Portland State University

Viking Motorsports

ECE Capstone 2014

Digital Dashboard

Noah Erickson
Rishal Dass
Chad Thueson
Jaime Rodriguez
Sean Koppenhafer

www.github.com/noahterickson/digitaldash

Abstract

The increased complexity that high voltage systems introduce into a racecar require increased diagnostics to keep the car running safely and correctly. The driver must have the status of all important subsystems visible to clearly see and determine the vital faults. With an updated dash system detailed information can be displayed in real time from the vehicle control unit. The driver can be kept up to date on system functionality both on track or in the pits. The solution to display all this critical information is a small 3.5 LCD screen that will receive information from the subsystems of the vehicle and display them in a simple Graphical User Interface. The interface will not only display the warnings and faults but will also display useful real time information such as motor torque, battery level and amperage draw of the motor.

Contents

1 Introduction

1.1 Purpose

The objective of this unit is to provide the operator of the Viking Motorsports Formula SAE electric vehicle with information about the vehicles operation. The information will be gathered from the on-board vehicle control unit (VCU), deciphered by a main processing unit and displayed on a liquid crystal display (LCD) mounted in the operators view. Mission critical information, such as warnings about battery management systems (BMS), insulation-monitoring devices (IMD) and system temperatures will be displayed on the digital dash.

1.2 Motivation

With the current set-up on the electric vehicle, it using Light Emitting Diodes (LED) that lights up when it detects an error. Although this works, it would be more productive for the driver to know what more information that is causing the error. For example, during a race if the battery is running low, it would be more efficient for the driver to know exactly how much battery life is left rather than know that the battery is low. With this extra bit of data it would be more helpful of the driver to decide if he/she can make it another lap or if they need to make a pit stop.

1.2.1 Vehicle Complexity

The Viking Motorsports team has an internal combustion engine vehicle for Formula SAE races and their new electric car intended for the same purpose was modeled closely to it. With the addition of complex electrical systems it is the desire of the Viking Motorsports team to have this reflected in the aesthetics of the vehicle. The digital dash will integrate with the vehicles systems to display relevant information to the driver and will look good doing so.

Figure 1: Traditional LED Dashboard

1.2.2 Setting a Standard

Another motivation for the digital dash is to guide the future of electric cars created for the purpose of Formula SAE. The Viking Motorsports team would like to have the digital dash be open source so they can share it with other teams that are interested. Perhaps if enough teams adapt to a digital dash the product can evolve over time to support new features outside the scope of this proposal.

Currently the Formula SAE rules require a row of LEDs on the dashboard of any competing electric car.(SOURCE) The LEDs display information about system voltage and IMD status. If the digital dash can be built in such a way that it displays this information constantly and reliably perhaps the LEDs can be removed and the digital dash could be the new standard.

2 Problem Statement

The LCD display must be readable in all weather conditions. A trans-reflective display that is 500-900 nits bright will allow the display to be readable in sunlight. A waterproof enclosure will allow the device to operate in rainy conditions and allow it to be readable in rain. The digital dash unit must be large enough to be readable but small enough to be mountable on the dash of the electric car. A 3.5 LCD screen will allow for both size constraints.

2.1 Project Requirements

2.1.1 Operation

- The unit must have a trans-reflective screen capable of 500-900 nits, for visibility in direct sunlight.
- The digital dash should gather data from the EVCU, BMS and Motor controller and be able to interpret data and display in a graphical interface.
- The displays interface must be simple and clear so driver can easily read at a quick glance; it should include big fonts and symbols with bright colors to differentiate indicators.
- Displayed data does not need to be real-time, but update and refresh at a rate that will be useful.
- The electric car must be independent of display system and run with or without it, in case of failure the electric car must still be operable.
- Display must boot up in less than 30 seconds, so it is not a nuisance to turn on and use.

2.1.2 Power

- The display must be able to be powered by a 12V DC supply, the unit will be powered from the main battery on the electric car.
- A power supply must be designed for the digital dash to be robust against electrical noise and be able to regulate battery voltage (9-13V) down to the display and controller voltage of 5V.

2.1.3 Health & Safety

- Display must be securely bolted to car frame and not stick out of car body to prevent module from flying of and injuring driver.

2.1.4 Environmental

- Must be made with non-toxic/hazardous chemicals that would require proper disposal.

2.1.5 Maintainability

- Must have minimal to no user maintenance over entire life cycle.

