Wireless Data Acquisition System (WiDAS)

Project Proposal

By: Justin Peters

Advisor: Mr. Steven Gutschlag

December 1, 2009

Introduction

Modern racing has become an extremely competitive sport. The theoretical design of a racing vehicle is the backbone for its success on the race track, but not being able to do accurate field testing can be detrimental to the overall performance of its implementation. Analysis of precisely logged data has become crucial to the design process for automotive engineers.

The Wireless Data Acquisition System is intended to be implemented on the Bradley University SAE Formula Car. The system will gather information from an assortment of different sensors throughout the vehicle, paying close attention to signal input ranges and desired error tolerance to provide acceptable data. Once the data has been gathered, it will be sent to both a LCD screen within the vehicle and a wireless transmitter to be received and compiled on an off-track computer on Excel, interfaced with the WinWedge software package. The data that will be collected initially includes car velocity, engine speed, acceleration, engine water and oil temperatures, and suspension travel.

This project is a continuation of many senior project attempts to implement a WiDAS for the Bradley University SAE Formula Car. Only a few, specific parts of legacy design will be used to implement this system, along with some conceptual ideas that, if sound, will be augmented to fit current project parameters.

High-Level Block Diagram

Fig 1.1 **WiDAS High-Level Block Diagram**

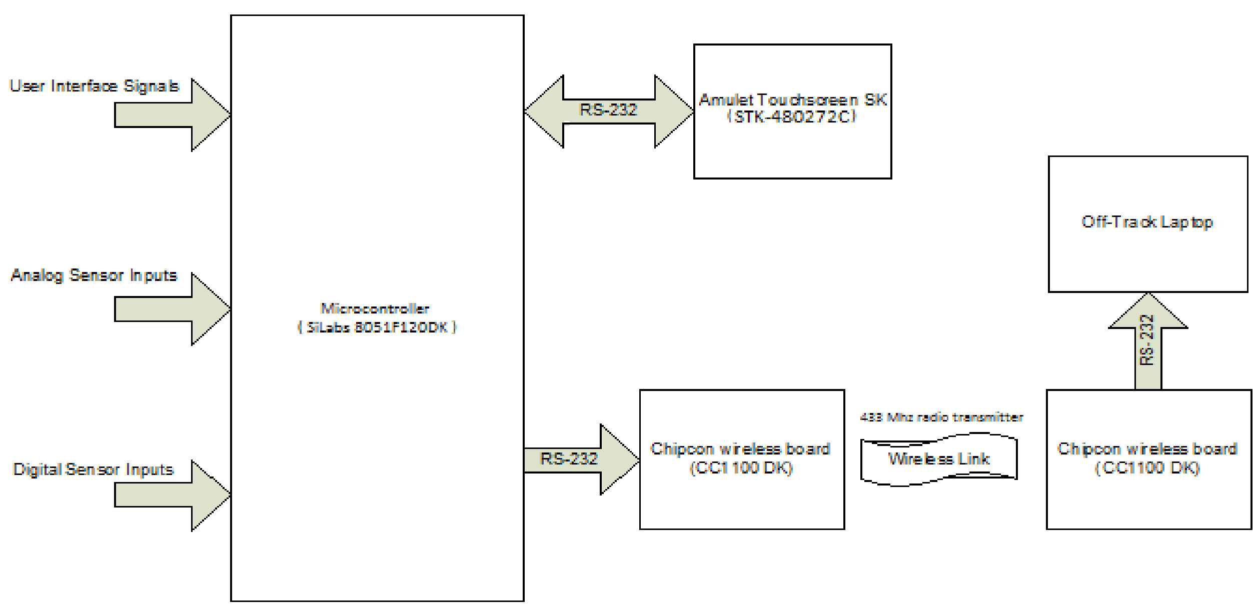

Figure 1.1 is a high level block diagram of the WiDAS system. Various inputs are fed into the microcontroller. Within software, the micro-controller will process this raw data and determine what to send to both the Amulet STK-480272C and the Chipcom CC1100 DK. From there, the touchscreen will display the data and also possibly send user inputs back to the microcontroller. As for the Chipcon device, it will send out the buffered data to its partner on an off-track location where that board will "dump" the data to a laptop through a RS-232 interface. The wireless boards have been programmed using their development libraries so that they act as a variable length FIFO wireless transmitter and receiver pair. Once the buffer has been filled it automatically sends the data and waits for the buffer to be filled again. Redundant transmissions can be implemented to decrease transmission errors and increase system performance.

Software

WiDAS has one primary and three secondary programming tasks to be executed. The primary task involved is programming the microcontroller, which will be the critical computational device involved. The programming of this device will be implemented with C code.

