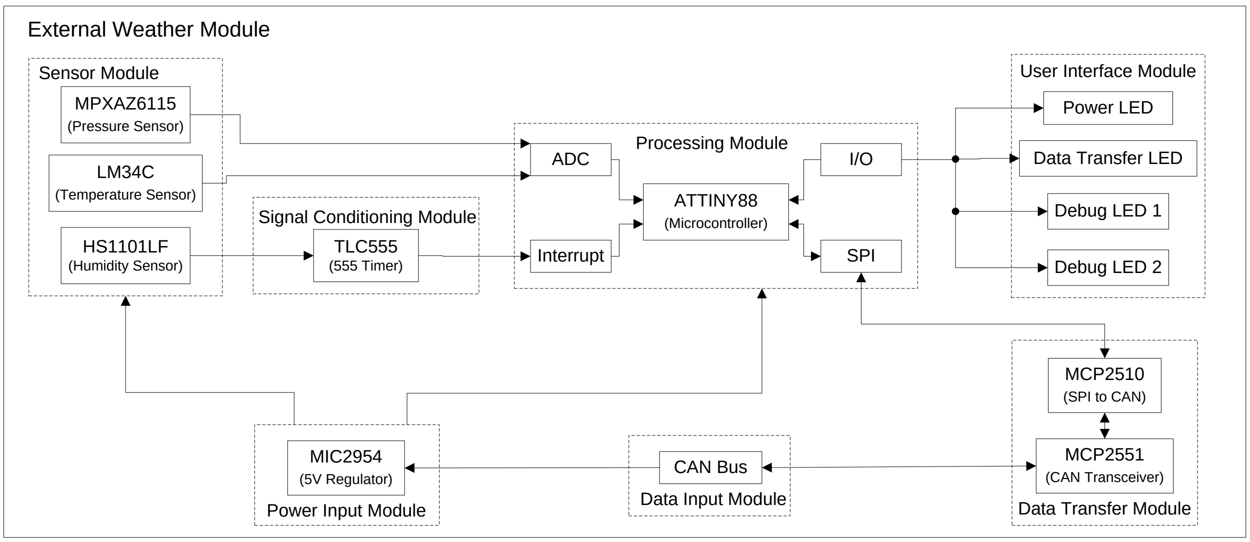


Figure 23: Detailed External Module Block Diagram

5.2.1 Power Input Module

The power input circuit of the external module is responsible for providing power to all subsystems of the external module. Since all components within the external module are able to operate off 5 volts, a single regulator was selected for supplying power. The other main components of the module include a RXEF025 PTC resettable fuse and a SMDJ30A TVS diode. Figure 24 is a representation of the power supply circuit.

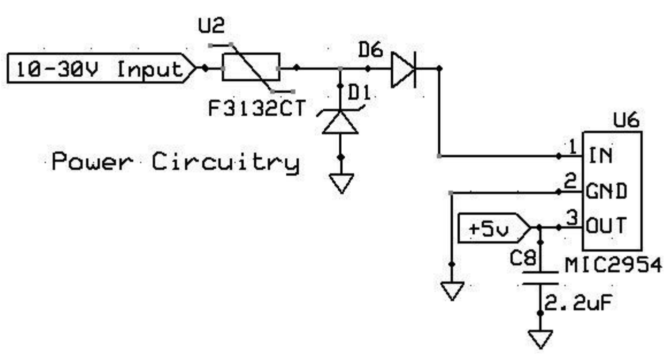


Figure 24: External Module Power Input Circuit

The input to the power supply circuit is a 10-30 volt signal that is provided to the module through the power lines of the CAN bus. The power received is unregulated, making transient suppression and voltage regulation necessary. As in the main module, current consumption calculations were performed for all the main devices of the external module in order to select proper power supply components. The current consumption results can be viewed in Table 5.

Component	Quantity Running Concurrently	Single Component Current (mA)	Total Current (mA)
HS1101LF (Hum. Sensor)	1	0.6	0.6
LM34CAZ (Temp. Sensor)	1	0.142	0.142
MPXAZ6115A (Bar. Press. Sensor)	1	10	10
MPXAZ6115AC (Wind Press. Sensor)	3	10	30
LEDs	2	10	20
MCP2551 (CAN Transceiver)	1	75	75
MCP2510 (CAN Controller)	1	10	10
ATTINY88 (Microcontroller)	1	8	8
		Total:	**153.742**
		Buffered Total:	**230.613**

Table 5: External Module Current Consumption

From the current calculation, it was determined that the transient suppression circuit should be able to support 160mA of current in normal operation; however, it should not trip unless current draw exceeds 235mA. The component chosen for current suppression, RXEF025, has a hold current of 250mA and a trip current of 500mA. Furthermore, it was found to have a maximum response time of 2.5s. As in the main module, the remaining circuitry was designed to withstand the potential lag of this component. The TVS diode chosen, SMDJ30A, is the same as used in the main module; therefore, it is able to effectively clamp voltage transients to 33.3 volts.

After passing through the transient suppression circuit, power is conditioned by the one voltage regulator on the board. This component is a MIC2954, 5 volt regulator, which is also used in the main module power supply. As mentioned previously, this device is a precision voltage regulator. It has sufficient accuracy at the current draw required for circuit operation; therefore, an additional voltage reference is not needed.

5.2.2 Sensor Module

The sensor subsystem of the external module is composed of a temperature sensor, humidity sensor, barometric pressure sensor, and 3 wind pressure sensors. These components were all specified as required in Section 3.

The humidity sensor chosen is a HS1101LF variable capacitor manufactured by Measurement Specialties or Humirel. The sensor has an accuracy of ±2% and a sensing range of

1 to 99%. The device operates by changing its capacitance as the humidity in the air surrounding the sensor varies. In order to monitor the changing capacitance values, a variable capacitor measurement circuit was created (Figure 25). The datasheet for the HS1101LF suggested such a circuit using a 555 timer to create a pulse signal whose frequency would change with the capacitance of the device. Therefore, by recording the frequency of the pulses, the humidity experienced by the sensor could be calculated.

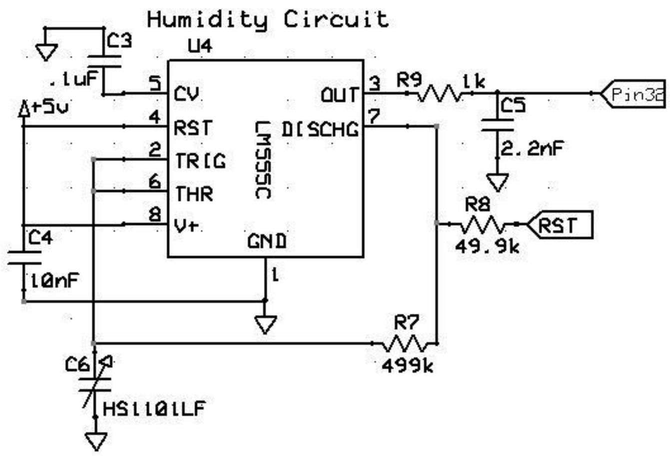


Figure 25: External Module Humidity Circuit

This circuit was found to be very sensitive to the values of R7 and R8 seen in Figure 25. For this reason, resistors with 0.1% tolerance were chosen. Even with this additional precaution, adjustments are needed in order to obtain an accurate capacitance reading. Despite this fact, after the system is tweaked, the capacitance values are very accurate over a broad range of humidity; therefore, it is unlikely that the sensor will be switched in the future.

In order to measure the humidity changes with the desired precision, it was necessary to create a timer that would interrupt every 0.00001 seconds to increment a counter, which was then used to measure the period of the humidity sensor pulse train. The system would time 100 humidity sensor pulses to further increase accuracy. This level of precision was necessary due to the fact that frequency of the humidity circuit could vary from 6200 to 7200 Hz. This corresponds to a pulse width between 0.000138 and 0.000161s. As can be seen, precision any lower than that used for the circuit would result in no distinction between humidity values.

The pressure sensor circuit can be seen in Figure 26. It was designed to handle up to 4 pressure sensors: 1 barometric pressure sensor and 3 wind pressure sensors. The sensors chosen to compose this module include 1 MPXAZ6115A barometric pressure sensor and 3 MPXAZ6115AC wind pressure sensors all manufactured by Freescale. The two components are of the same product line with the main difference between them being the case. The MPXAZ6115A is an open sensor allowing it to sense barometric pressure while MPXAZ6115AC is a closed sensor with a directed tube for measurement. A pitot tube can be attached to the directed output of the MPXAZ6115AC and placed at different points of the car in order to measure wind pressure.

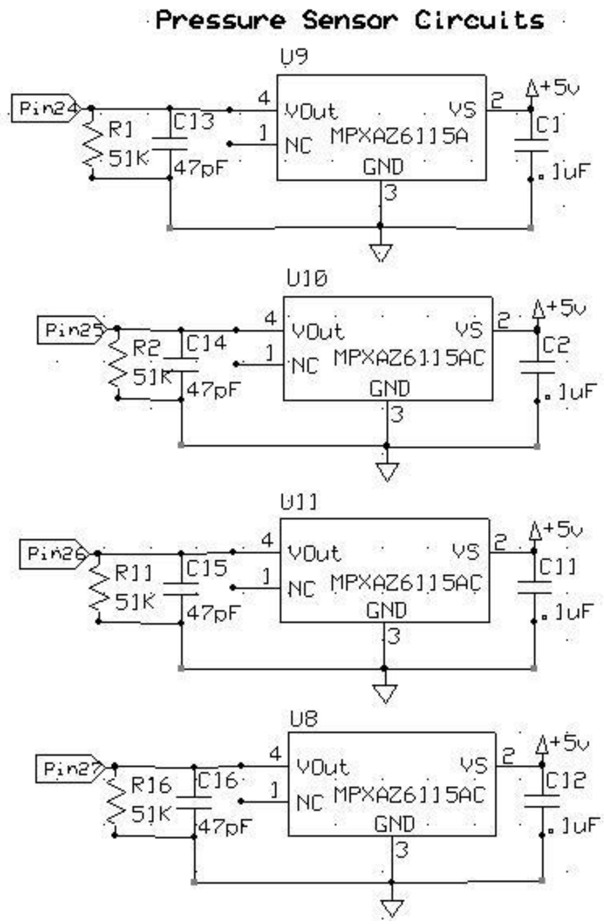


Figure 26: External Module Pressure Circuit

The MPXAZ6115A components are analog sensors that require little additional circuitry for correct operation. That which is seen in Figure 26 was suggested by the Freescale datasheet for an application circuit. The device provides an output signal from 0.2 to 4.7 volts; therefore, it can be fed directly into the microcontroller analog to digital converter without signal conditioning. The device produces a 45.9mV change for every change in kPa where 1kPa is equivalent to 0.2953inHg. The ADC of the microcontroller is capable of detecting changes of 4.88mV as can be calculated by inserting 1 into the variable ADC Value in Equation 1 and solving for V_{in}.

$$ADC\ Value = \frac{V_{in}(1024)}{5}$$

Equation 1: ADC Precision Equation

Therefore, the microcontroller is capable of detecting changes of about 0.0314inHg. Although this sensor line is convenient for operation and easy to use, a more precise sensor may be chosen in the future. The project sponsor is currently deciding whether this is necessary.

The final circuit of this subsystem is for sensing temperature. It can be seen in Figure 27. An LM34CAZ manufactured by National Semiconductor was chosen for this circuit due to its 1°F accuracy and simple setup. This device is also used for temperature sensing in the sensor subsystem of the main module; therefore, for more detailed information reference Section 5.1.2. It should be noted that the same measurement precision is achieved in the external module as in the main module, since the same reference voltage is used for analog to digital conversions.

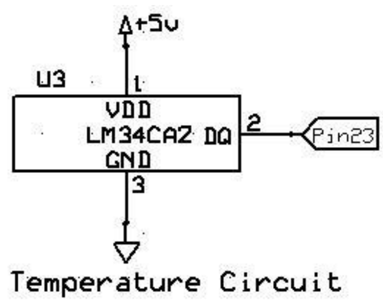


Figure 27: External Module Temperature Circuit

5.2.3 Data Transfer Module

The data transfer subsystem is responsible for establishing communication with the main module of the weather station. Since the method of connection between weather station modules is a CAN bus and the external modules do not contain CAN microcontrollers, a CAN controller component is necessary for correct operation. The CAN controller chosen is a MCP2510 manufactured by Microchip. The device is controlled by its serial peripheral interface (SPI) with the microcontroller, which it uses to receive all commands for operation. The circuit for the data transfer module can be seen in Figure 28.

