MetroSwift - Oregon State University
2013 Cornell Cup Final Report

Ryan Kalb
Steven Reid
Patrick Simonson
David Taylor
Advisor: D. Kevin McGrath

Table of Contents:

1. Challenge Definition 4
  Proposed Solution 5
2. Project Solution Updated 7
  Customer Value Proposition 7
  Solution Changes 8
  Summary 9
  Design Decisions 10
3. Product Performance Evaluation 13
  Performance Measures 13
  Performance Discussion 13
4. Technical Documentation 15
  Procedure 15
  Key Diagrams 16
  Assembly and Construction 17
  Test Data 18
  Test Results 20
5. Project Execution Performance Evaluation 22
  Timeline 22
    Original Timeline 22
    Timeline Adjustments 22
    Revised Timeline 22
  Budget and Expenditure Justification 23
  Mid-Project Reviews 23
  Process Understanding 24
6. Recommendations and Next Steps 25
  Further Improvements 25
  Necessary Resources 26
7. Glossary 27
8. References 28

9. Acknowledgments....29
10. Survey Results 30
Interview Questions 30
Interview Process....30
Answers to Research Questions 31
Other Insights 32
11. Appendix....35
Appendix A: Button Layout Example Test....35
Appendix B: List of Materials 37
Appendix C: Kontron Data Sheet....37
Appendix D: TFT LCD Monitor....37

1. Challenge Definition

Vehicles are getting smarter, with an increase of on-board controls and options which improve the driver's interaction and experience in their vehicle, but at the same time, this increases distractions. A driver's main purpose in a vehicle is transportation and safely arriving at the destination. Our plan is twofold: improve the driver's safety by decreasing the number of distractions while not decreasing functionality, and improving the driver's overall experience. We plan to do this by moving the center console controls (climate, entertainment, critical information etc.) to the steering wheel and the front dash board gauge cluster.

Distractions in vehicles are on the rise, taking the driver's attention away from the road and spreading it to different areas of the vehicle. There is a discontinuity between the components of a vehicle's interface, split between the dash and center console. There is yet to be a central interface close to the driver which has control over the whole vehicle's interface, though many of the vehicles features are starting to migrate to the steering wheel.

On one end of the spectrum you have basic vehicles interfaces which provide the driver with simple, concise controls that are intuitive and generally more tactile. These controls usually include knobs and dials for adjusting various settings, such as volume, climate, etc. and are located in the center console and dashboard. The driver's dashboard usually only contains basic vehicle information, such as fuel levels, temperature, mileage, speed and information from in-car sensors. These are all easy to read and intuitive to use, but are not very complex.

One the other end of the spectrum are high end vehicle manufacturers, such as Tesla, who provide more complex interfaces. While innovative, this interface may distract the driver away from the road. Increasingly, our society has seen a large push towards touchscreen devices, and this is seen in new vehicles. These interfaces remove the tactile dials and knobs and replace them with large touch screen sensors and voice recognition. We acknowledge this is an improvement in modern technology; however there is a decrease in safety with the increase in distractions. These interfaces take the driver's attention away from the road, forcing them to focus on manipulating touch screen controls. For example, in order to change the temperature or increase the volume of the radio, the driver must shift their focus away from the road and onto multiple surfaces. There is no universal layout for vehicle controls; some are located on the steering wheel, the center dash control or even the doors.

To unify these controls, vehicle manufacturers are building complex touch screen interfaces which isolate these controls for easy adjustments. However these interfaces lack any tactile feedback, and require increased focus over their basic vehicle counterparts, which also prevents any muscle memory for where buttons are located.

We do acknowledge that voice control does solve a lot of these issues presented, but it can be frustrating or even impossible to use with vehicle noise, passenger conversations, etc. In addition, would you really want to have a conversation with your vehicle to increase the volume or decrease the temperature? This is only feasible if the driver is the only person in the vehicle, otherwise it could be quite troublesome and not intuitive or universal.

Vehicle and road safety are some of the biggest concerns of modern society. Currently census data shows that over 10 million road accidents happen each year, and of those 35000 are fatal [2]. The main factors when it comes to vehicle safety are human error and distractions. Taking focus off the road and onto the vehicle's interfaces has been proven to lead to accidents. There needs to be an interface which allows the user to focus on the road while also being able to control all the critical features of the vehicle.

Another factor when it comes to fatal accidents is speed. Currently one of the only deterrents for speeding is possibility of getting a speeding ticket, but this is usually overlooked or ignored. The driver should have an additional incentive to encourage them to observe the speed limit, and better methods to alert them when they are exceeding it.

Proposed Solution

Our original proposed solution addresses these challenges we have identified above. First we are overhauling the dashboard and car interface system. The dashboard display gauge cluster will be a single unified screen that replaces the typical gauge cluster. The interaction with this interface will be via 6 OLED buttons easily accessible on the steering wheel.

One of the key features we have implemented in the dashboard is a new speedometer gauge. Current systems have either an analog system (a gauge needle that informs the driver of their speed) or a digital display (a numerical value informing the driver). We have incorporated both. There is a set of 2 speed needles and a set of 2 digital numerical values informing the driver of their current speed and the current speed limit of the road they are driving. When the driver begins approaching and exceeding the given road speed limit, the analogue speed dial will begin glowing yellow, then red, warning the driver that they are exceeding the speed limit. This new speed dial will enable the driver to understand how fast they are going and hopefully help them go at or even slightly less than the speed limit. This will not only help the driver drive safer, but also improve the efficiency of their driving. This key feature improves the safety of the driving experience by encouraging them to drive at the speed limit.

