Smokin' Buckeye Telemetry System

Photo: Smokin' Buckeye (3)

Telemetry Systems, Inc.
http://eewww.eng.ohio-state.edu/~hashmia

Dave Slyh
Julie Do
Jung-Hoon Kim
Noorgul Dada
Syed Abbass
Ahsan Hasmi
Shel Davies

3/10/2000
EE 582
Instructors:
Steve Bibyk
Darla Peelle

Table of Contents:

List of Figures:

List of Tables:

Smokin' Buckeye Telemetry System

Introduction:

Competitive automobile racing is a struggle for power management, because the winner of a race is usually the one who best managed the car's power. Every car requires some means of power in order to function; most cars are powered by gasoline. The gasoline is converted into power that runs the engine, the engine allows the drive train to function, and it provides the means of motion for the car. Cars can also be designed to use an alternate power source besides gasoline, such as electric power. An electric powered car can be designed by using DC batteries to power the engine. These DC batteries provide a constant voltage and current to the engine.

An electric energy source could be used to power the engine of a competitive automobile. Several universities in the United States build cars powered by electric engines, and they gather to compete against other schools in electric-car races. The Ohio State University has such a car, the Smokin' Buckeye, as seen on the cover page. The Smokin' Buckeye was built by students at the Ohio State University, and it is sponsored and managed by the Formula Lightning Team.

Background:

The Smokin' Buckeye is an electric Formula-1-style racing car. It uses an AC induction engine, composed of a stator and rotor, powered by DC batteries. The car uses thirty-one 12-volt batteries that are hooked together in series, providing over 372 volts to run the electric engine. The batteries are encased in 16 cells, and they are referred to as a pack. The pack can be removed during a race when it has discharged beyond efficient levels. The Smokin' Buckeye drives into the pit, the pit crew removes the discharged pack, and they install a fully charged one.

The voltage supplied by a fully charged pack is fairly constant for about ten minutes, and then it begins to drop off significantly, as seen in **Figure 1** below. The level at which the supplied voltage begins to drop off significantly is called the knee. If the pack reaches the knee, the engine will run out of power, and the car will have to coast into the pit to obtain a new pack. If the car is too far from the pit when the knee is reached, it may come to a complete stop. Most of the races last about 30 to 40 minutes, therefore, 3 to 4 packs are used for each race. Getting the full potential power out of each pack, without reaching the knee, is the key to winning the race. This is why the power management of the packs is critical.

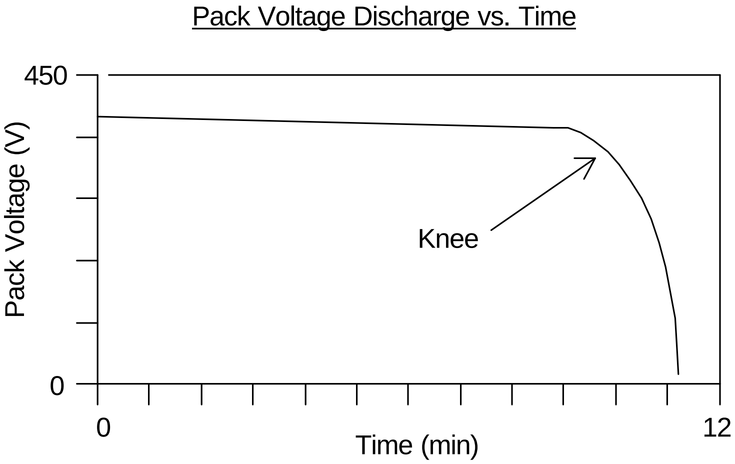


Figure 1: Pack discharge characteristics

Telemetry Advantages:

Power management means more than just having enough power to run a car at a given time. It means analyzing a car's past and present power usage, as well as predicting its future power usage. By using this method, the Smokin' Buckeye can get the optimum life from its packs. Using the batteries more efficiently gives the car a better chance of winning the race.

Effective power management requires looking at all the elements that can provide information about the power usage from the pack. The driver and pit crew work together to manage the power of the car. The team must know the length and shape of the track in order to determine the current position of the car. It also must know the voltage and current of the pack at all times. Knowing the temperature of the stator and rotor can provide insight into when the engine is draining more power from the pack. Analyzing these conditions throughout the race can provide the team with the best strategy for winning.