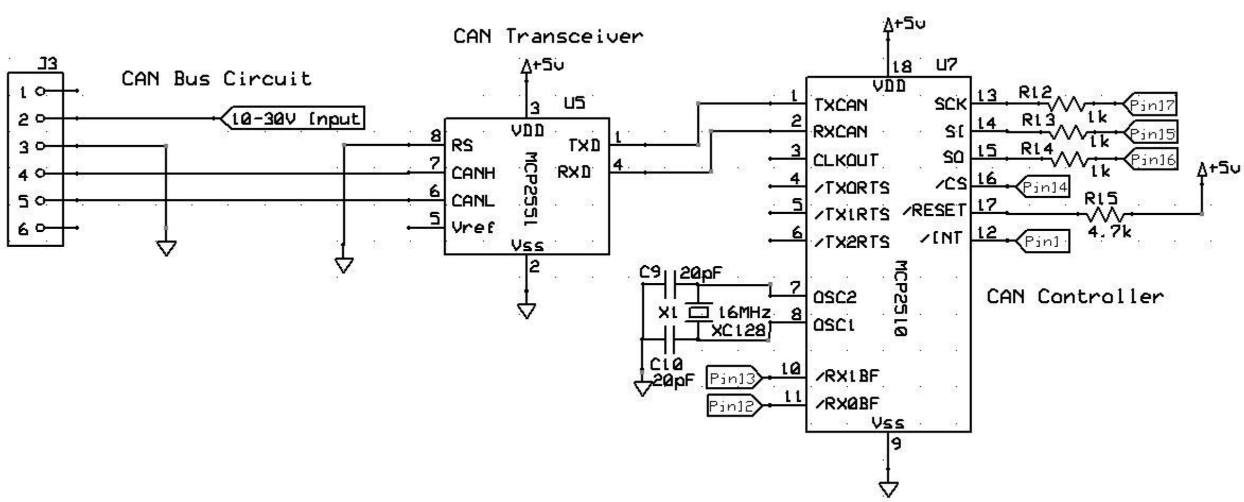


Figure 28: External Module CAN Circuit

Since the SPI connection is also required for programming the microcontroller of the external module, 1kΩ resistors are required to prevent contention. This was described previously in the main module design; therefore, details will be omitted. The component is setup as a SPI slave device allowing the microcontroller to control its operation. The MCP2510 is self-timed; therefore, limited microcontroller attention is required for device operation. Furthermore, the device provides interrupt signals on key events, which can be used to signal the microcontroller without using the SPI interface. A receive interrupt is used in the weather station system in order to rapidly inform the microcontroller that CAN data has arrived.

An additional feature of the circuit is the 16MHz input crystal to the MCP2510 controller. This component is identical to that chosen for the main module master clock. It is necessary in order to perform accurate bit timing for the CAN bus. As in the main module, a MCP2551 CAN transceiver is necessary for preparing signals to enter and exit the CAN bus. The RS line grounded on the transceiver is done in order to allow fast switching times for maximum bus speed. This configuration was chosen based on information from the CAN transceiver datasheet.

5.2.4 User Interface Module

The user interface for the external weather collection module consists of 2 LEDs: 1 red and 1 green. As in the main module, each LED circuit is designed to draw 10mA of current when on. Furthermore, the LEDs were made to be active low thus allowing a low signal to activate and a high signal to deactivate the components. Figure 29 shows the design.

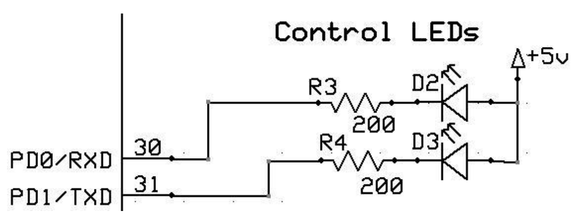


Figure 29: External Module LED Circuit

Prior to developing this circuit, research was conducted to ensure that the microcontroller output pins chosen would be able to properly sink the current to light the LEDs. The datasheet confirmed that 10mA would not damage the microcontroller pins; thereby, making the design viable.

5.2.5 Processing Module – Embedded Code

The final subsystem of the external data collection unit is the processing module. The main component in this module is the ATTINY88 microcontroller manufactured by Atmel. The ATTINY88 is an 8-bit microcontroller with 8KB program memory, 64B EEPROM, and 512B RAM. The device is able to operate at clock frequencies up to 12MHz; however, 8MHz operation was deemed sufficient for this application. The device supports SPI, which is the only communication method needed to interface with the other subsystems. Furthermore, it has external interrupts, digital input/output pins, and an analog to digital conversion peripheral making it ideal for external weather module operation. A pin-out of the ATTINY88 including the pins used for the external weather module can be seen in Figure 30.

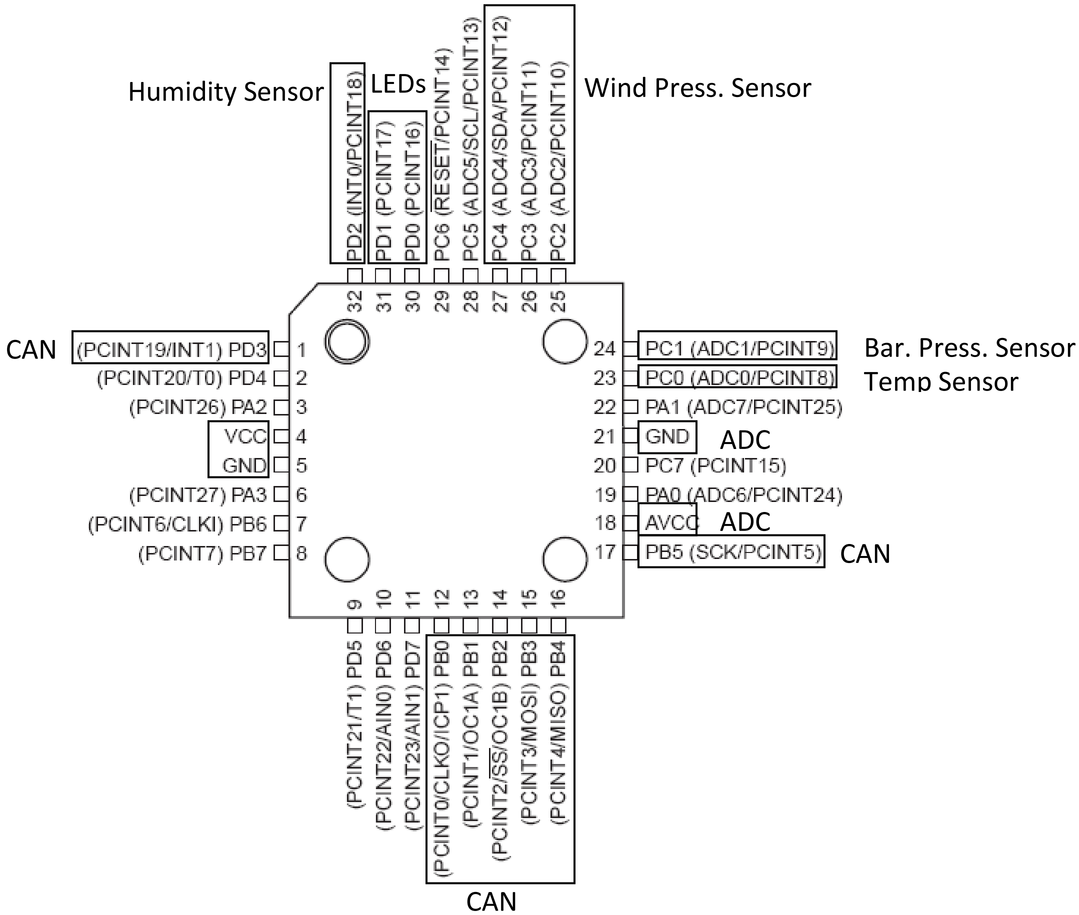


Figure 30: ATTINY88 Pin-Out

At the current time, the internal oscillator of the ATTINY88 is used as the master clock of the system. In the future, this may be changed to a more accurate external crystal. Atmel internal oscillators are known for their inaccuracies; therefore, in critical timing applications an external crystal is better suited. Since the humidity sensor requires precise timing for accurate measurements, an external crystal may be more appropriate for the system operation. A 12 MHz oscillator is unnecessary as the external module operation is not timing critical; therefore, maximum processing speed is not necessary.

The ATTINY88 is programmed using ISP through the SPI interface on the processor. Although the ISP connection allows a program to be downloaded to the processor, debugging is not possible. For this reason, debugwire must be used. Debugwire is a single line debug tool that can be activated by enabling a fuse bit within the ATTINY88 microcontroller. Once debugwire is enabled, the ISP connection no longer works and a special debugwire programming tool (operated through a JTAG debug module) is needed. Once debugging is complete, the JTAG-debugwire tool can be used to re-enable the ISP interface by disabling the debugwire fuse.

It should be noted to those wishing to try this that the JTAG-debugwire tool is absolutely necessary for this to work properly. Once the debugwire fuse is enabled, the only ways to disable it are through the JTAG-debugwire tool or high voltage programming. The remaining fuses of the microcontroller cannot be modified while the device is setup to use debugwire. For more information, the Atmel datasheet should be referenced.

As in the main module, a low-pass filter is used to reduce noise on the analog to digital converter power supply lines of the microcontroller. The filter parameters specified in the ATTINY88 datasheet are slightly different than those of the AT90CAN64. A 10mH inductor and 0.1µF capacitor are suggested. Otherwise, the filter setup is identical.

The embedded code for the processing module was completed in Atmel Assembly as specified in the project requirements. Three main files compose the assembly program: Interrupts.asm, Common.asm, and ExternalWeatherModule.asm. Interrupts.asm contains all interrupt subroutines for the system. Common.asm contains subroutines commonly used by the system, and ExternalWeatherModule.asm contains the code to run the general operation of the external modules. A flow chart for ExternalWeatherModule.asm can be seen in Figure 31. Only the main program will be described as this is the unique portion that defines the system design.

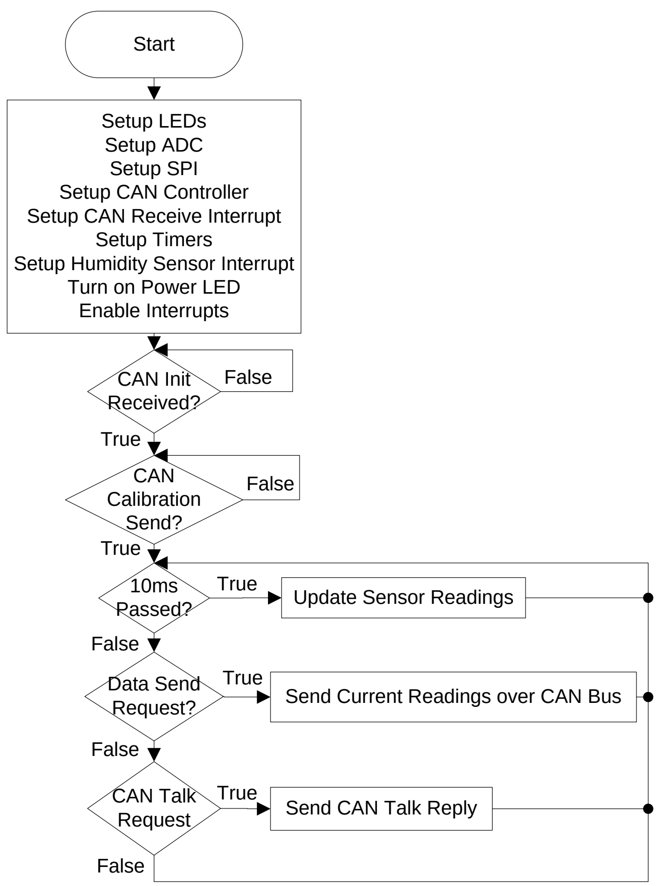


Figure 31: External Weather Module Flowchart

The program begins by calibrating the system clock in order to achieve a frequency close to 8MHz. This is done by writing to the OSCCAL register with a calibration value specified by the Atmel datasheet. At this point the ADC, LEDs, and SPI connection are setup to properly operate in the system. The SPI frequency is set to 4MHz as this is the maximum achievable frequency with a master clock operating at 8MHz. An external interrupt is also initialized and enabled in order to allow the CAN controller to directly signal the microcontroller on CAN receives. The CAN controller is connected through SPI for general operation. As part of the startup sequence it is reset and initialized to operate at a bit rate of 1Mbps.

A 0.01ms and a 10ms timer are setup for system operation as well. The 10ms timer is used to trigger sensor sampling while the 0.01ms timer is used to time the pulse frequency of the humidity sensor. The external interrupt for the humidity sensor is also initialized at this point. The final function of the initialization code is to light a power LED and enable global interrupts.

After initialization, the system waits for a CAN initialization signal to be received over the CAN bus. When this occurs, the device replies to alert the main module that it is present. The device must respond to every initialization signal received over the CAN bus (several may occur due to the looping search for CAN devices that the main module implements). After initialization on the CAN bus, the device must wait and reply to a calibration signal from the main module. This will only be received once, since the main module has already identified connected CAN modules and will be able to address all of them with one transfer.

In the final system, calibration data will be returned to the main module in response to the calibration signal. However, the best way to implement this functionality has not yet been decided by the sponsor; therefore, it is left as future work. After CAN initialization and calibration is complete, the red LED is lit in order to inform the user that the device has been initialized on the CAN bus.

At this point the system enters the main loop of the system. The loop begins by checking to see if 10ms has passed. If this is true, all sensor variables are updated with current sensor measurements. Otherwise, a check is conducted in order to determine if a receive interrupt has occurred from the CAN controller. If this is true, the received data is extracted from the CAN controller through the SPI connection, and the CAN receive register is reset. If the message signifies a data send, the sensor variables are packaged into 8-byte packets and sent to the main module. The system will wait for 3 additional data send transmissions before returning to the main system loop. This is done in order to give full attention to main module requests during its sampling period.