Another key feature is the overhauled gauge cluster. This is the biggest part of our user interface. The driver's gauge cluster will also be the primary interaction with the vehicle's various components (climate control, entertainment etc.). The gauge cluster is going to be split into 3 components as illustrated in figure 1:

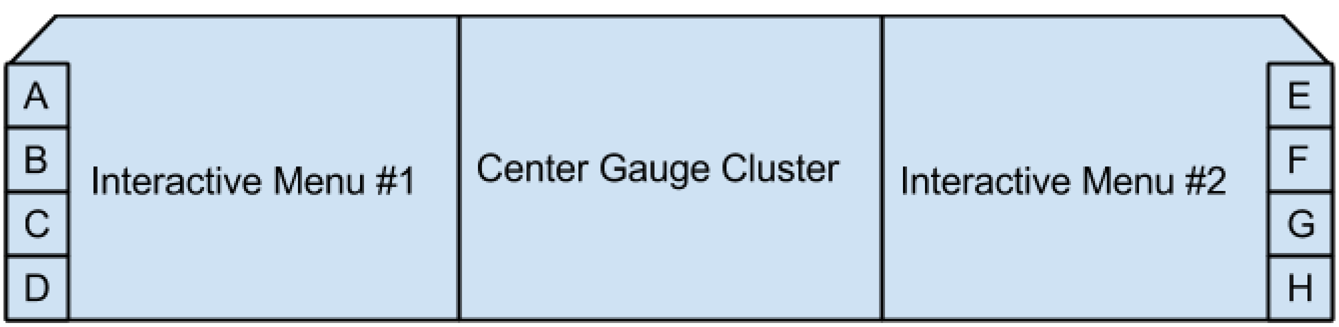

Figure 1: This shows the basic layout of our UI. The 2 interactive menus show where the driver will be able to make changes to the vehicle's systems. The Center Gauge Cluster will be stationary and will always be visible. A-H coincide with the 6 OLED buttons and 2 rocker switches.

The driver will be able to interact with these menus using 6 OLED buttons mounted on the steering wheel. Many vehicles already have buttons mounted on the steering wheel for cruise control etc. so this isn't a new concept. We are pushing the concept to the extreme by having the 6 OLED buttons controlling everything we need. Additionally we have 2 rocker switches mounted on the steering wheel for controlling things like volume or radio presets etc. We have basically removed the entire center controls (for climate, volume, radio etc.) and moved those controls to the steering wheel and center gauge cluster. The reason we decided to do this is because people get distracted with the center console controls, and especially with touch screens, they have to look down off the road to make adjustments. Our idea is that if the user does need to make adjustments, they use the tactile buttons, and only need to look down slightly at the gauge cluster, which is a lot closer to the view of the road than the center console. The buttons will also have images corresponding to the images on the screen. This should help improve safety as well as give the driver a better experience.

Another feature we are implementing is the "gamification" of the driver experience. We want to make the driver experience enjoyable and safe. As previously mentioned, our speedometer warns the driver when they are going over the speed limit. Studies have shown that exceeding 45-65 mph actually decreases the efficiency of typical sedan vehicles; this range gives you the optimal fuel mileage [2]. To encourage the driver to travel at the correct speed, the driver will be awarded a score for each driving session based off of their driving habits, based on speed, fuel efficiency, torque, throttle position, etc.

As seen in figure 2, this is our original concept for our proposed solution.