The Smokin' Buckeye currently has a telemetry system that measures these power management elements, however, it has one critical setback, the data can only be seen after the race is over. Only the driver is able to view the pack voltage and the engine temperature at a given time, therefore, the team is unable to take full advantage of the power in each pack. In order to create a better race strategy, the team requires the ability to see real-time statistics. Since some of the components in the current system are incapable of this function, Telemetry Systems, Inc. is designing a new telemetry system from the ground up.

Design Elements:

The new design has two main objectives. First, a stationary prototype will be built that will simulate the basic functions on the Smokin' Buckeye. This prototype will have an AC induction engine, an inverter, several rechargeable batteries, a rotating wheel and a display unit. It will have sensors similar to five of the main types on the Smokin' Buckeye; rotor temperature, stator temperature, pack voltage, pack current, and wheel speed. The prototype should be simple to build, yet still provide an effective test to verify correct operation of the new system. The second main objective is to meet the customer's expectations as closely as possible. The new system needs to achieve real-time statistics, accurate data, a high sampling rate, and a convenient format to display data.

Obtaining the real-time conditions of all the vital factors requires having a data acquisition system to measure the value of each element. The data acquisition system is comprised of sensors, a Microcontroller unit (MCU), communications links, a user interface, and power supplies, as is shown in **Figure 2** below.

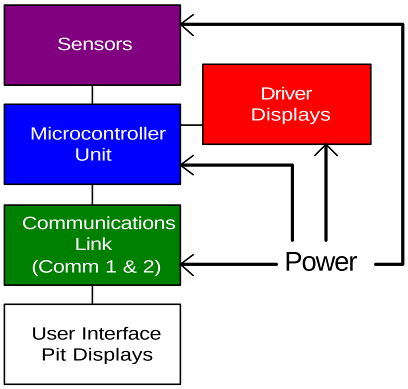


Figure 2: Block diagram overview.

All of these subsystems work together to provide a method for analyzing power management of the Smokin' Buckeye. The sensors read the data and send them to the MCU. The MCU displays certain information to the driver and sends the data to the communications link, and from there it is sent to the pit. In the pit, the communications link collects the data and hands it off to the user interface, where it is organized and presented to the pit crew. All of these components require power, so a separate battery on the car is used to perform this service. Each subsystem has an important role in the telemetry system.

Sensor Systems:

Sensors will be used to measure six critical conditions, such as pack voltage, pack current, stator temperature, rotor temperature, and wheel speed. During the race, the sensors will consistently output their current analog value to the MCU.

The sensors typically use a power supply range of 5 to 12 volts. **Figure 3** below shows the general pin-out of the sensors. The +5V and ground pins are used as power supplies. The output signal pin is the sensors' connection to the MCU. This pin sends the current analog output value, between 0-5 volts, of the sensors to the MCU where it is converted to an 8-bit value through an internal A/D converter.

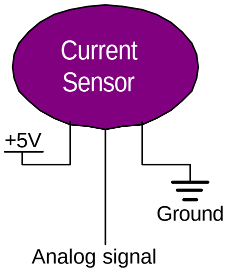


Figure 3: General sensor interface.

Voltage and current sensors on the pack play vital roles in the performance of the car during the race. Their combined input readings will contribute to the analysis of how much power is being consumed at any given instant. The readings will help show the discharge characteristics of the pack during the race. The pit crew can view the instantaneous power curve to determine when to bring the driver in to swap out the used pack. Using the packs to their full potential and knowing when to pit are the key elements of winning a race. This makes the voltage and current sensors critical to the Smokin' Buckeye's success. For the prototype, the Amploc CS50-P current sensor will be used to test the limits of 0 to 500 Amps. The Banner Mini-Beam voltage sensor will be used to test the limits of 0 to 500 Volts.

Stator and rotor sensors are used to monitor the temperature of the engine and power drain of the pack. The stator is the unit that contains the magnetic producing elements of the engine. The rotor is the rotating unit that resides inside the stator's magnetic field. It is this unit that feeds power to the drive train. The stator and rotor do the work of the engine; therefore, they will increase in temperature as they produce energy for the car's motion. The larger the increase in temperature is of these components, the more power that is drawn from the pack. Temperature sensors will be used to measure the temperature of the car's stator and rotor during the race. This will provide evidence of when the pack is being discharged at a higher rate. The Raytek Thermalert MI temperature sensor will test the approximate temperature range of 0 to 300°F (6).