2.1.6 Dimensions

- There are constraints to the size of the screen, the electric car dash does not have a lot of space available in the drivers cab so display must be restricted to max of a 4 diagonally, and must have a clear line of sight.

2.1.7 Reliability & Availability

- The display must rugged and water resistant (IP65 standard) as will be exposed to wet weather conditions.

- The display must be easily made to install and detach to various racecar frames.

2.1.8 Price

- Total parts and manufacturing should not exceed $500 as to accommodate the Viking Motorsport budget.

- The digital dash must also be relatively cheap to be a viable open source alternative to similar existing solutions.

3 Solution Space

The digital dash unit will have two inputs and one output. The unit will have two CAN transceivers that take one CAN bus input each, one for the battery management system (BMS) bus and one for the RMS bus. The digital dash will process these inputs and output pertinent information to an LCD.

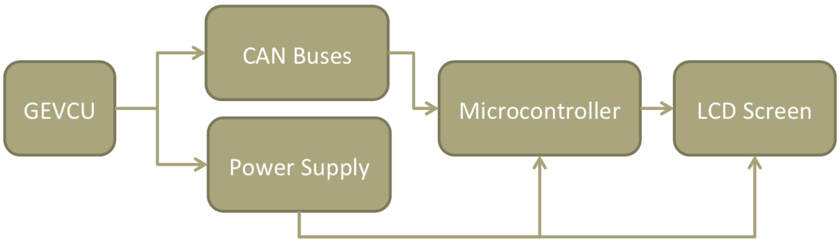

Figure 2: Functional Block Diagram

3.1 Mainboard

The mainboard is functionally a custom Arduino Due that has been reduced in size to the necessary components for the digital dash. The mainboard is a printed circuit board from OSH Park that includes a power supply and two CAN transceivers in addition to the Arduino Due architecture.

3.1.1 Micro-controller

After researching the many micro-controller that could be used the choice came down to four; Atmega328, AT90CAN, Arduino UNO, and Arduino Due. These four (4) micro-controllers that have the capabilities and would be possible to integrate the current system with the new one.
Atmega328:
AT90CAN:
Arduino UNO:
We decided to go with the Arduino Due because the system that is being using with the Viking Motor Sport team was a custom made Arduino Due board, so using another Due would make the integration with there system much easier and cost effective. The open-source electronic prototyping platform, Arduino, was the inspiration for the mainboard. The Arduino Due has an ATSAM3X Atmel micro-controller on it that has CAN capability. The ATSAM3X is a 32-bit, 84-MHz micro-controller that will operate quick enough to handle the messages being transmitted on the CAN bus. The ATSAM3X will acquire the messages sent across the two CAN buses through two CAN transceivers and decide what information is needed to be displayed on the dash.

The main choice for the ATSAM3X was its performance capabilities to keep up with the capacity of messages on the CAN bus. Additionally, the Viking Motorsports Team also uses an Arduino Due, with an ATSAM3X microcontroller, in their eVCU. Basing the digital dash off the Arduino Due helps with compatability issues with the current systems onboard the car.

By using the ATSAM3X the development team was able to quickly prototype the digital dash unit using an Arduino Due which was another major factor in choosing the micro-controller.

	Cost	Fab Board	CAN Controller	CAN Library	SPI, I2C, UART
Atmega328	$\approx$ \$3	Yes	No	No	Yes
AT90CAN	$\approx$ \$8	Yes	Yes	Yes	Yes
Arduino UNO	$\approx$ \$25	No	No	No	Yes
Arduino Due	$\approx$ \$50	No	Yes	Yes	Yes

Table 1: Micro-controller Choices

3.1.2 CAN Transceivers

Two MCP2551 CAN transceivers were chosen for integrating the CAN buses. The main choice for this The CAN transceivers job is to merge two CAN buses together. The BMS and RMS CAN buses have different baud rates and cannot be simply connected to each other. The job of the CAN transceiver is to take the two buses and merge them onto a single bus. The device comes in either a through hole or surface mount package.

	Cost	Max Transfer Rate	Multiple CAN Networks
AMIS-42700	$\approx$ \$6	1 Mbit/sec	Yes (2 networks)
Arduino Due	$\approx$ \$1.50	1 Mbit/sec	No (1 network)

Table 2: CAN Transceiver Choices

3.1.3 LCD Screen

3.2 Power Supply

The power supply was implemented using a TLF1936 voltage regulator and the necessary components to drive it. The power supply was designed to be used on either a 5 volt or 12 volt system for additional compatibility with other systems. For the Viking Motorsports Electric Vehicle we grabbed 12 volts from the on-board power supply.