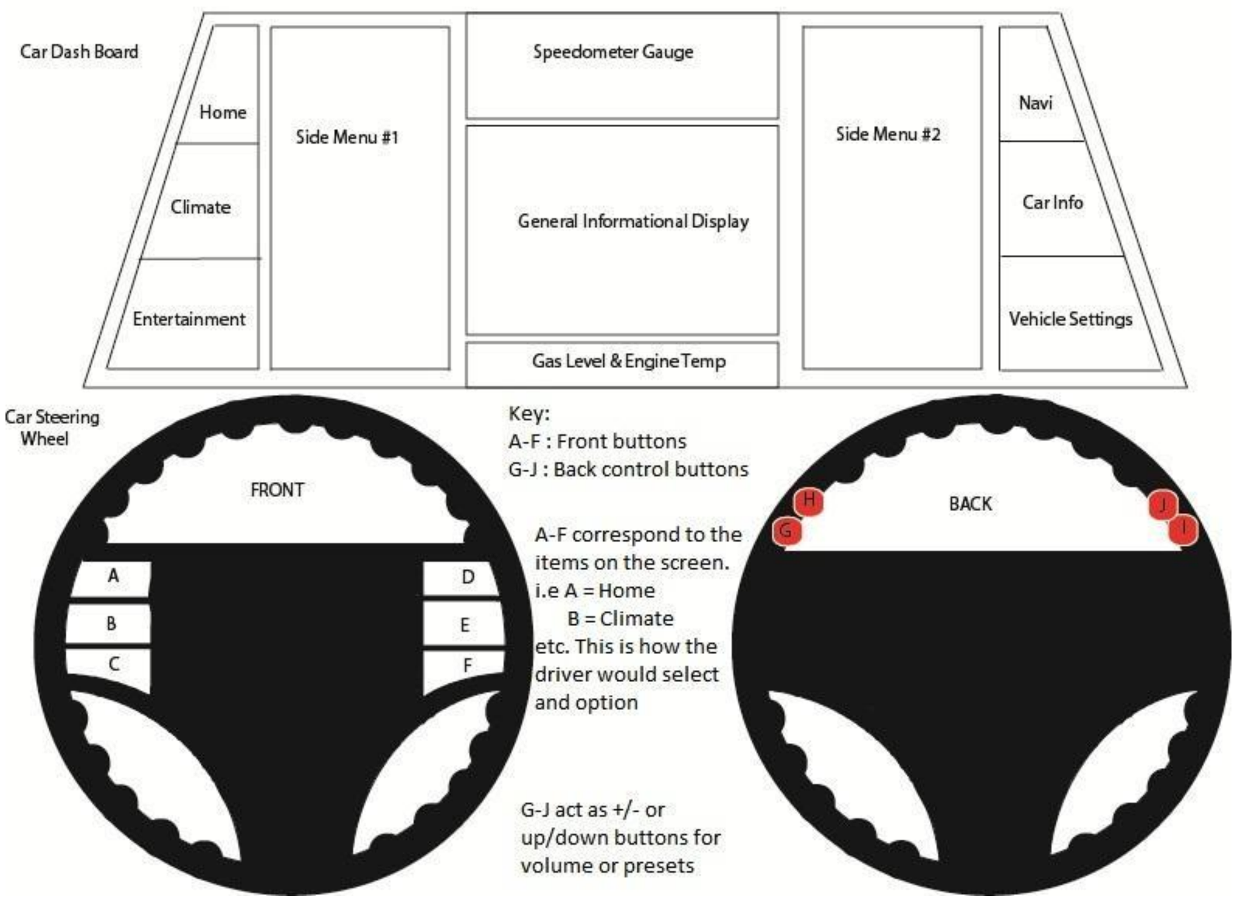

Figure 2: This was our concept for how our solution would be implemented.

2. Project Solution Updated

We have had many setbacks, delays, problems and issues with our project, and with any project, changes need to be made. We have made a number of changes from our original concept but still focused on our 2 key features: the driver experience and safety.

Customer Value Proposition

Our challenge for this project is to improve safety, by limiting distractions, while at the same time enhancing the driver experience.

For our enhanced design for the speedometer, we implemented a safety mechanism for the driver. As the driver begins to exceed the designated speed limit, the speedometer will glow yellow then red. Travelling above the speed limit increases the chance of a speed related accident. Our design in figure 3 shows the progression of exceeding the speed limit:

Figure 3: This figure shows how our enhanced speedometer works. When the driver is below the speed limit, there is no glow. Once they begin exceeding the limit, the glow begins to show, starting at yellow, becoming darker and darker, also indicating that the driving is breaking the speed limit.

Another aspect of safety is distractions. As earlier discussed, to improve safety, we will decrease distractions. If a driver has to look away from the road, it increases the chance that something could happen and they won't have time to react, or they won't be able to see oncoming traffic or obstructions. Since we are moving all the center controls to the steering wheel and to the gauge cluster dashboard display, the driver's vision will be significantly closer to the road. Since there is an increased distraction factor in touch screens, we have gone with tactile OLED buttons which gives the user feedback on the screen.

The other challenge we are focusing on is enhancing the driver experience. The challenge has been approached from various angles in modern vehicles; from internet radio to touch screen interfaces and smartphone integration. We've gone with a different approach: the user interface and vehicle interaction. First, we have digitized the gauge cluster while also keeping the classic analogue dial look. Secondly we have implemented a dynamic menu system for the user to interact with the vehicles various controls and settings. All of these menus are controlled by OLED buttons mounted on the steering wheel. These OLED buttons have corresponding images to the current menu they are manipulating. Everything the driver can control and change is at the driver's fingertips.

Another aspect to the driver experience is incentivizing the driver to drive safer. People are incentivized to drive more conservatively to increase their fuel mileage, and we have gone a step further by gamifying the experience. After each driving session, the driver will get a score based on how well and how efficiently they have driven. The higher the score, the better. Not only does this help the driver save money on fuel, it also gives them a different and more fun driving experience.

All of the features that are key to our project focus on the two key challenges of our project, safety and improving the driver experience.

Solution Changes

As with any challenging project, changes need to be made along the way, and this was no exception.

One of the first design changes we made was to the speedometer. Originally our design was to have a speedometer that would adjust based on the current speed zone, and would have 3 colored regions, one for the speed region that is safe or within the current road speed, one for above the speed limit but not considered breaking the law, and another region for dangerous and unlawful. Each region would change based on the current speed of the road. This changed to the design described earlier, with a speedometer that glows based on the speed limit. We decided to make this design change because it was significantly easier to implement, as well as it looks more appealing to the driver.

Another big design change was how we were going to prototype the vehicle integration. Our original idea was to actually mount this in a vehicle, and gather all the necessary information for the gauge cluster (speed, torque, fuel levels etc.) from the CAN bus protocols on the vehicle. We decided to change this from an actual vehicle to a simulator for many reasons. One of the main reasons was transportation. It was not feasible to transport the vehicle to Disney World for the Cornell Cup Competition. The simulator we decided to use is an open source simulator called Racer, which outputs the data over UDP to our system. We were able to fully integrate the steering wheel and add a pedal system to the simulator.

Another area where we made design changes is to the general scope of the project. Initially we wanted to implement a lot of vehicle features. This included navigation, rear end reverse cameras, climate controls, entertainment with internet radio support as well as an additional smartphone application. We did implement a basic version of the audio system that would act as a radio or mp3 device that would stream audio through our system. The functionality is built; it would just need to be developed further to include things like internet radio etc. We determined the scope of this project to eliminate things that would either be far too difficult for this project or impossible given the amount of time. Things such as navigation would not be applicable since we are using a simulator. Furthermore, good navigation systems have already been implemented. So we narrowed down our solution to include the fully functional menu system as if the features were part of the scope of this project. All the menus of the user interface are fully integrated and accessible, but some of them do not have real application, such as navigation. We didn't implement some of these components because a lot of these features have been implemented before, and therefore wouldn't be applicable to the novel idea of the project.

A key component to our project is the display. The initial plan for the display changed numerous times. We first designed the system around a total resolution of 2400x600. The two options were three 800x600 resolution screens or one large screen at the full resolution. We contacted many companies including Tesla about purchasing screens, but were unable to purchase a single screen at the desired resolution. The problem we faced with outputting to three screens was the hardware did not support output to all three simultaneously. After investigating actual vehicle dashboards, we decided a smaller resolution of 1280x480 at 12.2 inches would be the next best option. Since we had originally designed our user interface with a 2400x600 resolution screen in mind, we had to rework our user interface to work with a 1280x480 screen. We made the adjustment because the size and resolution didn't exist together. Ultimately we found that the resultant resolution performed very well for our purposes.