If a CANTalk message is received, the device must reply with a single CAN data frame to inform the main module that it is still present. In the future, the device will also be setup to handle a calibration request over the CAN bus. This will cause the system to go into a calibration mode where various sensor measurements will be made and returned to the main module for determining calibration parameters.

For additional information regarding the embedded code of the external module please reference Appendix D.

5.2.6 Printed Circuit Board Design – Device Enclosure

A printed circuit board was also designed for the external weather modules of the system. It can be seen in Figure 32. The board for this circuit has a width of 4 inches and a height of 2 ½ inches as specified in the design requirements. At the current time, it appears that the printed circuit board designed will suit the needs of the final system; therefore, major changes are not necessary. The two modifications that most likely will be made to the board before final production include the addition of an external oscillator to improve system timer accuracy and the elimination of the two debug LEDs, which will not be needed once debugging is complete. The board size may also be reduced (if possible) in the final revision before production.

Figure 32: External Module PC Board

A case has not yet been chosen for the external weather modules due to the limitations that have been imposed. The case must not obstruct air flow while still providing protection to the circuitry. Furthermore, the project sponsor has expressed a desire for extruded aluminum casing. Mounting considerations must also be made as the external weather module is to be placed in front of the air intake of an engine. Therefore, a final case design that meets all mentioned specifications remains to be found in the future.

Figure 33 shows the weather station in operation. As can be seen, the two external modules are connected to the main module through different length CAN bus connections. The left bus connection has a length of 2ft while the right bus has a length of 20ft. The external modules have both powered-on and initialized through the CAN bus. The main module has powered-on without any errors.

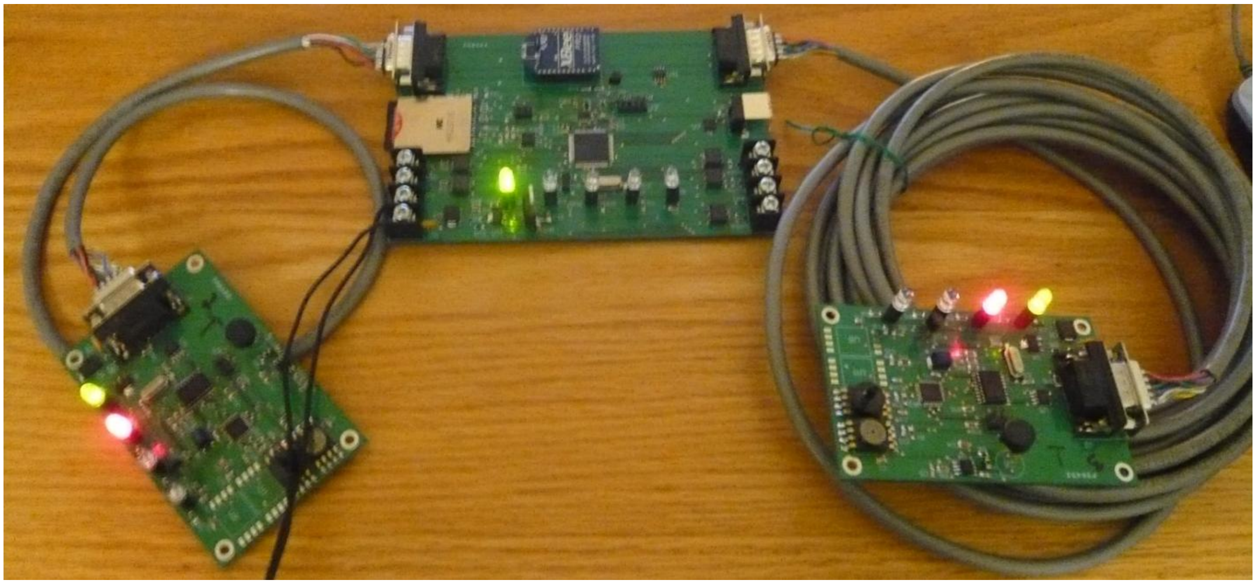


Figure 33: Complete Weather Station in Operation

5.3 Software Design

The final aspect of design is the weather station software. As specified in the project requirements, the software was written using the Delphi design environment. The software design will be described briefly as it is not the focus of this project.

Upon starting the software, a screen opens that allows the user to connect to the Onboard Weather Station (Figure 34). If the Onboard Weather Station is connected to the computer when the software starts, communication between the weather station and weather software will automatically be established. When connected, the version number and date of the weather station firmware will be displayed on the screen as well as a green connection panel.

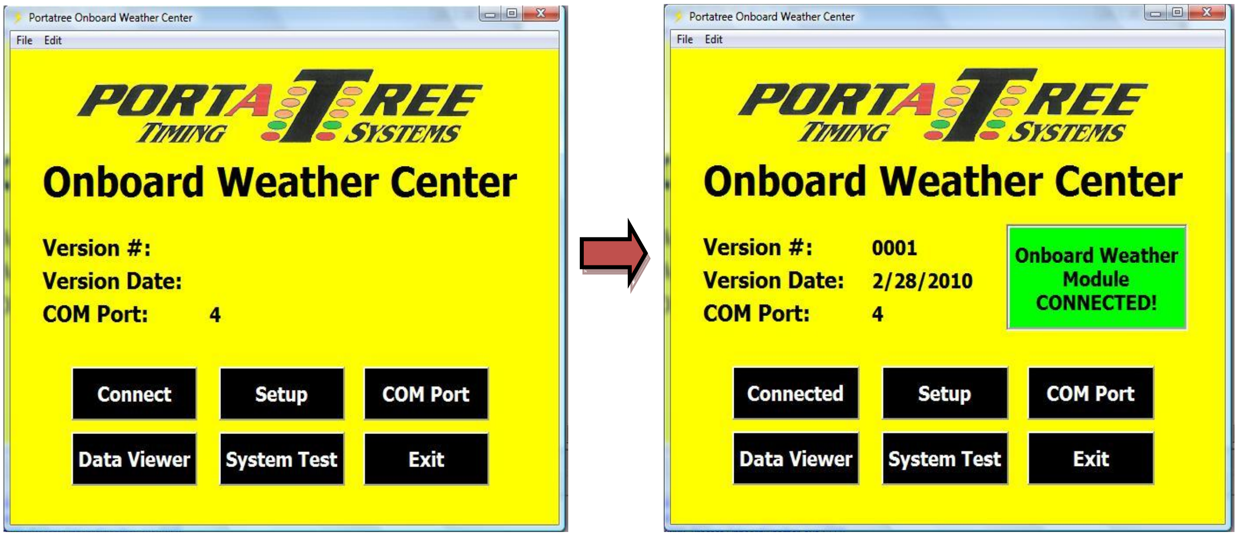


Figure 34: Weather Software Start Screen

Although the different screens of the weather software can be entered while the weather station is not connected, the greatest software functionality is achieved once communication with the device is established.

The Onboard Weather Station setup screen (Figure 35) is designed to allow the user to set all parameters in the system. This screen can also be used to verify the parameters currently set in the weather station. If the Onboard Weather Station is not connected and the user attempts to set or get parameters, a warning message will be displayed to inform the user that the requested operation cannot be performed.

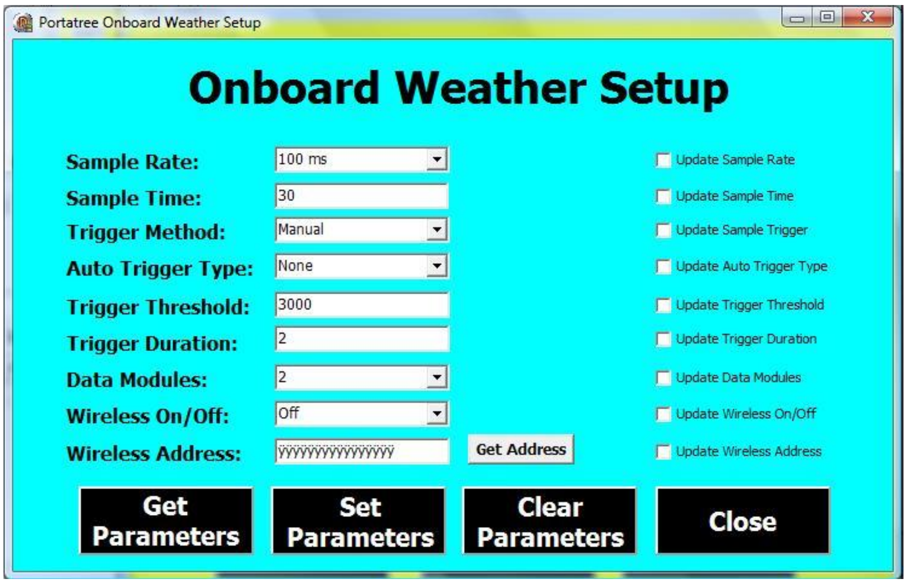


Figure 35: Weather Software Setup Screen

The weather center test screen is currently designed to monitor the values collected by the sensors from the main module as well as the sensors on a specified external module. The external module to test can be selected through an edit box on the right side of the screen. A single test request can be sent to the weather station or a periodic request. The periodic request occurs once per time period specified by the user in the second edit box on the right side of the test screen.

In the future, the test center may also provide calibration functionality. This decision has not been finalized at the current time as it may be decided that device calibration should only be performed at the factory. If this calibration method is chosen, separate calibration software will be created and the test center will be left with its current functionality.

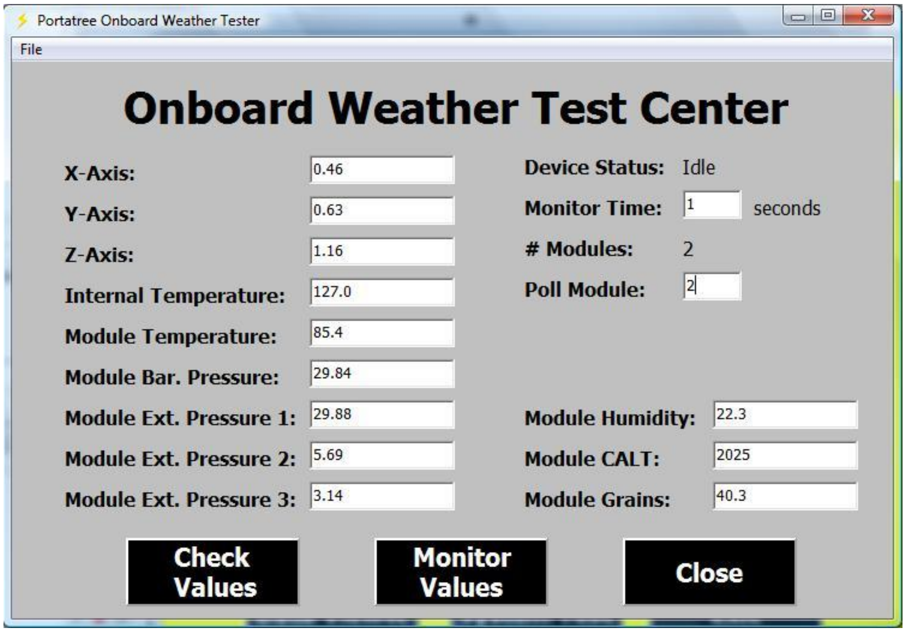
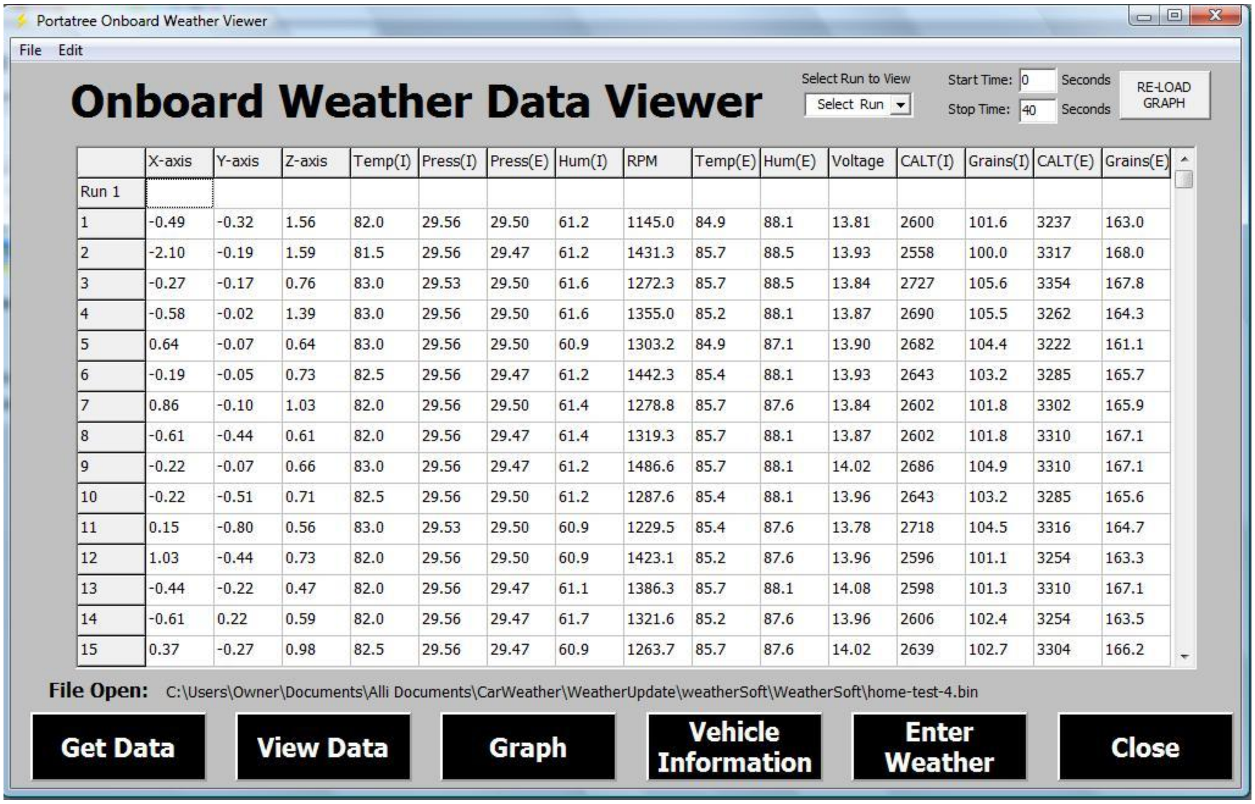


Figure 36: Weather Software Test Screen

The final screen of the Onboard Weather Station is the data viewer. This screen has the greatest functionality and is also farthest from completion. The data viewer is able to extract data from the weather station as well as erase both memory devices on the weather station board. All data extracted is stored in a data file that can then be opened and viewed in a table. Raw data is collected by the Onboard Weather Station; therefore, conversion to final values must occur before the data can be displayed. In the future, the data viewer must be setup to calculate horsepower correction factors as well as predict vehicle performance from the weather data gathered. A screenshot of the data viewer with an open file can be seen in Figure 37.