The wheel speed sensor is a Honeywell GT1 Hall-effect-gear-tooth sensor. This sensor is a magnetically biased, hall-effect integrated circuit that accurately senses movement of ferrous metal targets, specifically, the lug nuts, which are used to calculate the distance traveled. It will test the speed range of 0 to 160 mph (1).

All the sensors will read the corresponding data and send the outputs to the MCU when requested. These outputs will be read and converter to 8-bit values that can be stored in the MCU's memory for future reference. The sampling rate of the sensors is controlled by the MCU.

Microcontroller Unit Specifications:

The main purpose of the MCU is to read the data from the sensors at an efficient sampling rate, such as 5Hz or higher, and to process that data to be sent to the communication module and the display. The Microchip PIC16C765 CMOS MCU has been chosen, since it has many efficient features suitable for the telemetry system. The PIC16C765 contains a high-performance CPU. It operates on low power, and it is inexpensive and small in size. It is also designed to interface easily with other components such as the sensors and the communication module. The user of the PIC only needs to learn 35 single word instructions to execute the code (5).

The key features of the PIC16C765 are 8K programmable memory, 256 bytes data memory, 33 input/output ports, 8 analog/digital channels with 8-bit resolution, and the availability of serial communication ports, as shown in **Figure 4** on the next page. **Table 1** on the next page shows the specifications of the PIC16C765 (5).

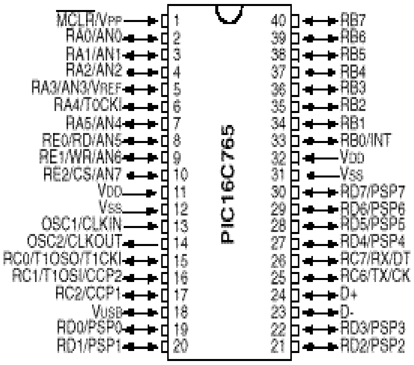


Figure 4: PIC16C765 pin-out structure. (5)

Key features	PIC16C765
Program Memory (14-bit words)	8K
Data Memory (bytes)	256
Dual Port Ram	64
I/O Ports	33 (Ports A, B, C, D, E)
Timer	3
Analog-to-Digital Converter Module	8 channel x 8 bit
Operating Frequency	6 MHz or 24 MHz
Operating Voltage (Volt)	4.35~5.25
Serial Communication	USB,USART/SCI
Parallel Slave Port	Yes
Brown Out Detect Reset	Yes

Table 1: PIC16C765 key specifications.

Microcontroller Unit Programming:

The PIC16C765 has great advantages, and it will speed up the implementation of the design. First, it has innovative programming options to choose from that address procurement issues by simplifying code revisions. One of these programming options is the In-Circuit Serial programming (ICSP). The ICSP allows the MCU to be programmed after being placed in a circuit board, offering tremendous flexibility, reduced development time, increased manufacturing efficiency, and improved time to market.

This popular technology also enables reduced cost of field upgrades, system calibration during manufacturing, the addition of unique identification codes to the system and system calibration. The other advantages include: high-performance, CMOS technology, and it is fully static (5).

The main task that will be required of the MCU will be to go through a loop and gather information from the specified sensors. The MCU interface to the sensors and the communications link can be seen in **Figure 5** below. The voltage sensor will be used to give an example of how the MCU will operate. The MCU will first receive an analog input from the voltage sensor using the ANO input (pin 2). It will then do an analog to digital conversion, which is a built in feature of the PIC16C765. After the conversion is completed, the MCU will output the raw data to the LCD logic to be displayed immediately to the driver using the RB data bus (pins 33-40). Then, a 3-bit sensor code will be tacked onto the data to distinguish it from the other sensor data. The MCU will then store this value in memory with the correct time and date. Finally, the MCU will give this data to the Aerocomm communication link using the Universal Synchronous Asynchronous Receiver Transmitter (USART) port (pins 25-26).

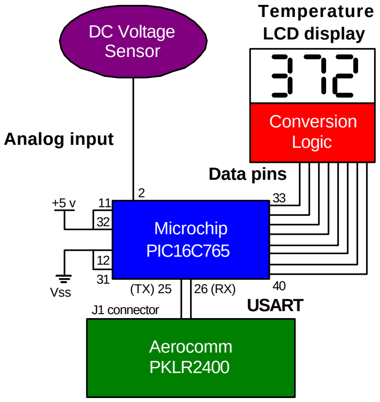


Figure 5: MCU interface design.