Due to time constraints and not having access to the hardware provided for the Cornell Cup for a while, our initial plan was to use a similar Kontron based hardware system. We also didn't expect to be using a simulator, so we are using one our laptops as the sole system to run the simulator. This is a high performance gaming laptop. This simulator system now feeds our Intel provided hardware the necessary information required for our system, via UDP. We are using the Kontron board to provide output to the screen panel, as the Cornell hardware doesn't support output over LVDS. However, if the screen supported VGA/HDMI or a proper adapter available, the entire dashboard would run on the Cornell hardware.

One of regrettable but necessary changes we had to make to our design was to testing. Originally we were attempting to implement this design in an actual vehicle, which we would be able test the user friendliness of our new system. We had originally planned to have a testing situation and have the driver interact with the system to determine how easy it is to change settings while keeping their eyes on the road. Since our project was to decrease the distractions of the driver experience, and once we decided to use a simulator, it became significantly more difficult to test our solution to prove that it is a better alternative to the existing solution. We used the simulator as a basis for testing in order to validate our idea, but further testing in an automobile is necessary.

Summary

Our project has gone through many iterations and changes. One of the biggest limitations to our project is time. When we started this project, we had a huge list of features. We included everything including entertainment system, full climate controls, navigation, rear view reverse cameras etc. From all these features we had to prioritize and determine what features of the initial list were worth implementing. We chose features which solved the two challenges focused on: safety and the driver experience. The new speedometer design focuses mainly on safety, the integration of the OLED buttons to the steering wheel limits the distractions of looking away from the road ergo improving safety, while the new user interface menu system which could support all the necessary features that we originally outline, improves the driver experience.

We initially aimed high, including features that we thought would push the boundaries of our system and pushing our time limits, but we included them because those features would be something that would be rather interesting to include and would improve the system as a whole. Plus we would like to eventually integrate this system into a real vehicle, and use GPS data for navigation and our speedometer system, as well as implement a full interactive entertainment system, rear view reverse cameras and have a fully functional climate control system. Given our time constraint, we needed to make tradeoffs between the priority of the feature and the time available to implement.

Design Decisions

Center for Critical Info

The center display allows the user to read critical information while keeping their head facing the road. Critical information includes the speedometer and an alert system. Since the overall car interface has been simplified immensely, the information in this display will stand out while not distracting the user.

Screen for Driver to Control Everything (in front of the driver)

This makes it easily accessible, and rids the center console of inputs. An interface in front of the driver keeps the controls exclusive, rather than spreading them out and dividing the driver's attention among different areas of the car. With the controls in front of the driver, he or she doesn't need to reach across anything else in the vehicle to make changes.

Separate Screen for Passenger

Allowing a screen for the passenger and driver gives the opportunity for the driver to leave car peripherals up to the passenger, and limit what must be focused while driving. If GPS is used, the passenger can read directions out without having to get in the driver's space, and while most functions and browsing will be distinct to each screen, some things will be synced such as what song is playing, climate information, etc.

Toggle Control

Menu items can be toggled through to select the preferred mode. For instance, when selecting the climate fan mode, one may toggle through the options of "head," "feet," and "head and feet." We found this was more intuitive for users during a usability evaluation than having a menu for "more options" where every option is visible.

Low Depth Menus

Users don't want to spend time scouring menus. Rather, tasks should be accomplished easily and attention returned to driving. Depth of the menus has been balanced with the number of available menus. Our testing found that users prefer low depth menus, with similar, mutually exclusive options accessible through a toggle. This was faster and more intuitive than deep menus with all the available options visible.

Physical Buttons

Users have indicated a preference for tactility while interacting with their car. Buttons are already familiar for car users, and it is less likely for the car to misinterpret a button-issued command than or it to misinterpret a swipe or a voice command.

Six Buttons and Two Rocker Switches

Because buttons are mapped to commands, the interface needs to balance number of submenus with too many buttons cluttering up the steering wheel. We felt 6 buttons and 2 rockers to be an appropriate amount. It gives the range of options required for the vehicle and maintains the simplicity of the interface. This worked well in our testing.

Horizontal Bars

Bars for information like those for the temperature or fan speed are horizontal. In our testing, vertically oriented scroll bars created an association with the left and right rocker buttons. This was okay for the left rocker, as it changes based on the context, but the right rocker is anchored to volume control. This resulted in users assuming the right rocker matched the right bar on screen. By orienting the scroll bars horizontally we break the chance of this association forming.

OLED Buttons

Users had difficulty matching the buttons on the keyboard to the onscreen button they control. By having the icon on the buttons change to match the icon on the screen, users will be more able to find and press the correct button. Having the buttons be OLED or some screen technology will allow the icon to change. It is worth noting that the buttons will still be physical buttons that the user must depress, not touch screens.

Figure 4 is a concept drawing of what our system intends to look like with the OLED and steering wheel integration interacting with the display unit. We also completed a Solidworks design for our display unit, concept design of what the display would look like [See file DashConceptDesign.pdf].