The data viewer screen also provides the user with the ability to graph the data received from the weather station. Data trends can more easily be recognized from graphing allowing the user to better predict the performance of his or her vehicle. Vehicle information must be input to the data viewer for proper calculation of RPM. Stroke type and cylinder count are necessary parameters in the RPM computation. Furthermore, static weather readings from a traditional weather station can be entered in the data viewer. This allows the user to compare the conditions on the track to conditions recorded at the trailer.

In the future, the data viewer must be setup to handle the modular structure of the weather station. It is currently designed to support a very strict data format, which was created for testing an initial weather station prototype. The data viewer must also be modified to store data as it is received from a race vehicle moving down the track. This function must be implemented carefully as data cannot be viewable until a race is complete. Therefore, a delay of 15-20 seconds may be implemented after the last packet is received before allowing the data file to be accessed.



	X-axis	Y-axis	Z-axis	Temp(I)	Press(I)	Press(E)	Hum(I)	RPM	Temp(E)	Hum(E)	Voltage	CALT(I)	Grains(I)	CALT(E)	Grains(E)
Run 1															
1	-0.49	-0.32	1.56	82.0	29.56	29.50	61.2	1145.0	84.9	88.1	13.81	2600	101.6	3237	163.0
2	-2.10	-0.19	1.59	81.5	29.56	29.47	61.2	1431.3	85.7	88.5	13.93	2558	100.0	3317	168.0
3	-0.27	-0.17	0.76	83.0	29.53	29.50	61.6	1272.3	85.7	88.5	13.84	2727	105.6	3354	167.8
4	-0.58	-0.02	1.39	83.0	29.56	29.50	61.6	1355.0	85.2	88.1	13.87	2690	105.5	3262	164.3
5	0.64	-0.07	0.64	83.0	29.56	29.50	60.9	1303.2	84.9	87.1	13.90	2682	104.4	3222	161.1
6	-0.19	-0.05	0.73	82.5	29.56	29.47	61.2	1442.3	85.4	88.1	13.93	2643	103.2	3285	165.7
7	0.86	-0.10	1.03	82.0	29.56	29.50	61.4	1278.8	85.7	87.6	13.84	2602	101.8	3302	165.9
8	-0.61	-0.44	0.61	82.0	29.56	29.47	61.4	1319.3	85.7	88.1	13.87	2602	101.8	3310	167.1
9	-0.22	-0.07	0.66	83.0	29.56	29.47	61.2	1486.6	85.7	88.1	14.02	2686	104.9	3310	167.1
10	-0.22	-0.51	0.71	82.5	29.56	29.50	61.2	1287.6	85.4	88.1	13.96	2643	103.2	3285	165.6
11	0.15	-0.80	0.56	83.0	29.53	29.50	60.9	1229.5	85.4	87.6	13.78	2718	104.5	3316	164.7
12	1.03	-0.44	0.73	82.0	29.56	29.50	60.9	1423.1	85.2	87.6	13.96	2596	101.1	3254	163.3
13	-0.44	-0.22	0.47	82.0	29.56	29.47	61.1	1386.3	85.7	88.1	14.08	2598	101.3	3310	167.1
14	-0.61	0.22	0.59	82.0	29.56	29.47	61.7	1321.6	85.2	87.6	13.96	2606	102.4	3254	163.5
15	0.37	-0.27	0.98	82.5	29.56	29.47	60.9	1263.7	85.7	87.6	14.02	2639	102.7	3304	166.2

Figure 37: Weather Software Data Viewer

6 Results

After the weather station was designed and constructed, testing began to confirm the proper operation of the system. Testing was conducted in two stages. First the custom hardware was tested through a series of embedded test programs. Once everything was found to be functioning properly, the weather station embedded code was programmed into the modules and tested. This section will be broken down to describe the hardware tests, embedded code tests, software tests, and complete system tests. In each section, the system requirements that were verified by the conducted tests will be mentioned.

6.1 Hardware Verification

Hardware verification was conducted in order to test the main circuits of each of the modules. For the main weather module, tests were focused on the RPM, two-step input, manual switch input, USB to PC interface, Flash, SD card, temperature sensor, accelerometer, LED, XBee-PRO, and CAN bus circuits. For the external weather collection units, tests were focused on the pressure sensor, temperature sensor, humidity sensor, LED, and CAN bus circuits. The CAN bus was tested last due to its complexity. Both the main weather module and external weather module must be setup properly for a successful CAN bus transfer to be achieved.

Prior to beginning verification, several design requirements relating to hardware were already met due to design choices made in constructing the weather station. One requirement met by design is that all system components are standard production line parts with direct drop-in replacements. The ability to be powered by batteries or a 10-30 volt input signal is also part of the design; however, it was additionally verified through using the voltage supply of a protoboard to input power to the circuit in order to ensure that the board would continue to run through the entire voltage range. Although the device does not meet the battery power requirement, amendments to the power input circuit have been made for the final system. Therefore, this requirement will be met on the next board revision. The board size requirements were met through designing the printed circuit boards to conform to the specified criteria. The board size was further verified by measuring the boards upon their arrival from the manufacturer.

The following sections will describe the tests conducted to further verify the hardware of each module of the system.

6.1.1 Main Module Hardware Tests

The first hardware test conducted was to verify XBee data transfer. One XBee device was mounted on a development board (available through Digi International) and connected to a PC using the board's USB interface. The Digi X-CTU software was then started and allowed to connect to the XBee chip. After connection, a hyper-terminal style interface within the X-CTU software was opened where characters to send between XBee modules can be entered and received data is displayed. By using the X-CTU software, one side of communication was guaranteed to work properly; thereby, reducing the possible causes of problems in the system.

The second XBee module was placed on the main weather station board, which was developed in this project. The main module was programmed to perform a simple operation. Upon receiving an ASCII character corresponding to a number (1 – 5), an LED synchronized with that number would light. If an unrecognized character is received, the LEDs are cleared. Therefore, by entering numbers into the hyper-terminal screen of the X-CTU software, LEDs are lit on the main module board. This function was found to work flawlessly. Furthermore, through this test both the board LEDs and XBee receive functions were tested.

Next, XBee transmit from the main module was tested. This was done by modifying the original program to echo the value received. Therefore, if a ‘1’ is received by the main module, a ‘1’ is sent as a reply. Values were once again entered into the X-CTU hyper-terminal screen and responses were reviewed. The transmit function was also found to work properly.

The final test conducted with the XBee modules was to verify that the control parameters of the XBee device could be changed. This is done by sending control sequences to the XBee module and observing the response. Problems occurred when first trying to modify control parameters due to a misunderstanding of the XBee datasheet. XBee control commands are a sequence of ASCII characters that specify a parameter to update followed by the new value. In the datasheet, it was not clear how the new value should be sent to the XBee device. Most parameters were numeric; however, when sending the desired parameter value to the XBee module, invalid response tokens were received. It was soon realized that the XBee device expected ASCII representations of numbers in order to properly accept the command. Once this was discovered, the XBee setup function was modified to work properly. At this point, it was determined that the XBee wireless transfer circuitry was completely functional.

Tests were conducted with the XBee devices restricting them to only communicate with modules of certain addresses. The wireless addressing was found to work properly; thereby meeting another design specification. A wireless range test was also conducted with the XBee modules. For line of sight testing, the range was verified to be approximately 1 mile as expected from the XBee datasheet. Range tests were conducted with obstacles including trees and hills between the XBee modules. It was found that 3/10 of a mile range could still be achieved. In motorsports, it is very unlikely that large obstacles will exist between the two XBee transceivers. Therefore, it was verified that wireless transfer could occur up to ½ mile in conditions expected at race tracks.

Next the main module sensors were tested in order to ensure proper operation. In order to do this, a sample loop was created that would collect sensor readings every 20ms. By running the system on the debugger and placing a breakpoint at the end of the sample loop, the sample readings taken during the loop could be checked. From the raw data, actual sensor values were calculated in order to determine if the sensor responses were reasonable. For the g-force sensor, the board was tilted on every axis and values monitored. From this test, the 3-axis accelerometer was verified to operate properly. Since by design it is setup to collect forces up to ±12g on all 3-axes, the system design requirement to collect 3-axis g-force data up to ±7g was met.

The ignition system voltage sensing circuit was also verified by the above test; however, the temperature sensor circuit was not. Since the JTAG debugger was needed to conduct this test, the system could not be setup to perform temperature sensor readings as the temperature sensor and JTAG debugger share a pin. It was decided that since the same temperature sensor circuit and ADC configuration are used for the external module, by verifying the temperature sensor operation on the external module, it would also be verified in the main module. For this reason, temperature verification was omitted from the testing process of the main module.

The most difficult circuit to test was the RPM input to the system. Prior to beginning tests of the RPM circuit, the author was fairly confident that it would function properly as she had previously designed and used the circuit for RPM monitoring. The circuit had already been modified to eliminate design quirks; therefore, it was expected to be a drop-in solution to the system. When tests were conducted (using a function generator to simulate the RPM pulse train), it was found that the RPM recorded by the microcontroller was always 0. Initial checks showed that the program was entering the RPM interrupts properly.

It was eventually noticed that the program would enter the RPM interrupt twice for every pulse received. The second entry would occur immediately after the first; thus, a time difference of 0 would be seen between the first entry and second entry resulting in the 0 RPM value. The problem was finally isolated to the interrupt flags failing to be cleared. Although the Atmel datasheet specifies that the flags are cleared when the interrupt is handled, the system operation observed did not match this functionality. By clearing the interrupt flags when inside the interrupt, the system worked as intended. The RPM values were found to be fairly accurate. From this test, it was verified that the system could measure 4 RPM inputs.

The two-step input and manual switch input circuits were tested by having them light an LED when in the "closed" position and clear an LED when "open." Both switch inputs were found to work as intended. The real time clock was setup and tested by comparing the microcontroller RTC timing to that of a computer system clock. A break point was placed at the one minute marker of the RTC interrupt, and a PC system clock was run. When the PC system clock reached the 12 location, the run button was pressed in order to start the debug environment. When one minute passed for the microcontroller RTC, the breakpoint would be reached and the system would stop. At this point, it would be compared to the computer system clock for timing verification. The system was demonstrated to be accurate within 1 second. Testing was also conducted over long periods of time. In one case, the system was left to run and then checked after 1 hour. The microcontroller RTC always matched the system clock of the computer for long tests resulting in further verification of the RTC circuit.

The USB circuit was tested in a similar fashion to that of the XBees. Embedded code was written that would light LEDs when numbers were sent over the USB connection. The characters received are also echoed to ensure both transmit and receive operations function properly. Hyper-terminal was started on the host PC and setup to open the port with the main module attached. Since a Type-B USB connector is used on the main module and a Type-A connector is present on all computers, a standard Type-A to Type-B cable could be used to connect the main module to the PC. By typing characters into hyper-terminal, responses were received from the main module. This test verified that the system was able to connect to a PC through a Type-A to Type-B USB cable.

Finally Flash and SD card read and write operations were tested. The most work conducted was to get an SD card to initialize. The command sequence for initialization was provided in the SD card standard; however, upon first implementation it was found to function improperly. After tweaking various aspects of timing, all SD cards were found to initialize. At this point, a test program was setup to combine the features of several other test programs. The system was setup to receive 16 values from the USB connection with the computer. As a value is received (number from 1 to 5), an LED is lit on the board. After 16 values are received they are written to the SD card or Flash (if the SD card is not present). The buffer holding the number values is also cleared. By sending a read command, the values can be read from the SD card or Flash and displayed on the LEDs one value at a time. If the resulting LED pattern matches the original pattern entered, the communication occurred properly.