3.3 Software

The software for the ATSAM3X was written in the Arduino IDE to provide compatibility and ease of use for further implementation by the Viking Motorsports Team.

3.4 Display

The display used is a 4D Systems uLCD-35DT-AR. It was chosen for its 3.5" size, compatibility with Arduino, its touchscreen capabilities and low cost. The display will output the information that is passed to it from the micro-controller and will be able to independently handle all the graphics processing to lessen the load of the ATSAM3X. A polarizing shield will be equipped to the display to help performance in direct

3.5 Graphical User Interface

The graphical user interface(GUI) uses a black background to minimize clutter and has three large icons in the upper portion indicating the insulation monitoring device (IMD) status, the BMS status and the RMS status. Under this there will be a battery meter, a battery amperage draw gauge and a motor torque gauge.

The GUI was designed with a minimalistic approach in order to reduce clutter and to give the digital dash a professional feel. In addition to the simple layout of the GUI each status icon has a sub-menu activated via the touchscreen that can show further information about a system. On the main screen of the GUI there is also a debug sub-menu that will show pertinent temperatures.

3.6 Enclosure

3.7 Communication Protocol

The GEVCU on-board the Viking Motorsports Electric Vehicle must have some integration with the digital dash unit in order to transmit data. There are a number of different protocols that are available to transfer data from the GEVCU to the digital dash unit. A few communication protocols the design team thought about using were: Wi-Fi, I2C, serial and CAN.
Since the GEVCU is Wi-Fi capable, this seems like an obvious choice in order to cut down on physical hardware. Implementing Wi-Fi into the digital dash unit would be simple as picking apart a Wi-Fi Arduino Shield and designing it to fit the custom circuit board, once again cutting down on space. However, Wi-Fi was nixed early on as we thought this would over complicate the design process without much benefit to the project.
Another option would've been serial communication between the GeVCU and the digital dash unit. The ATSAM3X has quite a few serial communication options in the form of UARTs, SPI and CAN interfaces. The design team decided against using UART and SPI because the components on the electric vehicle communicate to the GEVCU using CAN buses.
Choosing CAN as our communication protocol with the GEVCU will allow the design team to hijack data from the current CAN buses. This will allow the digital dash to see all the data that the GEVCU sees and not interfere with mission critical functions during races.

4 Implementation

4.1 Hardware

A custom PCB was fabricated to fit in the limited space allocated for the dash in the vehicle. The layout of the board was designed so the control pins of the LCD would connect directly with the header affixed on the board. Our board was created by modifying the existing Arduino Due design. (Figure 5)

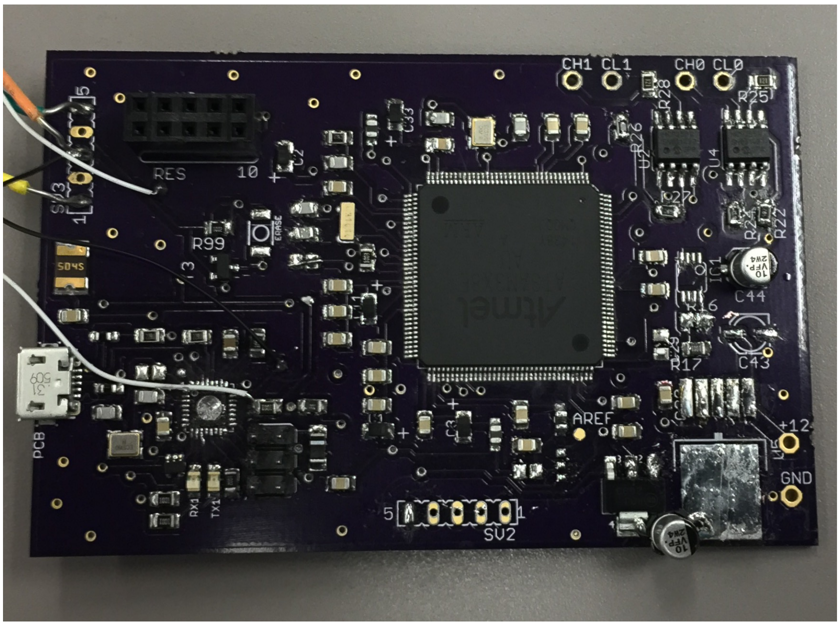

Figure 3: Final Board with Components

4.2 Software

Software implementation goes her

5 Explanations

5.1 Protocols Utilized

5.1.1 CAN Bus

The Controller Area Network or CAN bus is one of several central networking protocols that are used in vehicles that do not have a host computer. Companies are starting to move towards vehicles that are more technologically advanced and equally more environmental friendly. With these advances, all the individual working modules in the vehicle needs to be connected so that they can communicate with each other. For example, the engine reports the vehicles speed