Communications Systems:

The communications module will provide the wireless link between the car and the pit for the real-time data acquisition. There are two off-the-shelf components that can be used for this purpose, the Aerocomm PKLR2400 and Linx HP-II series transmission modules. A brief comparison between the two component choices available is shown in the **Table 2** below.

Aerocomm PKLR2400	**Linx HP-II Series**
–Transceivers	–Transmitter & Receiver separate
–Frequency Shift Keying	–FSK
–2.4GHz	–900MHz
–300ft – 3000ft	–1300ft
–800 kbps through air	–50Kbps
–800 kbps Serial – 4 Mbps Parallel	–Serial data Interface
–Serial Asynchronous or Parallel through 40 pin connector	–8 Channels
–77 Channels	

Table 2: Aerocomm and Linx module specifications.

Both the Aerocomm and Linx modules provide the features required to implement the design, but Aerocomm has some advantages over Linx. The features that separate

Aerocomm from Linx are; 2.4GHz carrier frequency, 300-3000ft range, channel hopping spread spectrum technology, fast data transfer, 800 kbps through air, and Aerocomm's ability to provide several modes for efficient data transfer. After analyzing the specifications of the Aerocomm and Linx modules, it is evident that Aerocomm best meets the customer's requirements (2)**.**

The communication system is divided into two separate parts: the transmitter (car) and the receiver side (pit), as seen in **Figure 6** below. On the transmitter side, the transmitter is interfaced to the MCU through serial asynchronous link. Data will be acquired through the MCU. To transmit this data, some type of protocol is required. Aerocomm provides four different ways for data transfer: three transparent modes and one Application Program Interface (API) mode (2).

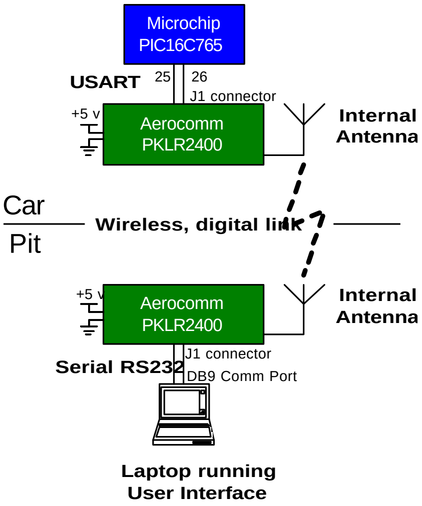


Figure 6: Communications module interface.

In transparent mode, the address of the defined PKLR2400s is saved into an internal EEPROM. Configuration of these transceivers is done off-line. The three different transparent modes are; end character, fixed packet length with timeout, and fixed packet length without time out (2).

In API mode, data transfer is controlled by the Aerocomm's command set. This command set includes transmission and reception configurations and status features. A common application of API mode is to support multi-point wireless link (2).

Due to real-time data acquisition, it is extremely important to verify that the data is error free. Several error detection techniques can be applied to the incoming data packets. The most reliable and commonly used method is cyclic redundancy check or CRC. To implement CRC, the sender uses a CRC-16 or CRC-32 polynomial and attaches it to the data frame in the form of a checksum, as shown in **Figure 7** below. The receiver also keeps the same polynomial. As the receiver receives the data, it quickly divides the data frame with the polynomial. If the remainder is zero, the data is error free, in the case of a non-zero remainder, the sender will retransmit the data (2).