This test proved that both the SD card and Flash circuits were functioning properly. Occasionally, an SD card write or read operation would fail due to insufficient time for the operation. By increasing the time allocated for reading and writing, the system was found to function reliably.

6.1.2 External Module Hardware Tests

The first test to verify the external module hardware focused on the sensor circuitry. As with the main module, a sample loop was created that would update temporary variables holding sensor values every 20ms. The external module was then run with a breakpoint at the end of the sample loop. Through the debugwire connection, the values collected from all sensors could be reviewed. The temperature sensor and barometric pressure sensor were both found to have reasonable outputs. Tests were conducted by warming the temperature sensor through body heat. Changes were noticed; therefore, verifying operation. Pressure was increased by blowing on the various pressure sensors. All were found to record reasonable changes thus verifying operation.

The humidity sensor was verified last. At first inaccuracies were found with the value measured by the microcontroller. The code was checked and found to be operating properly. Therefore, the system clock speed was checked. Although the clock speed appeared to be set appropriately, it was increased slightly in order to see if result values would change. A slight change was noticed; therefore, the clock was returned to its original frequency. When this occurred, the humidity sensor began to work. The author believes that this mishap was caused by the device becoming setup improperly when transferring between debugwire and ISP programming modes. Debugwire affects the system fuses, which also control the clock frequency; therefore, this explanation appears reasonable.

From these tests it was verified that the system could collect temperature, humidity, and barometric pressure with accuracy of 0.1°F, 0.1inHg, and 10% RH, respectively; therefore, a further system requirement was met.

The external module LEDs were also tested by creating an LED blink function, which would turn the LEDs on and off approximately every second. This was found to function properly; therefore, verifying the LED user interface requirement.

6.1.3 Combined Hardware Tests

The final hardware test was created to verify CAN bus operation. The main module was setup to request information from the external modules by sending a single 8-byte data frame with a character to represent the request. The external modules would then send 8 bytes of sensor data as a reply. This system was difficult to debug, as it was often tricky to determine in which module an error was occurring. During the first attempt, the system did not function at all. After debugging both the main module and the external module, it was determined that the problem was most likely occurring with the main module.

After reviewing the embedded code for the AT90CAN64 microcontroller and reorganizing several CAN functions for transmitting data, CAN transfer was once again attempted. During the second test, communication was immediately established between the devices. Though the system appeared to be operating properly, debugging proceeded in order to ensure that the CAN communication was free of errors. From this continued testing, a mistake was identified in the external module code. After every CAN receive, a flag in the CAN controller is to be cleared before further receives can occur. Clearing of this flag was omitted in the original code.

Clearing the flag also causes the data in the receive register to be discarded. Therefore, if the flag is not cleared the data will remain for viewing. When the CAN test was originally setup, the SPI connection between the microcontroller and CAN controller was used to read the receive register in order to determine if a new data request had arrived. Since the register data was never cleared, the microcontroller would always be informed that a new data request had arrived. Therefore, the mistake was not found until the external module code was modified slightly in order to use the receive interrupt feature of the CAN controller. The device would trigger a microcontroller interrupt only when new data was received. However, without clearing the receive flag, new data was never permitted to enter the CAN controller. Thus, after the first receive interrupt, no other interrupts occurred. By debugging the code with receive interrupts, the error was found and solved.

In the end, it was verified that 2 external modules could properly communicate with the main module on the CAN bus. In the future, a 4-module test should be conducted in order to verify the design requirement of the system. Currently testing 4 devices on the CAN bus is not possible, due to a last minute change made to the external weather boards. The original external module design called for 2 CAN connectors; however, the project sponsor requested to eliminate 1 connector in order to make the external weather modules as compact as possible. Therefore, an additional CAN connector for module daisy-chaining was not included. As a result, a 4 module CAN test remains to be completed in the future.

6.2 Embedded Code Verification

A benefit of conducting the hardware tests first was a complete set of code for interacting with every piece of hardware in both the main and external weather modules. Therefore, the final embedded code could be written by properly piecing the test code together. In this way, it was possible to know that a majority of the final embedded code was functioning properly, and only the logical program flow needed to be verified. Since the only user interface provided by the system is through LEDs, a majority of the embedded code verification was performed through the JTAG boundary scan or debugwire features of the microcontrollers.

As in the hardware verification section, one system requirement was met prior to the start of testing. Through observation, it could be concluded that both the external and main modules were written in Assembly, thus meeting the design requirement.

Due to the simplicity of the external weather collection module and its structure, which is similar to several of the embedded test programs described in Section 6.1, in-depth verification was not performed for this module. It should be noted that from its interaction with the main module, the external module appears to be functioning properly. It responds with correct replies to all CAN commands, and the data returned from sample requests are always reasonable. Furthermore, the data returned are not constant over time (sample values are seen to change), which further confirms its proper operation. Therefore, although the program was not step-through debugged to ensure proper operation, it was verified through its interaction with the main weather module.

Unlike the external module, the main weather module was debugged almost strictly by step-through. Complex areas of the program were targeted for debugging (the CAN initialization loop) while other areas (timing register setup) were often skipped. The first portion of the code verified was the EEPROM initialize that occurs on the first power up of the weather station. During this initialize, the weather station is setup such that it could immediately be placed in a race vehicle and used. It was verified that the default values for the main run time parameters were placed in RAM variables allowing the system to operate; therefore, meeting the design requirement that the system shall require few inputs for off the self use.

When the EEPROM test was first conducted, it appeared that writing to the microcontroller EEPROM was failing. It was soon realized that every time the device was programmed, the EEPROM was erased by the programmer. When using the reset option on the debugger instead, data was found to properly be stored in the EEPROM.

The next code segment verified was the SD card initialize. It was found to properly detect if an SD card was present or write-protected. Furthermore, on the first pass through, the system was found to properly initialize the SD card for operation in the Onboard Weather Station system. On subsequent times through the SD card initialization sequence, the system was found to properly get the memory pointer from the first position on the card. It should be noted, that since an SD card is used which is able to store data until it is full, the system is able to store well over 12 minutes of data between downloads. Therefore, another system design requirement was met. After the SD card was found to be initializing properly, the Flash was tested as well. The Flash initialization was also performed properly by the embedded code.

After SD card initialization, the CAN bus search is performed where the main module attempts to identify the addresses of all external modules on the bus. When first setup, the CAN bus search was not working properly. The system would return that no modules were connected. When examining the search code, it was found that a response was being received from the module connected (therefore the external module was not at fault); however, an error code was also triggered causing the system to ignore the response. The error code was for data length over run error (system receiving more data than expected). The receiver setup was examined, and it was realized that the register for setting the data length was written twice. On the second write, the data length code to expect was cleared. When this was fixed, responses from the external modules were properly received.

The remainder of the initialization code and "Pit Road Mode" was found to work flawlessly. At this point, the manual data record switch was tested. The system was found to enter data record mode when the switch was closed and leave data record mode when the switch was re-opened. The automatic race conditions were more difficult to test in the lab environment; however, verifying the two-step operation was achieved. The two-step input was found to properly trigger data record. If released prematurely, data record would exit and the memory pointer would retract to its position prior to entering data record mode. If held for the proper duration, data record would continue for the time period specified by the user.

From this test, several system requirements were verified. First, it was verified that recording could be triggered through a mechanical switch or automatic race condition. Next it was verified that a two-step press (which is equivalent to a transbrake release) could trigger data record. Thereby, verifying a portion of another system requirement. Finally, it was verified that the automatic trigger condition had to endure a certain period of time for data to be recorded.

The other main aspects of the record loop were found to work properly. Data was captured from all sensors in the main module and appeared to be accurate. Furthermore, 32 bytes of data were received over the CAN bus from every module attached. This portion of the code was stepped-through carefully due to the complex buffer storage scheme implemented. The system is setup to receive the first 8 bytes for every module on the bus almost simultaneously. Before starting the request for the next 8 bytes, the first 8 bytes of each module must be read. If the 8 bytes from each module were to be read and stored in the data buffer in order (which is eventually written to the Flash or SD card), the module data would be interlaced in the buffer. This was not desired as it would make decoding the data more complex. Therefore, buffer spaces are skipped such that certain blocks of the buffer are specified for a particular module.

This code was found to work properly and allowed the design to meet a further system specification. Through this scheme, the system will be able to monitor other data in the future. The only requirement is that new modules have data lengths less than 32 bytes.

The SD card and Flash writes occurred properly and only on every 3rd sample period. All three sampling rates were also tested in order to ensure that timing requirements were met. These tests were conducted without wireless transfer enabled for a reason described in a following paragraph. In order to determine if the sample period was occurring regularly, an LED was made to toggle on every pass through the sample loop. The LED was found to blink at a steady rate for all three sample periods. Therefore, it could be inferred that timing requirements were met for all sample periods.

In the future, a more robust test should be conducted. A counter could be implemented within the loop that waits for the sample period to expire. Using a JTAG breakpoint, the value of this counter could be examined when the wait-loop exits. As long as the counter is always greater than 1, it is known that timing requirements for the system are met. At the current time, it can still be said that the system meets the requirement of supporting 20, 50, and 100ms sampling periods. Timing should also be verified in the future with wireless transfer enabled.

The one feature of the main module embedded code that remains to be tested is wireless transfer. Complications with the weather software prevented wireless transfer from being tested during this project, since the software is needed to initialize the XBee device. If an easy scheme for initializing the XBee device could have been created without the weather software, it would have been possible to verify operation using the X-CTU terminal in order to confirm that data is transmitted every cycle. In the future, wireless tests must be conducted in order to prove that the system is able to transfer 1 packet of data in a sample period.

The computer interface code was initially verified with hyper-terminal. Through this connection, it was confirmed that the SD card and Flash erase, memory check, and data read requests worked properly. The get parameters functions were also tested through hyper-terminal; however, it was difficult to determine if the proper data was transferred, since hyper-terminal displays data as ASCII characters while the weather station transmits actual numeric values. For this reason, the set parameters operations as well as the real time clock setup were impossible to test through hyper-terminal. The remaining computer interface functions were left to be tested with the software.

6.3 Software Verification

Although the software is not yet complete, initial verification was performed in order to prove that the test version of the software complies with system specifications. Through observation, it can be verified that the software is written in Delphi; therefore, meeting one system requirement. The verification status of the remaining software system requirements will be discussed based on the weather software screen they relate to.

The first step in testing the weather software was to ensure that it properly connected to the Onboard Weather Station and maintained the connection after it was established. Both the Onboard Weather Station and weather software were designed to incorporate timeouts. Every second, the weather software sends a character to the weather station to which the weather station must respond. This feature allows both the weather station and weather software to recover if connection is lost.

The weather station was found to properly connect to the weather software and all version information was correctly transferred; however, on the first communication attempt, the connection between weather station and weather software was not maintained. It was eventually determined that the wait period for timeout in the weather station was too short. The wait period was increased, and the connection was found to be maintained. From this error, it was also proved that the timeout feature was fully operational; however, an additional timeout test was completed after the embedded code was modified to ensure that it continued to work properly. The disconnect feature of the weather station was also tested at this time and found to work as intended.

The setup screen was entered second for verification. This screen was first tested to ensure that the user was not able to enter invalid inputs to the system parameter boxes. The get parameters and set parameters buttons were tested next. A problem was found with the get parameters button, which has not yet been rectified. At times, the numbers displayed in the edit boxes do not represent the actual parameters in the system. Set parameters appears to be working properly as the numbers entered in the software were found to be set in the weather station when using the JTAG debugger to verify. Therefore, at the current time, the setup screen is able to meet the system requirement of allowing users to update system parameters; however, work needs to be done in order to remove lingering errors in the code.

The test screen was the third screen verified. It was found to properly display error messages if a module number is specified that is greater than the number of external modules attached to the weather station. Furthermore, as with the setup screen, it correctly displays error messages when communication attempts are made to the weather station and the weather station is not connected. The main functionality of the test screen to be verified was that data could be gathered and displayed from the main weather module and a specified external module. This operation was found to function flawlessly; thereby, verifying another system requirement (the software is able to test data collection modules).

Currently, the software does not have calibration capabilities; therefore, the final requirement for the test screen (calibration of all data modules) was not met in this project. This function will be added in the future; thus, allowing the system to meet all design requirements.

The final screen to be verified was the data viewer. This screen had the most system requirements related to it. At the current time the data viewer is capable of extracting data through a USB connection, storing data in a file, as well as graphing data; however, all of these functions cannot be performed for the data format supported by the Onboard Weather Station. Therefore, these functions must be updated in order to truly meet the system requirements of extracting data from the weather station, storing data to a file, and graphing weather data for the user.

The data viewer screen is also required to erase the memory of the weather station. Although it has this functionality, the communication protocol is also setup for a prototype weather station; therefore, this requirement is not met for the current weather station. The data viewer screen is able to calculate corrected altitude, water grains, and vapor pressure; thus partially meeting a system requirement for the software. However, it is not yet able to calculate the horse power correction factor; therefore, future work remains for this requirement to be completely fulfilled.

Performance predictions must also be added in the future to fulfill another system requirement. At the current time they are not supported due to the fact that further data interpretation is necessary in order to determine how performance will be affected by the weather fluctuations during a race. Different algorithms will be tested before this system requirement is fulfilled.