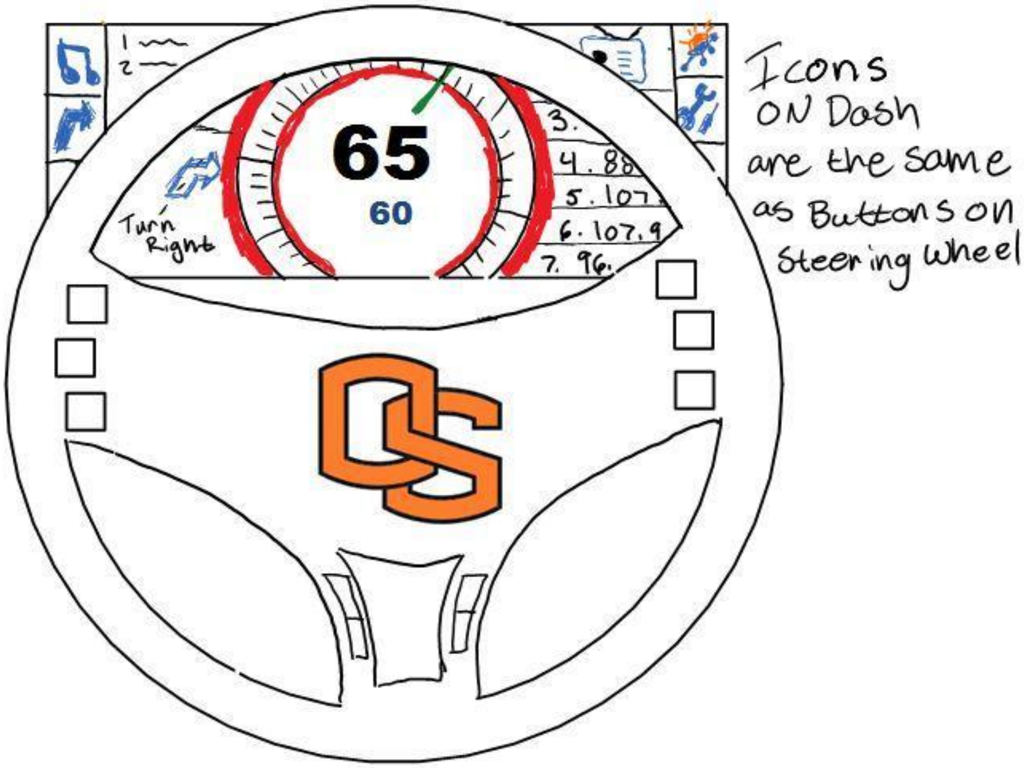

Figure 4: This is a concept drawing of the driver's point of view as if they were to interact and use our system.

3. Product Performance Evaluation

Performance Measures

The key challenges we wanted to focus on, as mentioned earlier, were safety and the driver experience. To test whether we were improving safety, one of the things we focused on was eye tracking. We were aiming to reduce the distance the eyes travelled from focusing on the road to adjusting settings in the car. If we found that people's eye line focused closer to the road and there was less time spent looking away from the road, since the gauge cluster is in the peripheral view of the driver, then we knew we had met and completed that challenge of safety. Another aspect of safety was the speed limit indicator. Our speedometer was designed to deter people from going over the speed limit. If were able to discourage people from exceeding the current speed limit by using yellow and red glowing indicators, then we feel we have completed that challenge of safety.

When it comes to the driver experience, it is a subjective and hard to test metric. We believe that this requires a lot of surveying of individuals using ours system, and deciding for themselves if having all their controls on the steering wheel, with all the information on the gauge cluster menus is a good idea. If we get positive feedback from an array of drivers, telling us that they feel comfortable and would like this system in their vehicle, then we feel we are successful in improving the driver experience.

Another aspect of the driver experience was incentivizing and gamifying the experience. Since each driving session would be awarded a score on how well the driver drove, with live updates as they drove on their driving abilities, we would hope that not only would they would enjoy the experience, they would also try to drive more conservatively, to help the environment. If this system were in a vehicle of a household, it might become a competition to see who could drive the best in family. Not only would this make driving fun, but it would also help encourage safer driving.

Since safety and the driver experience were the main challenges we were focusing on for this project, if we could improve both these aspects with positive testing results, then we feel we have succeeded in completing this project.

Performance Discussion

Though our testing was limited, and we wanted to do more of it, we were able to get a lot of positive feedback about our system, and a significant amount of people who we interviewed would want to see this system in their own vehicles. One thing that is noteworthy is that even though we only gave users a short amount of time to interact and become familiar with the system, in particular using the OLED buttons, we were able to see a large improvement in their ability to interact with it without making errors or getting frustrated with the controls. Users were able to quickly and intuitively navigate the menu systems in just a short amount of time. We were also glad to see that many users did not rely on looking down at the buttons to see what menu to interact with which was one of main goals of using the OLED buttons.

One of things we really want to test is user's eye tracking when interacting and using our system. In order to do that we would need to integrate this into a real vehicle and compare it to other vehicles. One problem we faced with our current simulator is that it is designed to be a game. The observed that people interacted with the system more like a game than an actual vehicle. Ideally, we would integrate this into a real vehicle, and use eye tracking and ask users to change various settings while driving the vehicle. We would look at how much the driver's eyes move away from the road, and compare these results to other vehicles on the road that use a standard dash gauge cluster with a center control panel. We would also want to test this against vehicles with touch screen based units. Positive results in eye tracking would give us the greatest justification that our system is superior to other systems on the road.

4. Technical Documentation

Procedure

The first step in our project was to decide on the framework we would use to design and build our user interface. Since the majority of our project was redesigning the UI of a vehicle, we were advised to use Qt as the framework, since it has great graphical user interface tools. From here we discovered QML which really enabled us to build our interface. Built on top of this QML we used C++ to describe the actual logic in our UI. The C++ was our primary language for reading the simulator data, parsing and interpreting it. This was the majority of the project. Below is a detailed description of how the tools listed above were used.

Qt Framework

We selected the Qt framework for C++ as it provides us a way to rapidly develop complex and functional user interfaces that run across Windows, Linux, and Mac platforms. This cross platform capability gives the project a lot of flexibility and allows for future deployment on any platform.

QML/C++

Qt Quick is a software framework inside of the Qt framework which allows for the design of graphical interfaces. Qt Quick has a declarative scripting language called QML (Qt markup language). We have designed an interface which uses both QML and C++ to provide a graphical interface and complex logic. All of the graphics are written in QML and the application logic is split between QML and C++.

Menu Framework

The menu system was developed to be a flexible framework that will accommodate a wide range of menus, and allow each menu module to be fully self-contained. The menus are each a separate QML module, and have a configuration file which the main program uses to load and unload the menu. Menus can easily be switched in and out without having to recompile the program. Thus, a separate menu system could be swapped in by changing one line in the configuration file. This allows for lightweight updates and modular code. A similar philosophy was implemented for various aspects of the system, such as button icons, general car information, and the look of the speedometer.