to the transmission, which in turn must tell other modules when to shift gears. To be able to connect all these different modules together to work efficiently and effectively a central networking system becomes necessary.
Having all the modules connected together into one system has numerous benefits. The most significant benefit is the price to performance ratio that it offers. This makes the current CAN bus technology the most reliable and cost effective choice for the automotive companies. If a major event occurs in one of the modules it is shown in the main controller and will allow the rest of the modules to take the proper actions. If one of the modules of the CAN bus happens to fail other modules are not affected unless they are directly related to each other. Diagnostics of the vehicle using the CAN bus becomes easier as well. It can pinpoint what has cause the module to fail and fix that module instead of tearing the whole vehicle apart.

5.1.2 Serial

We used serial to communicate with the display.

5.2 Tools Used

5.2.1 Arduino IDE

The Arduino IDE was used for the data acquisition of the CAN bus and the processing of that data. The data was input using CAN protocol and output to the LCD module using serial.

5.2.2 Eagle CAD

Eagle CAD was used to create the hardware schematic as well as the custom printed circuit board layout.

5.2.3 4DSystems Visi-Genie

The GUI portion of the LCD was created using 4D Systems Visi-Genie software. This IDE integrated nicely with the Arduino IDE with an external library.

5.2.4 Hardware tools

Creating the wiring harness needed some tools such as wire strippers, wire crimpers, multi-meters, oscilloscopes and soldering irons.
Creating the enclosure and the demonstration unit included using the laser cutter in the Electronics Prototyping Lab (EPL) at Portland State University.
Creating the first prototype PCB required the use of the LPKF PCB Router in the EPL.
Creating the final PCB required the use of the reflow oven in the EPL as well as the surface mount soldering station.

6 Testing

6.1 Testing Objectives

6.1.1 Testing Purpose

The following test plan documentation supports the following objectives:

- Identify existing project information and the software and hardware that should be tested.
- List the recommended test requirements (high level).
- Recommend and describe the testing strategies to be employed.
- Identify the required resources and provide an estimate of the test efforts.
- List the deliverable elements of all tests.

6.1.2 Testing Scope

This test plan describes the integration and system tests that will be conducted on the Viking Motorsports digital dash following the integration of subsystems and components identified in the Design Considerations for prototyping document [1].
The purpose of this test plan is to test the feasibility and performance of the Digital Dash. It is critical that all subsystems be tested before integration into

the system so as to ensure a proper user experience when using the digital dash is integrated into racing vehicle.
The following modules and interfaces will be tested:

- CAN message interpretation
- Data transmission to LCD
- GUI on LCD screen

Critical performance measures to test are:

- Minimal time to capture and interpret CAN data
- Time to display data on LCD screen after data interpretation
- Initialization time before the digital dash is ready to use

6.2 Requirements for Test

The lists below identify items (functional requirements, non-functional requirements and use cases) that have been targeted for testing. This listing represents what will be tested.

6.2.1 Function Testing

- Project Requirements: The system must be powered by a 12V source
- Data from CAN network is being relayed to Arduino Due for analysis
- Data from Arduino Due is being sent to LCD
- Data is being properly interpreted and displayed in GUI on LCD

6.2.2 User Interface Testing

- LCD Screen must be viewable in sunlight
- Project Requirements: "The digital dash should be easily readable, big fonts, non-cluttered display."

- The following variables must be displayed on the LCD screen
  - Main battery voltage
  - PM100 temperature
  - Max cell temperature
- The following warnings must be displayed on the LCD screen
  - Main battery voltage low
  - PM100 temperature high
  - Max cell temperature high
- The following errors must be displayed on the LCD screen
  - Battery voltage below limit
  - PM100 temperature above limit
  - Max cell temperature above high
  - BMS error
  - Insulation Monitoring Device
  - PM100 Error

6.2.3 Performance Testing

Project Requirements: "The unit must boot up in less than 30 seconds."