Fig 2.1 High-Level Microcontroller Flowchart

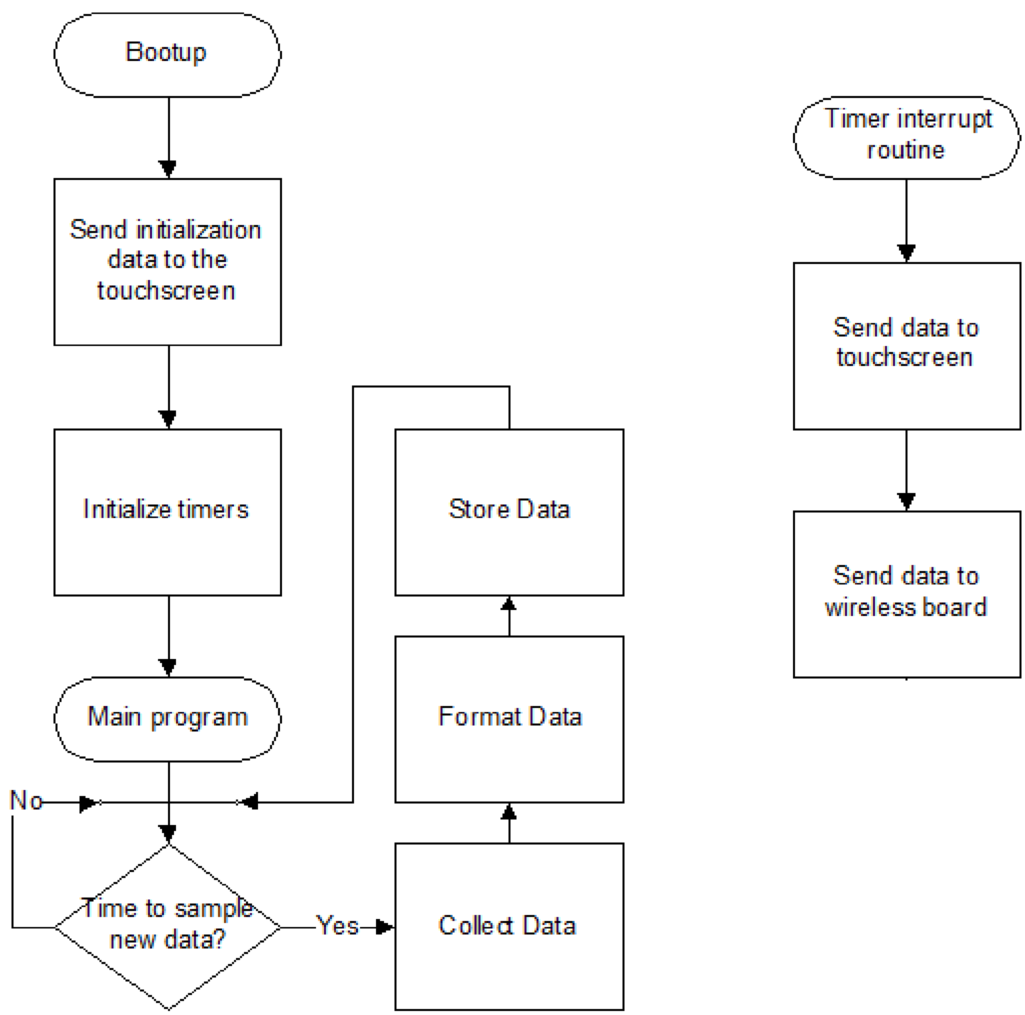

Figure 2.1 shows a first glance at the necessary processes needed to implement WiDAS' micro-controller. This system will rely on timer interrupts due to the time sensitive calculations and the desired data refresh rate that the system will compile data at for this project. The code will be made to be relatively flexible as the SAE Formula Car does not have the data signal specifications defined within well determined limits. The three secondary programming tasks are the Chipcon devices, Amulet touchscreen, and interfacing RS-232 with Excel software. More information is currently being assessed to determine the best method of implementing the three secondary programming tasks at this time.

Functional Requirements

Hardware

- The inputs to the 8051F120DK will be read from one of the various A/D converters, which requires protective circuitry to be designed so that the -0.3 [V] – 5.3 [V] input range is not exceeded. This is specified on the absolute maximum ratings on page 38 of the C8051F12x-13x data sheet supplied by SiLabs.

- Circuitry interface between the 8051F120DK's RS-232 voltage levels with the CC1100 DK's will be created to be within specified data sheet limits. Also, the interface between the CC1100 DK's RS-232 and a standard computer serial port will be ensured to be within manufacturer limitations, dictated by their data sheets.

- The interfaces between all respective connections, which includes all Fig 1.1's connections, will be created with standard shielding to limit noise within the system.

Software

- Critical data shall be sampled every 10 milliseconds. Critical data includes car velocity, engine speed, acceleration, and suspension travel.

- Less critical data shall be sampled every 500 milliseconds. Less critical data includes engine coolant, air temperatures, and oil pressure.

- UART software will be interrupt driven so that it is not poling. This will free up a large amount of processing power that is needed to do all the calculations involved with the data processing and sending that information out to multiple destinations.

- The CC1100 DK's time to process the information inside its buffer, both sender and receiver, is faster than 10 milliseconds so that the system does not get into a situation which it cannot catch up.