The final system requirement that remains to be completed is that the system should accept wireless transfer of data during a race and prevent the data from being viewed until after the race is complete (after the transmission has ended). This functionality will be added in the future in order to fulfill the final software requirement for the Onboard Weather Station system.

6.4 System Verification

From the initial tests described in the first 3 sections, the Onboard Weather Station appears to be functioning properly and within design specifications; however, the true test of the system will be on-car testing. This testing will prove that the system is able to withstand a race environment, which has many external factors that cannot be simulated in a lab. At the current time, the system is prepared for on-car testing. The only necessary feature for it to be conducted is a case for the external sensor circuitry. The data viewer would also have to be updated such that the recorded data could be retrieved in order to verify proper system operation. This testing will be completed in early June 2010.

One feature of the embedded code that will best be tested in a race environment is automatic data record triggering using RPM. Through stepping, it was possible to verify that the device can be setup to use RPM as a trigger. However, it is difficult to simulate the RPM fluctuations that could occur during a race in a lab environment. For this reason, on-car testing is necessary before the system requirement of triggering with RPM can be completely verified.

Although the current weather station has not yet been tested on a car, graphs exist from a prototype weather station that was constructed by the author over the summer of 2009. This weather station did not undergo the same in-depth design process needed to develop the Onboard Weather Station; however, from this device, preliminary weather results were captured. A sample graph can be seen in Figure 38. The run is from 18 to 28 seconds. As can be seen, conditions drastically change from those recorded prior to the race (around 10 seconds). The groundwork that this weather station provides further shows the necessity of the Onboard Weather Station. It will be interesting to see what is learned from weather readings in the future.

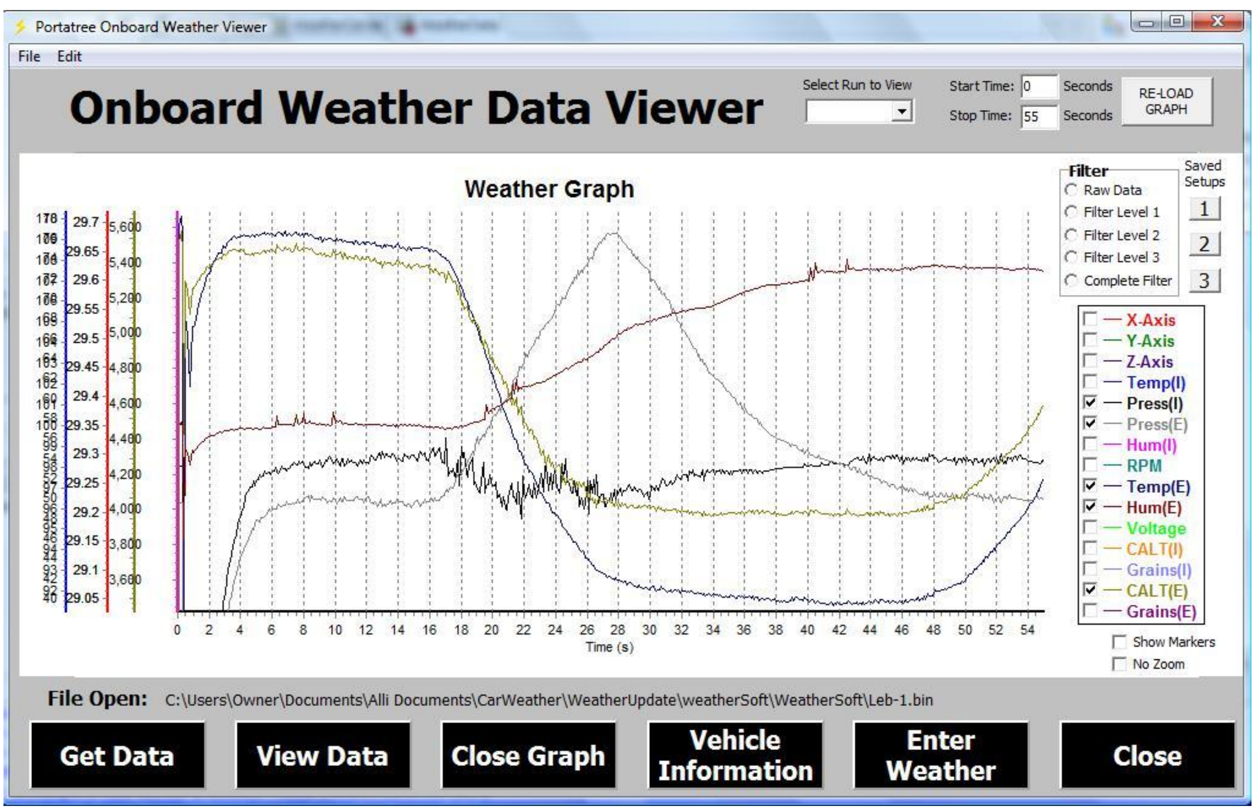


Figure 38: Weather Software Graph

7 Future

Although the system designed in this project met all design requirements specified by the project sponsor, there are still necessary future improvements before it can be put into production. First, a battery backup circuit must be added to the device in order to allow the system to shut down without corrupting Flash or SD card memory if power is abruptly lost during data recording. The battery backup circuit is also necessary for proper operation of the real time clock of the weather station. In the current system, when the device is shut down, power is also removed from the real time clock circuit. This results in the real time clock being reset at each system start, which defeats the purpose of a real-time clock. A battery backup circuit has already been designed for the Onboard Weather Station and the firmware has been updated for battery backup circuit operation. Work is being done to incorporate this modification into the PC board.

Before mass manufacturing of the weather station begins, additional small modifications are needed to the main module PC board. These modifications involve increasing the size of footprints for 2 components on the board as well as reversing the direction of two components whose footprints were placed backwards in the original layout. It may also be beneficial to remove the ground plane from the temperature sensors on both the external and main weather boards. The ground plane connects all components on the board and accumulates heat as the device is in operation. This heat may radiate from the board to the temperature sensor; thereby, affecting temperature readings. Therefore, removing the ground plane from around the temperature sensors may result in better system accuracy after the device has been in operation for long periods of time. The final required PC board change involves moving the various connectors of the boards to better locations to make them easier to access in the final casing.

Another area where future work remains is in the software design. The purpose of this project was to create the hardware and firmware for the weather station system; therefore, the software developed was for the sole purpose of testing the system hardware and firmware. Although the software created provides a good base for future work, updates to the data viewer screens and the addition of a calibration screen are needed before it can be released to the public.

Toward the end of the project, the sponsor expressed a desire to develop the Onboard Weather Station into a complete data acquisition system in the future. Although there was enough time to generalize the communication protocol of the system to support this improvement, there was not enough time to develop new external modules for testing. In the future, these devices can be easily added to the system using similar firmware and hardware to that originally created for the external weather modules. The only required changes will be to replace the sensors with the new ones desired and to update the sampling loop of the external module firmware to support these new sensor types. The sponsor has indicated a desire to create thermocouple and oxygen sensor modules, which are standard sensor packs on data acquisition systems.

Another possible external module that could be developed in the future is a touch screen display for updating weather station parameters without requiring a computer. This improvement would require further development than the additional sensor packs mentioned above. An external module of this type may eventually lead to the development of a completely standalone data acquisition system. Such a system may be beneficial to racers who prefer not to bring a laptop to the track, yet still want all the benefits of a data acquisition system.

Finally, additional data analysis needs to be completed before the device can be released to the market. This is necessary, since it is unreasonable to expect racers to buy a tool that the developers have not yet figured out how to use. Several approaches to interpreting the new weather data have been suggested. These approaches will be tested over the summer of 2010 in order to determine if they result in valid performance predictions. If the data interpretation methods are found to function properly, the device will be released to the market in late August 2010.

Overall, the necessary improvements for selling the system in the future have been identified as well as future improvements that would enhance the value of the Onboard Weather Station to the customer. As a whole, the future improvements to the system are minimal making it likely that the August 2010 release date will be achieved.

8 Acknowledgments

I would like to thank Bruce Land for his consultations throughout the year regarding this project as well as his great advice for many aspects of the design. Furthermore, I would like to thank him for offering to advise this unusual project. Without his support, the Onboard Weather Station design would not have been possible.

Special thanks are also given to Portatree Timing Systems, Inc. (www.portatree.com) for sponsoring this project. The advice, patience, and support of the staff at Portatree have advanced this project beyond what would have been achievable from simple student design work. Furthermore, the Atmel design tools they provided for work in this project have significantly improved the results achieved. Enough thanks cannot be given.

Final thanks should be given to Tom Chatt, who generously lent his camera to guarantee premier quality photographs of the actual Onboard Weather Station.

9 References

[1] *SD Specifications Part 1 Physical Layer Simplified Specification*, SD Group Standard 2, September 25, 2006

[2] *Universal Serial Bus Revision 2.0 Specification*, USB-IF 24, March 23, 2010

[3] *Controller Area Network (CAN) – Part 1: Data link layer and physical signaling*, ISO 11898, November 19, 2003

[4] Atmel Corporation. (2008, August). *AT90CAN32/64/128 Datasheet* (rev. H) [online]. Available: //www.atmel.com/dyn/resources/prod_documents/doc7679.pdf

[5] Atmel Corporation. (2010, March). *ATtiny48/88 Preliminary Datasheet* (rev. D) [online]. Available: //www.atmel.com/dyn/resources/prod_documents/doc8008.pdf

[6] National Semiconductor. (2000, November). *LM34 Precision Fahrenheit Temperature Sensors* [online]. Available: //www.national.com/ds/LM/LM34.pdf

[7] Measurement Specialties. (2008, April). *HS1101LF – Relative Humidity Sensor* [online]. Available: //www.meas-spec.com/downloads/HS1101LF.pdf

[8] Freescale Semiconductor. (2007, January). *Media Resistance and High Temperature Accuracy Integrated Silicon Pressure Sensor for Measuring Absolute Pressure, On-Chip Signal Conditioned, Temperature Compensated, and Calibrated* (rev. 4) [online]. Available: //www.freescale.com/files/sensors/doc/data_sheet/MPXAZ6115A.pdf

[9] Freescale Semiconductor. (2008, April). *±4g, ±12g Three Axis Low-g Micromachined Accelerometer* (rev. 0) [online]. Available: //www.freescale.com/files/sensors/doc/data_sheet/MMA7331L.pdf

[10] Microchip. (2007). *Stand-Alone CAN Controller with SPI Interface* [online]. Available: //ww1.microchip.com/downloads/en/DeviceDoc/21291F.pdf

[11] Silicon Labs. (2008, August). *Single-Chip USB to UART Bridge* (rev 1.3) [online]. Available: //www.silabs.com/Support%20Documents/TechnicalDocs/cp2102.pdf

[12] Digi International. (2007, May 31). *XBee/XBee-PRO OEM RF Modules* (rev 1.xAx) [online]. Available: //ftp1.digi.com/support/documentation/manual_xb_oem-rf-modules_802.15.4_v1.xAx.pdf

[13] Atmel. (2009, April). *16-Megabit 2.5 volt or 2.7 volt DataFlash* (rev. M) [online]. Available: //www.atmel.com/dyn/resources/prod_documents/doc3500.pdf

[14] Portatree. (2010). *Snap-In Weather Station* [online]. Available: //www.portatree.com/item—Snap-In-Weather-Station--snap-in_weather.html

[15] Altronics. (2010). *PerformAire* [online]. Available: //www.altronicsinc.com/

[16] Computech Racing. (2010). *RaceAir Pro* [online]. Available: //computechracing.com/Product_Guide/weatherstation-portable.html

[17] Racepak Data Systems. (2010). *Sportsman* [online]. Available: //www.racepak.com/Drag_Racing/