83H	Length Low	Length High	Data	checksum

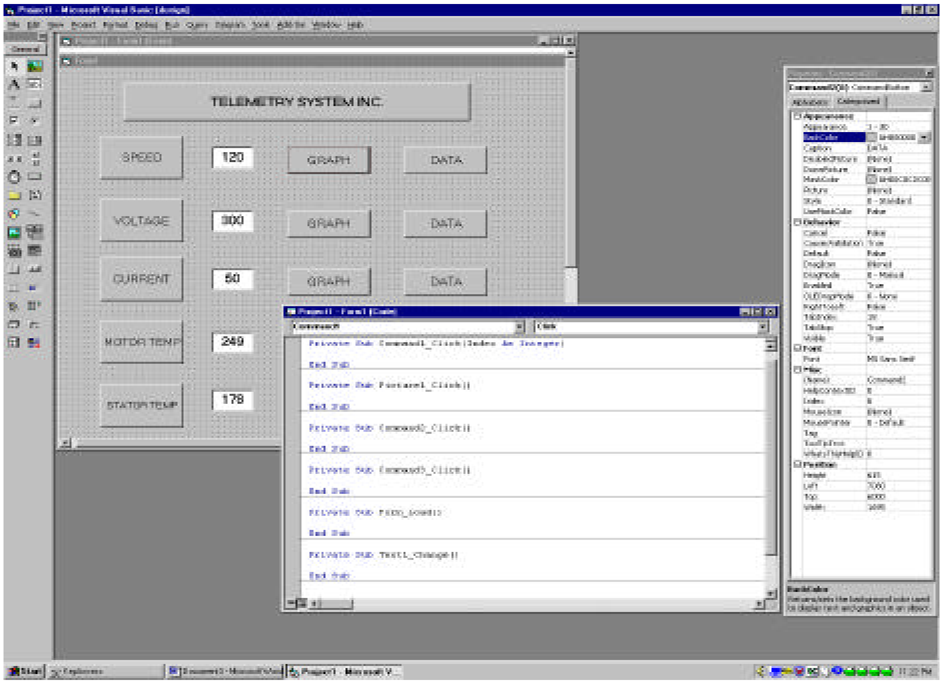
Figure 7: IEEE 802.3 frame format with checksum.

The data is transferred to the laptop, for the user interface, through the serial adaptor board, which has a RS-232 connection. For future enhancements, a high-gain, bi-directional antenna can be used to increase the range of the Aerocomm transceivers.

User Interface:

User Interface (UI) is a broad term that refers to all sorts of communication between a program and its users. It is the method of presenting the data and the interaction of the system with the users. User Interface is not only what the user sees, but also what the user hears and feels. Even the speed with which a program interacts with the user is an important part of the program's UI.

The goal of the user interface is to provide the customer with the real-time values of the average voltage and current of the pack, the temperature of the rotor and stator, and the wheel speed. The display will also contain real-time values graphically. Furthermore, all the data will be stored in a file so that the customer would be able to view the overall data during or after the race. The computer takes data from the communication module and displays it on the laptop screen in the pit via the user interface. A sample of what the pit display looks like is shown in **Figure 8** below.



```
Private Sub Command1_Click(Index As Integer)
End Sub
Private Sub Picture1_Click()
End Sub
Private Sub Command2_Click()
End Sub
Private Sub Command3_Click()
End Sub
Private Sub Form_Load()
End Sub
Private Sub Text1_Change()
End Sub
```

Figure 8: Format of display running on laptop.

The interface will be a Graphical User Interface (GUI), which uses pictures rather than words to represent the input and output of a program. The GUI runs on a laptop in the pit. There are many languages that can be used to accomplish this task, for example C/C++, JAVA, and Visual Basic.

The JAVA programming language is powerful and versatile. The JAVA programs are portable and they can run on multiple systems such as screen phones, mobile phones, desktop computers and network computers, because of Java Virtual Machine (JVC). JVC is a kind of translator that turns general JAVA platform instructions into tailored commands that make the devices do their work. JAVA contains complete set of classes for writing GUI programs, such as Abstract Window Toolkit (AWT) and Swing. The Swing widgets extend AWT by adding a new set of components and a group of related support classes. The Swing collection consists of 17 packages, each of which has its own distinct purposes. The Swing components have had the most immediate impact on Java development. They provide a set of well-groomed widgets and a framework to specify how GUIs are visually presented independent of platform (4).

The development process is divided into two phases; the computer takes the data from the communication module through the RS-232 interface and displays it on the supervisor screen. The outline of the development process is shown in **Figure 9** on the next page.

In Phase 1, the C++ program takes the data from the AEROCOMM module through the COMM port and then stores the data into the data file. At the same time the C++ program stores the data into the data storage files respectively. The data storage files contain overall race data. There are five data storage files for each of the real time value.

In phase II, the JAVA program takes the data from the data file and displays it on the user screen. Before displaying the data on the user screen, the program retrieves the date, time, sensor code and desired data from the preamble. The IEEE 802.3 frame format that the program receives from the communication module is shown in **Figure 10** below.