Simulator

The Racer simulator used in the project features a number of methods for scripting and adding functionality to the program. We created a script that runs on every frame drawn, fetching various car parameters, and outputs them to the debug console. This debug information is then sent via UDP (a native option in Racer) to the system running the dashboard. The dashboard listens on a socket to the incoming UDP packets, filters the messages for the specified message format (since all debug information is sent over the network), and parses them to update the GUI.

The following types of car data are being obtained from the simulator at this time:

- speed
- rpm
- throttle
- current gear
- braking amount

We have an option in place in our Racer script to adjust the rate of messages being generated. While it can generate a message every frame, and the dashboard software can handle it well, updating too frequently made it more difficult to read, and generated unnecessary network traffic.

Key Diagrams

As mentioned above, our project is composed of many entities and all connect to the central goals - safety and the driver experience. As shown below, this was our first block diagram, in Figure 5a and our modified, updated block diagram as seen in Figure 5b.

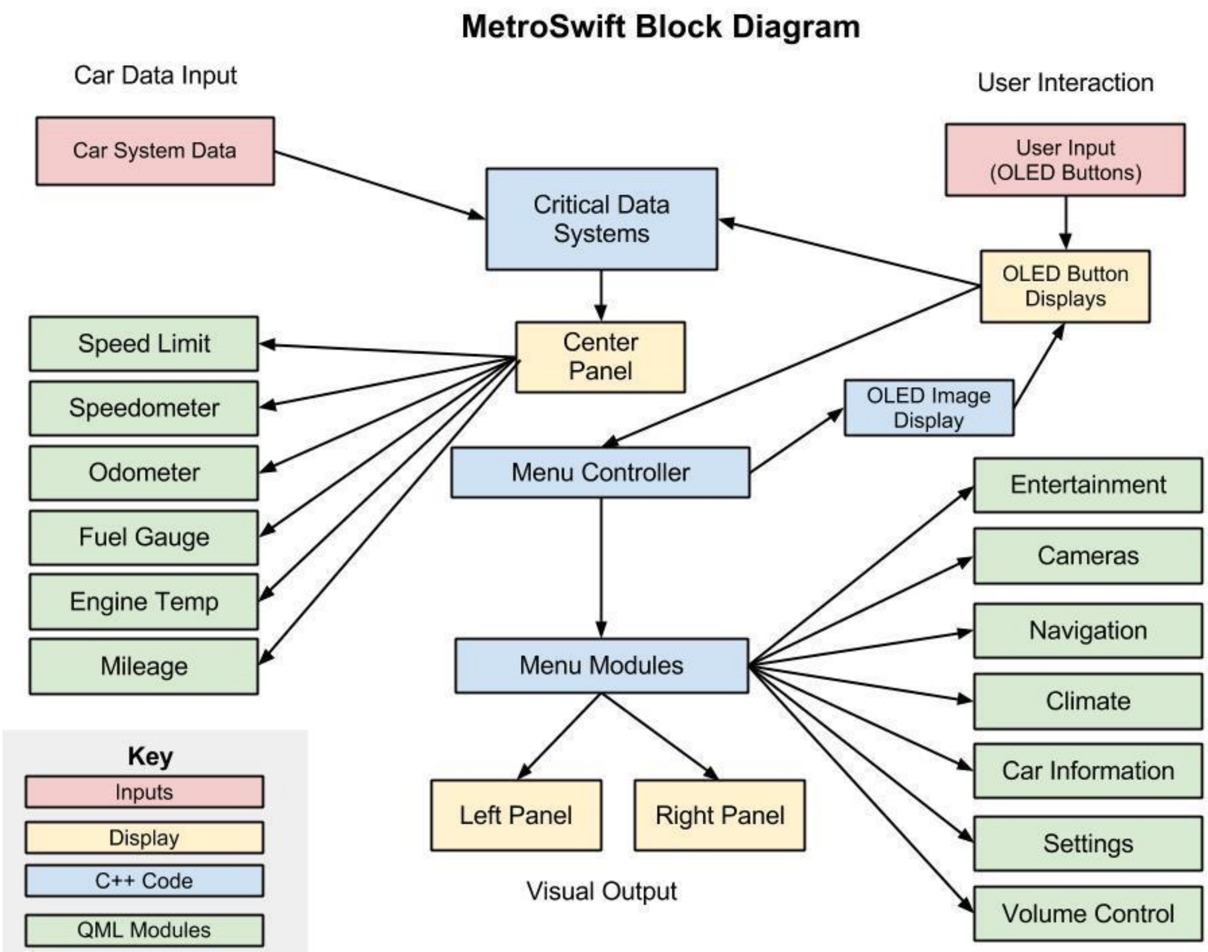

Figure 5a: This image shows the original theorized block diagram of the high level interactions between all the components of our system.

Since our system has undergone many changes including narrowing our scope, the interactions between the various components of the system have changed quite significantly. As seen in Figure 5b. There were a lot of changes we decided to make, as mentioned later, and these changes are reflected in our new block diagram.