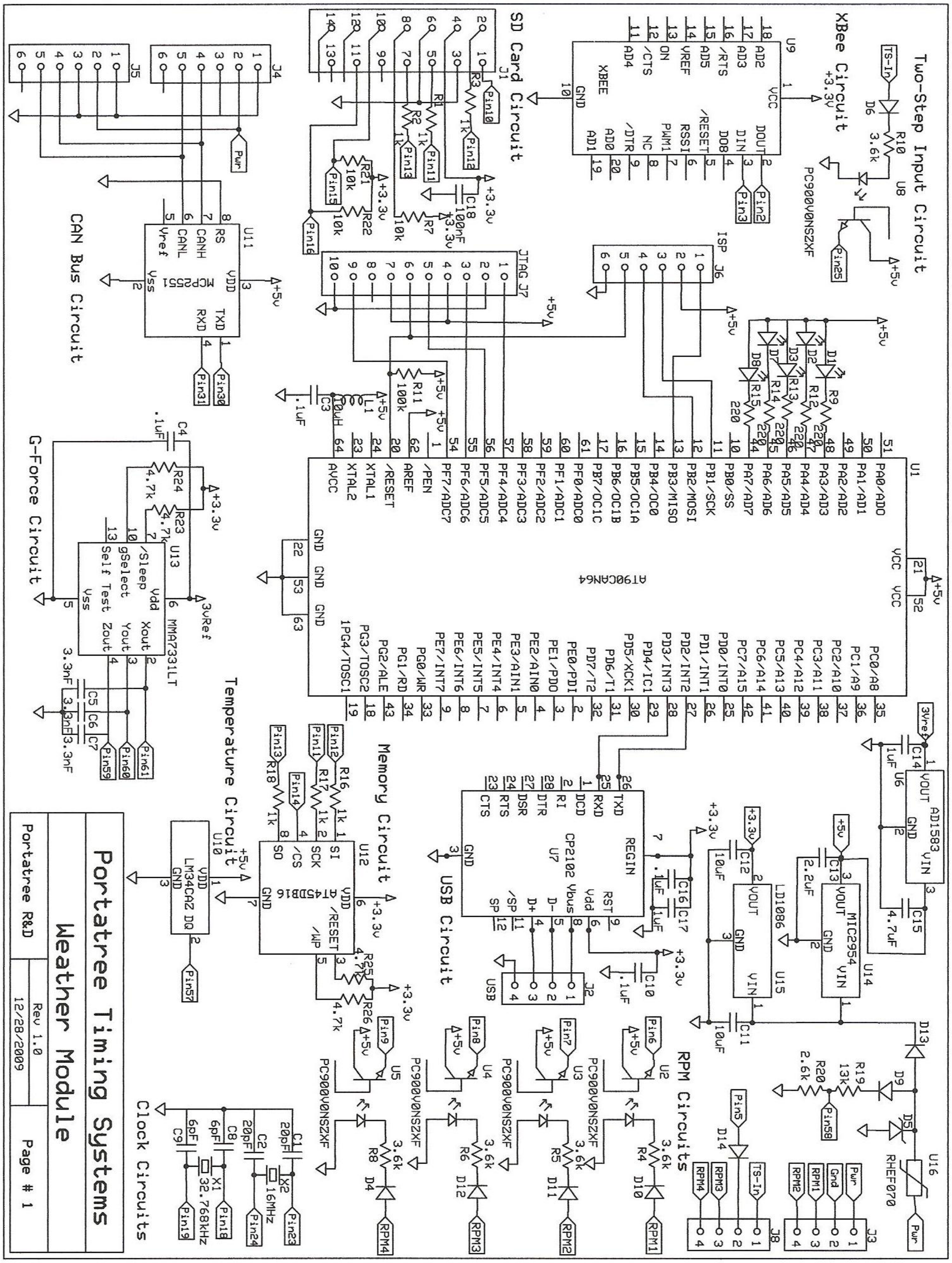
[18] Corsa Instruments. (2010). *The Corsa Data Logger – EZ II* [online]. Available: //www.corsa-inst.com/ez1.html

[19] L. Elwell. (2008, January 26). *Why Miles per Gallon Go Down in Cold Weather* [online]. Available: //www.scribd.com/doc/2160815/-Why-Miles-per-Gallon-Go-Down-in-Cold-Weather

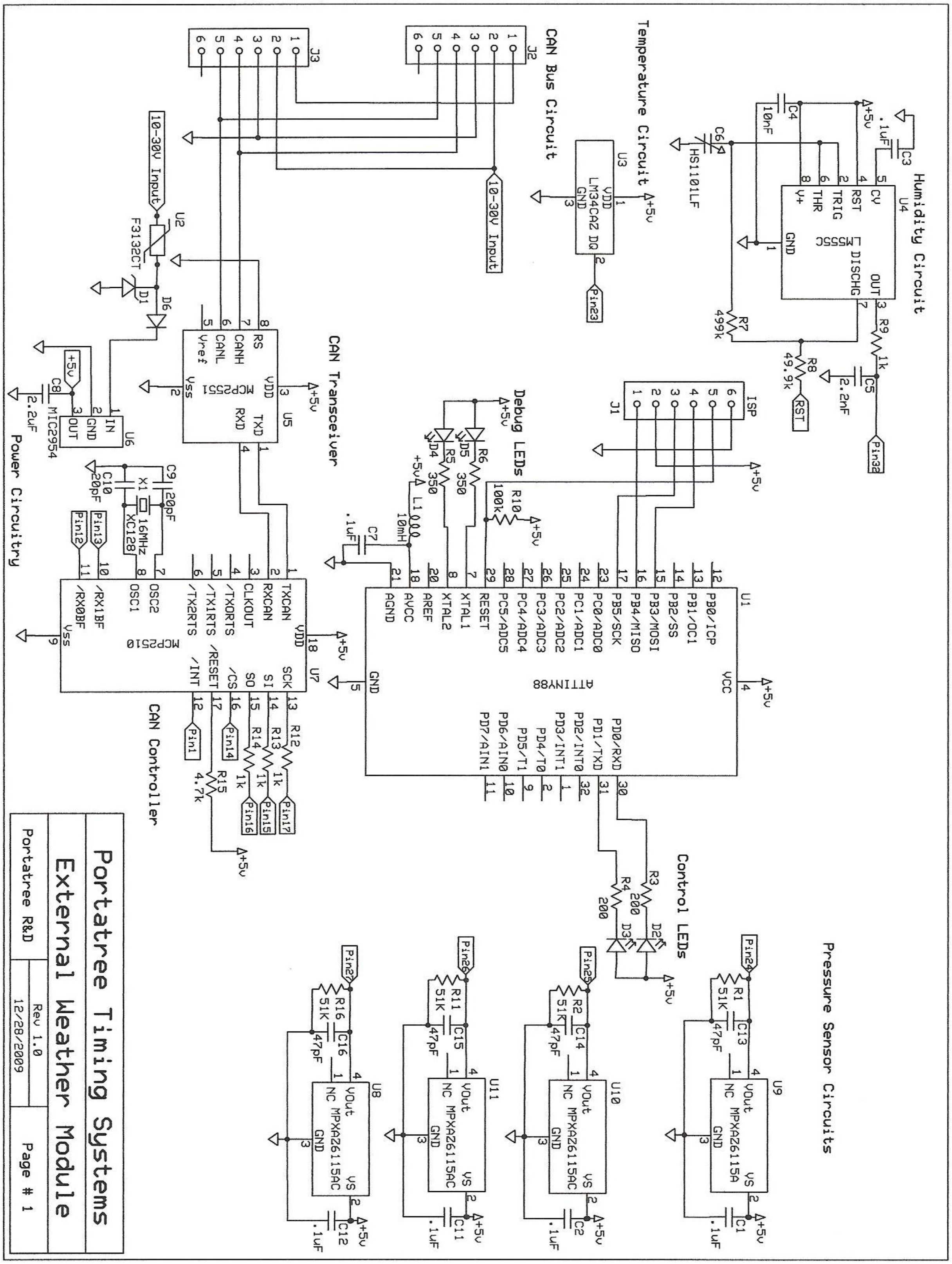
[20] *2010 Official IHRA Motorsports Rulebook*. International Hot Rod Association, Norwalk, OH, 2010, pp. 92-93

[21] *2010 Official NHRA Motorsports Rulebook*. National Hot Rod Association, Glendora, CA, 2010, pp. 49-50.

Appendix A: Main Module Schematic



Appendix B: External Module Schematic



Appendix C: Software User's Manual

The software created for this project is subject to change in the future as it was strictly designed for testing of the firmware and hardware of the Onboard Weather Station.

Main Screen:

Upon starting the program, it enters the main screen where 6 options are available. The options are also available under the drop down menu located at the top of the screen.

Connect: This button will cause the software to attempt to connect to the Onboard Weather Station through the COM Port displayed on the screen. While attempting to establish communication, the button will change to "Connecting." After communication has been established "Connected" will be displayed. When communication is established a green panel will also appear on the screen as well as the Version # and Version Date of the Onboard Weather Station connected. Once the Onboard Weather Station is connected, an option appears under File in the main menu to disconnect the weather station from the software.

COM Port: This button changes the COM port the software uses to attempt to communicate with the Onboard Weather Station.

Setup: This button opens the Onboard Weather Station setup screen where system parameters of the weather station can be altered.

System Test: This button opens the test screen for the Onboard Weather Station where data collection modules can be monitored to ensure they are functioning properly.

Data Viewer: This button opens the data view screen where data can be downloaded from the Onboard Weather Station and graphed.

Exit: This button exits the Onboard Weather Station software.

Setup Screen:

The setup screen contains drop down or edit boxes for changing the sample rate, sample time (length of time to collect data after an automatic trigger), trigger type (manual switch or automatic based on race condition), type of automatic trigger (two-step or RPM), automatic trigger threshold (if RPM is selected), trigger duration (amount of time a trigger condition must be maintained to mean a run), number of external data modules, wireless on/off, and address for the wireless device.

Get Parameters: This button will get the parameters for all selections above from the weather station. This button will only function if the weather station is connected to the software. Otherwise a warning is given.

Set Parameters: This button will update the parameters modified by the user in the weather station if it is connected. Otherwise a warning is given.

Clear Parameters: This button clears the parameter boxes listed above.

Close: This button closes the setup screen and returns to the main screen.

System Test Screen:

The system test screen contains read-only edit boxes for x-axis, y-axis, and z-axis accelerometer data as well as the internal temperature of the main module. It also contains edit boxes for the temperature, barometric pressure, wind pressure, humidity, corrected altitude, and water grains experienced by a particular module connected to the Onboard Weather Station. The module to monitor can be selected in the "Poll Module" edit box on the right side of the page. For continuous monitoring, the interval between data collections can be selected in the "Monitor Time" edit box also on the right side.

Check Values: The check values button will request data from the selected module of the weather station. It will update all read-only edit boxes with data from the selected module. If the weather station is not connected an error message will occur and the edit boxes will not be updated.

Monitor Values: The monitor values button will perform the same operation as the check values button; however, it will perform this operation continuously until the button is pressed again. The rate that it requests data from the weather station is given by the "Monitor Time" parameter. If a weather station is not connected, the software will not attempt to perform this operation.

Close: This button closes the test screen and returns to the main screen.

Data Viewer Screen:

The data viewer screen is still in progress; however, the general operation will be described.

Get Data: This button will make a request to the weather station to send the data stored in the Flash and SD card. When it is pressed a file selection screen will appear allowing the user to select or create a file in which to store the data from the weather station. This screen will also allow the user to select to open the data immediately after the download completes and erase the data from the weather station after transfer. During transfer, it will break the data into runs and store them in separate files based on a name specified by the user. If no data is present to download, a message will be returned to the user. This button will only work when the system is connected to the Onboard Weather Station.

View Data: This button opens a file selection screen where a data file to view can be selected. After a file is selected, the data will be opened in the array present in the center of the Data Viewer Screen.

Graph: This button can be pressed once a data file has been selected by the view data button. The opened data is graphed. Time markers are available to denote important areas of the run. The data to be viewed can be selected by clicking the check boxes next to the series names on the right side of the graph. The graph can be closed by pressing the graph button again.

Vehicle Information: This button opens a vehicle information selection screen where the number of cylinders and stroke type of the vehicle can be entered. This information is important for calculating RPM. If these numbers are not entered correctly for your vehicle, the RPM numbers will be wrong.

Enter Weather: This button opens a screen to enter weather values collected by a traditional weather station. This allows one to see how the weather data collected on the race car is different from the weather seen at the trailer.

Close: This button closes the data viewer screen and returns to the main screen.

Future:

In the future, a calibration screen will be added in order to account for variations between different sensors on the weather module. The data viewer screen will also be completed.

Appendix D: Weather Station Embedded Code

The code written for this project is not available for public viewing at the current time. If desired, a request can be submitted to the author for access to the assembly source code. This request will be reviewed by the project sponsor, Portatree Timing Systems, who will make the final decision in determining if the code should be released.

Appendix E: Main Module and External Module Parts Lists

Please See Next Page.