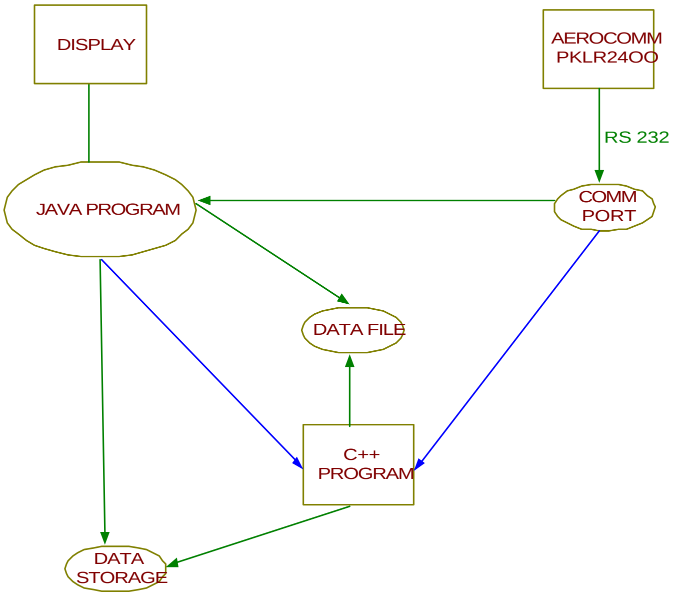


Figure 9: Outline of development process.

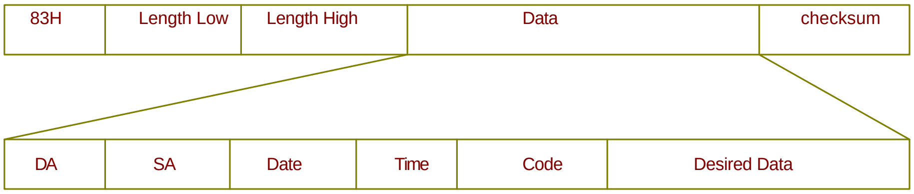


Figure 10: Frame format analyzed by user interface.

The program removes the 83H, length low, length high and check sum from the data portion. The program also verifies the source and destination addresses. The program checks the sensor code and assigns the data value to a specific real-time value. Then the program displays the value on the user screen. When the pit crew wants to view the overall data, the JAVA program goes and grabs the data from the data storage center and displays it on the user screen in tabular format.

The main goal of the user interface in this project is to provide the user with the real time values of the sensor data gathered. Java allows the pit crew to see the Smokin' Buckeye's statistical data in easy-to-read graphs and charts

.

Power Systems:

There are two main aspects concerning power: the first is supplying and regulating the voltage to the system on the car. The second is supplying and regulating the power to the communications module, or receiving unit, which will be transmitting data to a laptop.

Currently, the crew uses Optima spiral wound batteries to run the car. These are 12-volt batteries that are a lighter weight battery with exceptional vibration resistance. The pack is made up of 31 of these batteries that are used to power the engine. Another battery is used to power everything else on the car. This battery is a rechargeable 12-volt Cyclon battery, and it is smaller in size than others. When the power of this system battery starts to run low, the pack batteries are used to actively charge it. It's possible that a switch will be added so that the driver can turn the system off when it's not required in order to conserve energy.

The voltage being transmitted from the diagnostics battery is 12 volts. Some of the components will require this voltage to be regulated to 5 volts in order to work with the equipment that is being installed. To accomplish this, voltage regulators will be installed where needed. A low-dropout, linear regulator has been chosen for this task. Only one regulator should be required for this part, the solution to the problem is simple and inexpensive.

Finally, the last problem concerns supplying power to the receiver unit that will collect the data for the laptop. The receiver unit will be powered by an AC source from a normal outlet. An AC to DC converter will convert to the DC required by the components, and another voltage regulator will be required to obtain the 5 volts.

The first power concerns have clear solutions. The equipment will utilize the power on the car, and the power must be regulated only where needed. The last part is simple as it is just a matter of finding an adequate AC power source and then managing the power as needed. From a power aspect, the biggest challenge will be to manage the power.