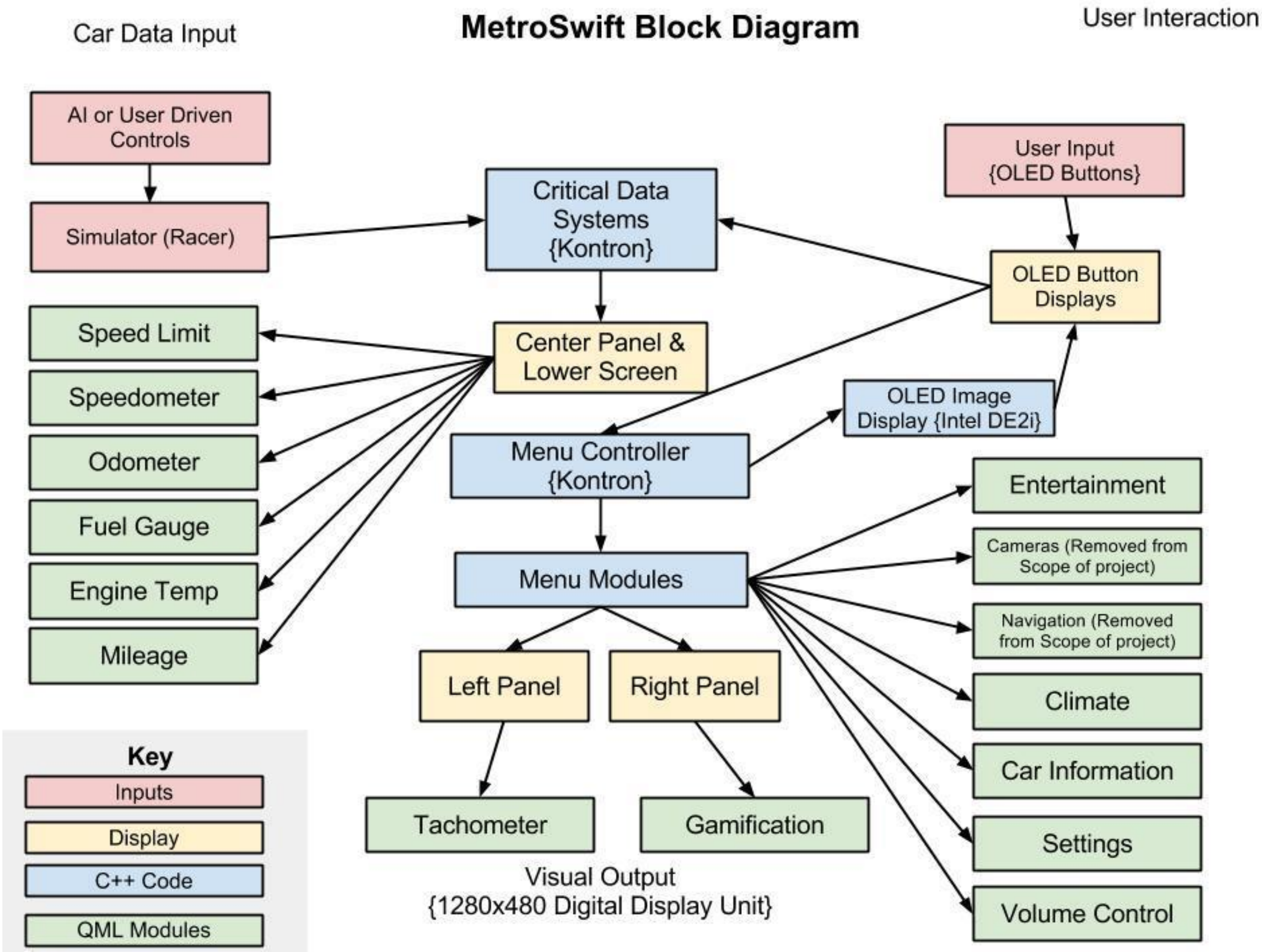

Figure 5b: This block diagram shows how our system currently interacts, including all hardware components and changes we made to our scope.

Assembly and Construction

There are 5 main connections and interactions in our system: We have the interaction between the steering wheel and the simulator, the simulator and the Kontron hardware, the Kontron and the display unit, the OLED buttons with the user interface on the Kontron and the OLED buttons with the DE2i.

For the connection between the Kontron and the Sharp display unit we are using a 4-channel LVDS output from Kontron connected to the LVDS Sharp display. There is a back light inverter connected to a constant 12v line, outputted from the Kontron's power distribution board.

The steering, with OLED buttons, is connected to a Logitech force feedback GT steering wheel integrated into a Acura 2007 Touring Edition MDX steering wheel via a custom steering wheel adapter designed and built by the MetroSwift team [As seen in the included file SteeringWheelAdapter.pdf]. The OLED buttons will be mounted on the steering wheel with PCBs, and surrounded by a fiberglass molded bezel.

Our simulator runs off of an Asus G53SX Laptop and the live simulation data is sent to the Kontron board over a UDP which our user interface reads and displays on the Sharp display unit.

Currently we are anticipating having the OLED buttons interact with the DE2i hardware as a slave on the SPI bus. We have been using the Wunderboard as a development tool since the SPI bus is fully functional. The DE2i is used as our output for the images on the OLED buttons.

The OLED buttons will interact with the menu system over a keyboard adapter. Currently in construction, we plan to use a PS/2 numpad and integrated each OLED button such that the buttons act as keyboard inputs. As an alternative, we may use the Wunderboard, and integrate them via the I/O pins and program the Wunderboard as a keyboard.

As of the report writing, the OLED system is still in development, but we anticipate having it complete or mostly complete in time for the event.

Test Data

One of the things that we did a significant amount of testing with is with the button layout for our OLED buttons on the steering wheel. Since we had 3 OLED buttons on each side of the steering wheel, left and right, we thought it would be a good idea to test the layouts of those buttons for people to feel most comfortable with. In order to do this we tested many different designs. In order to test this, we used 2 usb numpads. We oriented the 3 buttons in different layouts corresponding to layouts on the steering wheel. Figure 5 shows the various layouts we tested.