- Information received from inputs, either from buttons or the LCD, will be processed during non-critical times. Critical times include taking data, processing data, and sending data.

  → Interrupt driven software execution will occur in the following order for the 10 [ms] routine:

    - Transmission of previously stored data to the CC1100 DK
    - Sampling of all necessary sensors
    - Storing sampled data in memory
    - Count to the fiftieth interrupt to take non-critical sensor input samples for processing (500 ms)

Project History and Preliminary Work

This project is an ongoing effort to complete a wireless data acquisition system for the SAE racing car. Many of the previous attempts built upon each other, integrating the previous projects into their own. Due to the substantial hardware changes that have occurred this year, this will not be the case. First, the wireless component of this project has been modified to the Chipcom devices in Fig 1.1. This was largely due to the fact that the previous Aerocom devices (AC4790-200) were a bottleneck for the system. The new Chipcom devices remedy this problem by having the capability to reach as high as a 500 KHz Baudrate for wireless transmissions. Second, the microcontroller was switched from the previous MCU (EMAC 80515 DK) to the SiLabs 8051F120DK to facilitate higher functionality and ability to fully implement the system in C coding. As the previous projects were implemented in assembly code, the MCU needed to be upgraded. Finally, the LCD touchscreen (STK-480272C) is now a color display. The previous display was a black and white touchscreen display that had been damaged over the years (Amulet STK-A0B3202405). As one can see, the project hardware has changed dramatically, and the old code is no longer viable to be used in this project. The design specifications from the previous years is the only thing that will be followed, as of now. This is a list of SAE racing car sensor data that is needed to design the sensor interfaces.

Preliminary Tasks

The tasks that have been completed so far, ready for final implementation tweaks, have been numerous so far this year. The MCU has been programmed to intake analog and digital signals, which can be converted into whatever format is necessary. An interrupt driven UART system has been implemented for the two UARTs on the MCU, communicating with the Chipcom and Amulet devices. The only thing left to do with the MCU is to understand the data format necessary to communicate with the Amulet device and WinWedge software. Also, test programs with the MCU will be made to assess the performance of the system in separate parts before the entire system has been put together.

The Amulet device has not been thoroughly investigated thus far in the design process. This device is programmed in µHTML. Development tools for this device will be assessed during the winter break period. The WinWedge software will also be looked through to see how to make aesthetically pleasing view of the data on the off-track computer through Excel software.

As for the Chipcom wireless devices, they have been made through the use of the development library supplied by the manufacturer of their development boards. Their design is a FIFO buffer that is emptied when it receives a specified amount of bytes. The number of bytes is configurable and very easy to change.

The hardware in the system will be investigated to assess if the system needs conversions for the sensor inputs and/or the RS-232 communications between different devices.

Schedule

Fig 6-1 **Preliminary Schedule**

Lab week	Date	Task
Break	Break	Amulet Programming. C programming. Winwedge Integration/Design.
1	1/28	Finish Winwedge Integration/Design.
2	2/4	Finish Winwedge Integration/Design.
3	2/11	Finish Winwedge Integration/Design.
4	2/18	Finish Amulet Programming.
5	2/25	Finish Amulet Programming.
6	3/4	Finish Amulet Programming.
7	3/11	Use waveform generators to simulate external data intake
8	3/18	Use waveform generators to simulate external data intake
9	3/25	Finish C coding
10	4/1	Finish C coding
11	4/8	Finish C coding
12	4/15	Test/Troubleshoot entire project.
13	4/22	Test/Troubleshoot entire project.
14	4/29	Test/Troubleshoot entire project.
15	5/6	Final Report
16	5/13	Final Project Presentation

Patents

The information in Fig 7-1 below shows the relevant patents involving wireless data acquisition systems. As all of the other components inside the system are purchased from companies, there are no infringements involving their patents if they are purchased from their respective manufacturers.

Fig 7-1 **Patent Search Information**

Patent #	Description
20090204310	Palm sized wireless data acquisition system for internal combustion engines
20090040034	Wireless data acquisition and on-board display
20080270074	User defined wireless data acquisition

Equipment

Chipcom CC1100 DK wireless boards (2)
SiLabs 8051F120DK MCU
Amulet STK-480272C LCD touchscreen
WinWedge RS-232 software interface

Bibliography

Pieper, Tim. “Wireless Data Acquisition System : Functional Requirements List & Performance Specifications” [online], available from World Wide Web: <http://cegt201.bradley.edu/projects/proj2008/widas/pdf/Functional%20Requirements.pdf>.

Terdiman, Daniel. “In race cars, high-speed data has new meaning” [online], available from World Wide Web: <http://news.cnet.com/In-race-cars,-high-speed-data-has-new-meaning/2100-11389_3-6100399.html>.

SiLabs, 8051F120DK Datasheet.

Chipcom C1100 DK Users Guide.

Amulet Technologies STK-480272C LCD touchscreen Users Guide.

WinWedge32 Standard User Guide.