Conclusions:

The overall design of the project should accomplish the two main objectives. The interfacing of the subsystems is shown in **Figure 11** on the next page. The sensors gather the required data from the car. The MCU cycles through a continuous loop, reading one sensor value at a time, with a sampling rate of about 5Hz or higher. If a given sensor value happens to be the pack voltage or the engine temperature, then the value is immediately sent to conversion logic that runs to a LCD display in the prototype (car).

Then, all sensor values read in are combined with a sensor code, a date stamp, and a time stamp. This value is stored into the MCU memory until the end of a loop cycle is reached. Here, the data is formatted and handed to the communications link to be delivered to the user interface in the pit. After transmitting the data via a wireless, digital link, the data communications link sends the data to the laptop serial port. The data is analyzed by the Java program and displayed to the pit crew.

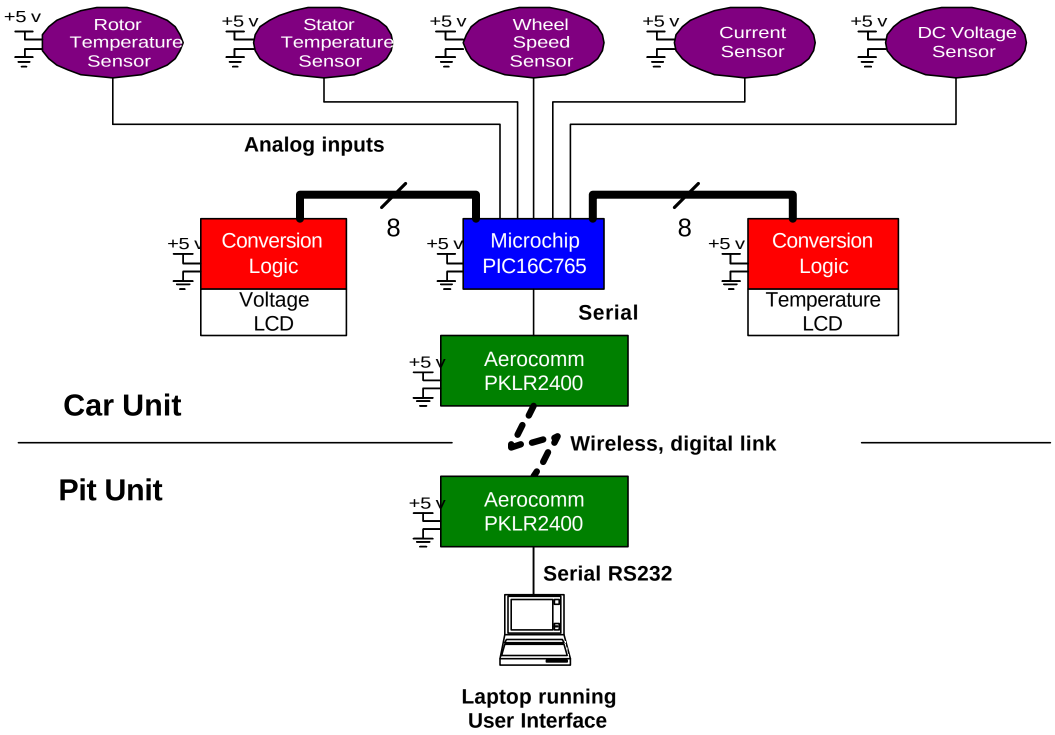


Figure 11: System design overview.

All of the subsystems of the telemetry system are critical to its functioning. They allow for efficient data acquisition of the various power management factors. With the information that the telemetry system provides to the pit crew and driver, the team can create a race strategy that will give them a better chance at winning a race. The availability of analyzing data during and after the race provides a method for managing power in the Smokin’ Buckeye.

Bibliography:

1. http://content.honeywell.com/sensing/prodinfo/solidstate/ then click "GT1 Series Hall Effect Gear Tooth Sensors", downloaded 3/9/2000.

2. http://www.aerocomm.com/OEM2.htm, downloaded 3/9/2000.

3. http://www.eng.ohio-state.edu/archives/9903/9903_smokin.html

4. http://www.java.sun.com/, downloaded 3/9/2000.

5. http://www.microchip.com/10/Lit/PICmicro/16C7XX/index.htm, then click "Datasheet" under PIC16C765, downloaded 3/9/2000.

6. http://www.raytekshop.com/, then click "The Raytek Shop", and then click "View/Print Specifications & Dimensions".