Onboard Weather – Main Module
Master Parts List – Rev. 1
Allison Smyth, Al Smyth, & Frank Legassey
December 23, 2009

Design Team: Allison Smyth, Al Smyth, Frank Legassey — aps243@cornell.edu

#	QTY	Ref	Value	Description	Dist.	Part No.	MFG	MFG Part No.	Unit	Sub
1	2	C1, C2	20pF	CAP CERAMIC 20PF 50V NP0 1206	Digi-Key	311-1153-1-ND	Yageo	CC1206JRNP09BN200	0.11400	0.22800
2	5	C3, C4, C10, C16, C18	0.1uF	CAP CERM .10UF 50V 5% 0805 SMD	Digi-Key	478-3352-1-ND	AVX Corporation	08055C104JAT2A	0.05000	0.25000
3	3	C5, C6, C7	3.3nF	CAP 3300PF 50V CERAMIC X7R 0805	Digi-Key	399-1153-1-ND	Kemet	C0805C332K5RACTU	0.04600	0.13800
4	2	C8, C9	6pF	CAP CERAMIC 6.0PF 50V NP0 0805	Digi-Key	311-1095-1-ND	Yageo	CC0805DRNP09BN6R0	0.06200	0.12400
5	2	C17, C14	1uF	CAP CER 1.0UF 50V X5R 0805	Digi-Key	587-2229-1-ND	Taiyo Yuden	UMK212BJ105KG-T	0.14300	0.28600
6	2	C11, C12	10uF	CAP CER 10UF 16V X7R 1206	Digi-Key	490-3911-1-ND	Murata Electronics North America	GRM31CR71C106KAC7L	0.49800	0.99600
7	1	C13	2.2uF	CAP CER 2.2UF 50V X7R 1206	Digi-Key	490-3367-1-ND	Murata Electronics North America	GRM31CR71H225KA88L	0.48400	0.48400
8	1	C15	4.7uF	CAP CER 4.7UF 50V X5R 1206	Digi-Key	587-1962-1-ND	Taiyo Yuden	UMK316BJ475KL-T	0.63800	0.63800
9	3	D1, D2, D3	_	T 1 ¾ (5mm) Flanged LED	Bivar	ELM15005UYC-2K	Bivar	ELM15005UYC-2K	0.27200	0.81600
10	1	D7	_	T 1 ¾ (5mm) Flanged LED	Bivar	ELM15005UCG	Bivar	ELM15005UCG	0.49400	0.49400
11	1	D8	_	T 1 ¾ (5mm) Flanged LED	Bivar	ELM15005URC	Bivar	ELM15005URC	0.21200	0.21200
12	8	D4, D6, D9, D10, D11, D12, D13, D14	_	SCHOTTKY RECT 30V 1.5A SOD323F	Digi-Key	568-4128-1-ND	NXP Semiconductors	PMEG3015EJ,115	0.58000	4.64000
13	1	D5	_	DIODE TVS 30V 3000W 5% UNI SMD	Digi-Key	SMDJ30ACT-ND	Littelfuse Inc	SMDJ30A	1.05000	1.05000
14	1	J1	SD Card	CONN CARD SD PUSH-PUSH R/A SMD	Digi-Key	3M5646CT-ND	3M	SD-RSMT-2-MQ-WF	1.39000	1.39000
15	1	J2	USB	CONN USB RECEPT R/A TYPE B 4POS	Digi-Key	A31725-ND	Tyco Electronics AMP	292304-1	0.67137	0.67137
16	2	J3, J8	Terminal Strip	CONN BARRIER STRIP DL 4POS .325	Digi-Key	A98474-ND	Tyco Electronics	2-1437667-6	1.15000	2.30000

17	1	J4	CAN	Female						
18	1	J5	CAN	Female						
19	1	J6	ISP	CONN HEADER 6POS .100 STR TIN	Digi-Key	609-3218-ND	FCI	67996-406HLF	0.37000	0.37000
20	1	J7	JTAG	CONN HEADER 10POS .100 STR TIN	Digi-Key	609-3243-ND	FCI	67997-410HLF	0.35000	0.35000
21	1	L1	10uH	INDUCTOR 10UH 150MA 10% SMD	Digi-Key	587-2045-1-ND	Taiyo Yuden	LBR2012T100K	0.12500	0.12500
22	6	R1, R2, R3, R16, R17, R18	1kΩ	RES 1.00K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM1.00KFRC T-ND	Rohm Semiconductor	MCR18EZPF100 1	0.04600	0.27600
23	5	R4, R5, R6, R8, R10	3.6kΩ	RES 3.60K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM3.60KFCT-ND	Rohm Semiconductor	MCR18EZHF360 1	0.04700	0.23500
24	5	R9, R12, R13, R14, R15	220Ω	RES 220 OHM 1/4W 1% 1206 SMD	Digi-Key	RHM220FRCT-ND	Rohm Semiconductor	MCR18EZPF220 0	0.04700	0.23500
25	1	R11	100kΩ	RES 100K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM100KFCT-ND	Rohm Semiconductor	MCR18EZHF100 3	0.04700	0.04700
26	1	R19	13kΩ	RES 13.0K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM13.0KFCT-ND	Rohm Semiconductor	MCR18EZHF130 2	0.04700	0.04700
27	1	R20	2.6kΩ	RES 2.61K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM2.61KFCT-ND	Rohm Semiconductor	MCR18EZHF261 1	0.04700	0.04700
28	3	R7, R21, R22	10kΩ	RES 10.0K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM10.0KFRC T-ND	Rohm Semiconductor	MCR18EZPF100 2	0.04700	0.14100
29	4	R23, R24, R25, R26	4.7kΩ	RES 4.70K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM4.70KFRC T-ND	Rohm Semiconductor	MCR18EZPF470 1	0.04600	0.18400
30	1	U1	_	IC MCU AVR 64K FLASH 64-TQFP	Mouser	556-A90CAN64-16AU	Atmel	AT90CAN64-16AU	12.25000	12.25000
31	5	U2, U3, U4, U5, U8	_	PHOTOCOUPLER OPIC DGTL OUT 6-DIP	Mouser	852-PC900V0 NSZXF	Sharp Microelectronics	PC900V0NSZXF	0.78000	3.90000
32	1	U6	_	IC V-REF PREC 3V SOT-23	Digi-Key	AD1583BRTZ-REEL7CT-ND	Analog Devices Inc	AD1583BRTZ-REEL7	1.83000	1.83000
33	1	U7	_	IC USB-TO-UART BRIDGE 28MLP	Digi-Key	336-1160-ND	Silicon Laboratories Inc.	CP2102-GM	2.21000	2.21000
34	1	U9	_	MODULE 802.15.4 100MW WIRE ANT	Digi-Key	XBP24-AWI-001-ND	Digi International /Maxstream	XBP24-AWI-001	32.00000	32.00000
35	1	U10	_	IC SENSOR TEMP PREC FAHR TO-92	Digi-Key	LM34CAZ-ND	National Semiconductor	LM34CAZ/NOPB	7.14000	7.14000
36	1	U11	_	IC TRANSCEIVER CAN HI-SPD 8-SOIC	Digi-Key	MCP2551-I/SN-ND	Microchip Technology	MCP2551-I/SN	1.12000	1.12000
37	1	U12	_	IC FLASH 16MBIT 66MHZ 8SOIC	Digi-Key	AT45DB161D-	Atmel	AT45DB161D-	2.26000	2.26000

						SU-ND		SU		
38	1	U13	_	IC ACCELEROMETER 4G-12G 14-LGA	Mouser	841-MMA7331LT	Freescale Semiconductor	MMA7331LT	2.32000	2.32000
39	1	U14	_	IC REG LDO 250MA 5V 0.5% SOT223	Digi-Key	576-1152-ND	Micrel Inc	MIC2954-02WS	2.05000	2.05000
40	1	U15	_	IC REG LDO POS 3.3V 1.5A TO-220	Mouser	511-LD1086V33	STMicroelectronics	LD1086V33	1.04000	1.04000
41	1	U16	_	POLYSWITCH RHE SERIES 0.7A HOLD	Digi-Key	RHEF070-ND	Tyco Electronics	RHEF070	0.45000	0.45000
42	1	X1	32.768kHz	CRYSTAL 32.768KHZ 6PF SMD	Digi-Key	728-1004-1-ND	Seiko Instruments	SPT2AF-6PF20PPM	0.70000	0.70000
43	1	X2	16MHz	CRYSTAL 16.00MHZ 20PF SMD	Digi-Key	XC1282CT-ND	ECS Inc	ECS-160-20-5PX-TR	0.73000	0.73000
									Total:	**$86.78**

Onboard Weather – External Module
Master Parts List – Rev. 1
Allison Smyth, Al Smyth, & Frank Legassey
December 21, 2009

Design Team: Allison Smyth aps243@cornell.edu
Al Smyth
Frank Legassey

#	QTY	Ref	Value	Description	Dist.	Part No.	MFG	MFG Part No.	Unit	Sub
1	6	C1, C2, C3, C7, C11, C12	0.1uF	CAP CERM .10UF 50V 5% 0805 SMD	Digi-Key	478-3352-1-ND	AVX Corporation	08055C104JA T2A	0.05000	0.30000
2	1	C4	10nF	CAP 10000PF 50V CERAMIC X7R 1206	Digi-Key	399-1234-1-ND	Kemet	C1206C103K5 RACTU	0.12100	0.12100
3	1	C5	2.2nF	CAP CER 2200PF 50V C0G 5% 0805	Digi-Key	445-1332-1-ND	TDK Corporation	C2012C0G1H 222J	0.19700	0.19700
4	1	C6	_	SENSOR RH CAPACITIVE SIDE OPEN	Digi-Key	HS1101LF-ND	Measurement Specialties/ Humirel	HS1101LF	14.43000	14.43000
5	1	C8	2.2uF	CAP CER 2.2UF 50V X7R 1206	Digi-Key	490-3367-1-ND	Murata Electronics North America	GRM31CR71 H225KA88L	0.48400	0.48400
6	2	C9, C10	20pF	CAP CERAMIC 20PF 50V NP0 1206	Digi-Key	311-1153-1-ND	Yageo	CC1206JRNP0 9BN200	0.11400	0.22800
7	4	C13, C14, C15, C16	47pF	CAP CERAMIC 47PF 50V NP0 0805	Digi-Key	311-1107-1-ND	Yageo	CC0805JRNP0 9BN470	0.06000	0.24000
8	1	D1	-	DIODE TVS 30V 3000W 5% UNI SMD	Digi-Key	SMDJ30ACT-ND	Littelfuse Inc	SMDJ30A	1.05000	1.05000
9	1	D2	_	T 1 ¾ (5mm) Flanged LED	Bivar	ELM15005UCG	Bivar	ELM15005UC G	0.49400	0.49400
10	1	D3	_	T 1 ¾ (5mm) Flanged LED	Bivar	ELM15005URC	Bivar	ELM15005UR C	0.21200	0.21200
11	1	D6	_	SCHOTTKY RECT 30V 1.5A SOD323F	Digi-Key	568-4128-1-ND	NXP Semiconductors	PMEG3015EJ, 115	0.58000	0.58000
12	1	J1	ISP	CONN HEADER 6POS .100 STR TIN	Digi-Key	609-3218-ND	FCI	67996-406HLF	0.37000	0.37000
13	1	J2	CAN							
14	1	J3	CAN							
15	1	L1	10mH	INDUCTOR 10MH 25MA 5% 2220	Digi-Key	445-3666-1-ND	TDK Corporation	NL565050T-103J-PF	0.90000	0.90000
16	4	R1, R2, R11, R16	51kΩ	RES 51.0K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM51.0KFRCT -ND	Rohm Semiconductor	MCR18EZPF5 102	0.04600	0.18400
17	2	R3, R4	220Ω	RES 220 OHM 1/4W 1% 1206 SMD	Digi-Key	RHM220FRCT-ND	Rohm Semiconductor	MCR18EZPF2 200	0.04700	0.09400
18	1	R7	499kΩ	RES 499K OHM 1/8W .1% 0805 SMD	Digi-Key	RG20P499KBCT -ND	Susumu Co Ltd	RG2012P-4993-B-T5	0.54900	0.54900
19	1	R8	49.9kΩ	RES 49.9K OHM 1/8W .1% 0805 SMD	Digi-Key	RG20P49.9KBCT	Susumu Co Ltd	RG2012P-	0.54900	0.54900

						-ND		4992-B-T5		
20	4	R9, R12, R13, R14	1kΩ	RES 1.00K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM1.00KFRCT -ND	Rohm Semiconductor	MCR18EZPF1 001	0.04600	0.18400
21	1	R10	100kΩ	RES 100K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM100KFCT-ND	Rohm Semiconductor	MCR18EZHF1 003	0.04700	0.04700
22	1	R15	4.7kΩ	RES 4.70K OHM 1/4W 1% 1206 SMD	Digi-Key	RHM4.70KFRCT -ND	Rohm Semiconductor	MCR18EZPF4 701	0.04600	0.04600
23	1	U1	_	MCU AVR 8K ISP FLASH 1.8V 32TQFP	Digi-Key	ATTINY88-AU-ND	Atmel	ATTINY88-AU	2.30000	2.30000
24	1	U2	_	POLYSWITCH RXE SERIES 0.25A HOLD	Mouser	650-RXEF025	Tyco Electronics	RXEF025	0.34000	0.34000
25	1	U3	_	IC SENSOR TEMP PREC FAHR TO-92	Digi-Key	LM34CAZ-ND	National Semiconductor	LM34CAZ/NO PB	7.14000	7.14000
26	1	U4	_	IC CMOS TIMER 8-SOIC	Mouser	595-TLC555IDR	Texas Instruments	TLC555IDR	0.60000	0.60000
27	1	U5	_	IC TRANSCEIVER CAN HI-SPD 8-SOIC	Digi-Key	MCP2551-I/SN-ND	Microchip Technology	MCP2551-I/SN	1.12000	1.12000
28	1	U6	_	IC REG LDO 250MA 5V 0.5% SOT223	Digi-Key	576-1152-ND	Micrel Inc	MIC2954-02WS	2.05000	2.05000
29	1	U7	_	IC CAN CONTRLER IND TEMP 18SOIC	Mouser	579-MCP2510ISO	Microchip Technology	MCP2510-I/SO	3.28000	3.28000
30	3	U8, U10, U11	_	SENSOR ABS PRESS 16.7PSI MAX	Mouser	841-MPXAZ6115 AC6U	Freescale Semiconductor	MPXAZ6115A C6U	9.57000	28.71000
31	1	U9	_	SENSOR ABS PRESS 16.7PSI MAX	Mouser	841-MPXAZ6115 A6U	Freescale Semiconductor	MPXAZ6115A 6U	9.28000	9.28000
32	1	X1	16MHz	CRYSTAL 16.00MHZ 20PF SMD	Digi-Key	XC1282CT-ND	ECS Inc	ECS-160-20-5PX-TR	0.73000	0.73000
									Total:	**$76.81**