6.2.4 Load testing

Verify the system response when CAN network is heavily loaded.

6.2.5 Stress Testing

None.

6.2.6 Ruggedized Testing

- Requirements Document: "The unit must be weatherproof."
- Ruggedized to survive one season of racing.

6.3 Test Strategy

The test strategy is the recommended approach to test the hardware and software. The previous section described what will be tested. This section describes how it will be tested.

6.3.1 Function Testing

The goal of these tests is to verify proper data acceptance, processing and transmission. This testing is based on black and white box techniques. These tests are to check that modules can handle typical situations.

Test Objective:	Ensure proper data entry, processing and transmission.
Technique:	Execute each use case, case flow, or function, data, to verify the expected results occur when valid data is used.
Completion Criteria:	All planned tests have been executed. All identified defects have been addressed.
Special Considerations:	N/A

Table 3: Function Testing

6.3.2 User Interface Testing

The goal of these tests is to verify proper data acceptance, processing and transmission. This testing is based on black and white box techniques. These tests are to check that modules can handle typical situations.

Test Objective:	Verify the following: Ensure the LCD screen displays information properly on the screen and is visible to the driver. Verify that screen is readable in sunlight.
Technique:	Predefined messages are broadcast over the CAN bus and the corresponding information should be displayed onto the screen. Take the unit outside and see if screen is readable.
Completion Criteria:	All user interface modules functioning properly. All identified defects have been addressed.
Special Considerations:	If visibility of screen in sunlight is sub-par a new type of screen will need to be sourced and all interface testing will have to be repeated.

Table 4: User Interface Testing

6.3.3 Performance Testing

Performance testing measures the transaction time and other time sensitive requirements. The goal of of performance testing is to tune the system to ensure that interaction with the system is not excessively slow and doesn't use excessive energy.

Test Objective:	Validate system response time for designated transactions.
Technique:	Measure the time it takes to perform predetermined operations to ensure they complete within a satisfactory time range.
Completion Criteria:	Completion of operation is found to be satisfactory based on predetermined time range.
Special Considerations:	N/A

Table 5: Performance Testing

6.3.4 Stress Testing

This section is not applicable to test the Digital Dash prototype.

6.3.5 Load Testing

Load testing measures the ability of the system to function properly under different workloads. The goal of the load testing is to determine and ensure the system functions properly beyond the expected maximum workload.

Test Objective:	Validate system response time for designated operations.
Technique:	Saturate the CAN bus with data to determine if data can properly be captured and processed.
Completion Criteria:	Data still properly displayed on screen without system locking up or becoming unusable.
Special Considerations:	N/A

Table 6: Load Testing

6.3.6 Ruggedness Testing

Ruggedness testing verifies the operation of the unit while meeting Formula SAE Rules requirements and sponsor design requirements.

Test Objective:	Verify unit adheres to requirements of Formula SAE article 7 EV7.33. Ensure meets any additional ruggedness requirements.
Technique:	Physical tests to ensure that enclosure can function after being exposed to water for 240 seconds.
Completion Criteria:	System is still operational after all tests have been completed.
Special Considerations:	Exposed to water means rain like conditions not immersion.

Table 7: Ruggedness Testing

6.4 Testing Equipment

Equipment needed to carry out testing:

- Windows compatible PC
- Viking Motorsports electric race car
- EVCU Programming Software
- Spray hose

6.5 Resources

Role	Minimum Resources Recommended	Specific responsibilities/ Comments
Test Designer	Chad Thueson	Identifies, prioritizes, and implements test cases.
System Testers	Chad Thueson, Sean Koppenhaffer, Jaime Rodriguez, Noah Erickson, Rishal Dass	Executes tests, log results, recover from erros, document defects
Designer	Chad Thueson	Identifies and defines the operations, attributes, and associations of the test classes.
Implementer	Chad Thueson	Implements and unit tests the test classes and test packages.

Table 8: Resources

6.6 Results

Test good.

7 Conclusion

The digital dashboard performs as intended when simulated with the demo program that we created. Instead of testing all functionally of the screen, the demo program focuses on only the most important functions.
Due to circumstances outside of our control, the digital dashboard has not been officially tested on the electric vehicle but we are confident that it would function correctly.

8 LICENSING INFORMATION

References

[Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.