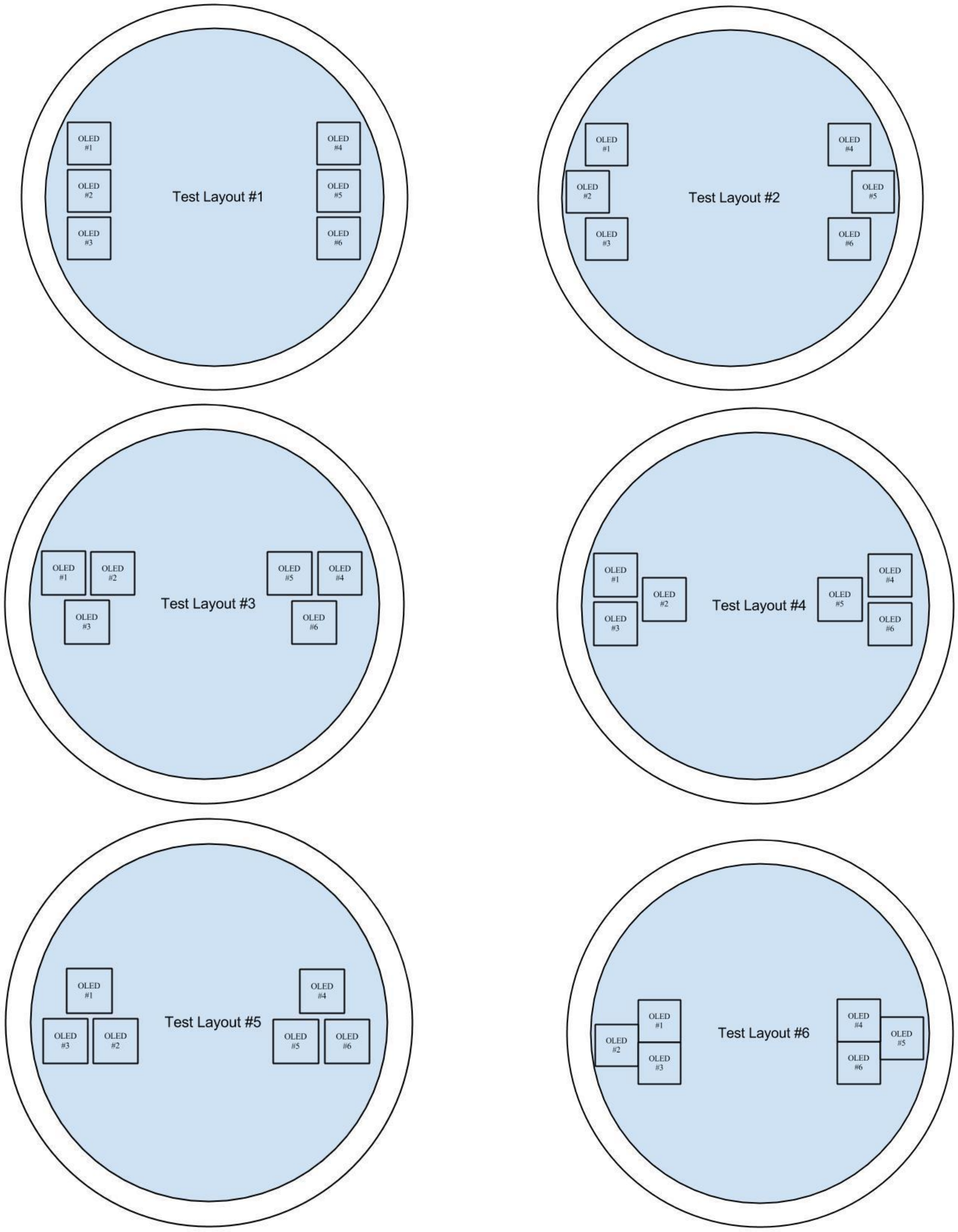

Figure 5: This shows the multiple layouts of the buttons we tested on various drivers in order to decide how to layout the buttons on the steering wheel.

We then had multiple people test these layouts on a version of our user interface. We tested quickness, how easily they were able to learn the corresponding buttons to the on screen layout and how comfortable the button layout was. We tested various experienced drivers, from teenagers to young adults to elderly drivers. From this data we were able to discern which button layouts on the steering wheel would be the best. We had 2 major layouts that we narrowed it down to. Due to limited size on our steering wheel we had to go with the secondary layout.

We had to test how we would align our buttons for the driver to be most comfortable, and easy enough for the driver to make changes without straining their hands. We tested this on various drivers with significant driving experience, new drivers, and soon to become drivers. We averaged how long it took all the drivers to complete each exercise to determine the best layout. We want to limit the distraction, so the less time the driver has to take to make an adjustment, the better. These are the various orientations that we tested for our button placement:
TEST LAYOUT #1: This orientation has the button alignment being purely horizontal, matching the alignment of that on the screen.

TEST LAYOUT #2: This orientation below keeps the horizontal alignment while also keeping to curve of the steering wheel, making it more comfortable for the driver.

TEST LAYOUT #3|4|5|6: These orientations use triangle arrangements, which are unorthodox and were mainly tested to see if there were any recommendations.

When it came to testing these layouts, there were 8 participants who tried all of the layouts. The order of the layouts tested were T.L. 5, 2,6,3,1,4 because layout 1 and 2 were only different by the how each button was positioned vertically, as so the user could get used to the testing scheme. When testing the layout, the user was timed on a series of actions they had to perform, and how quickly they could adjust and perform these actions. Initially, the buttons correlated to numbers on the display to allow the user to acclimate to the various layouts. They were timed on this acclimation. This test usually involved pressing the buttons in a particular order, i.e. button 3, button 5, button 1, button 2 etc. Then the user was then tested on how quickly they could complete each task at hand. The tasks ranged from merely entering the navigation menu and returning to the home screen, to a series of tasks such as changing the fan speed, entering entertainment menu and changing the mode to CD etc. After each test, the users were interviewed as to any recommendations they had, how comfortable the layout felt, how intuitive it was and why they did or did not like it. Refer to Appendix A1 for an example of the exercise we gave testers for the button layout.

Test Results

As mentioned, we timed each user on their ability to complete the given exercise tasks. We then averaged the times over all users to create an estimate of the best layouts, based on completion time as seen in the Button Layout Survey Results below:

Steering wheel Test Layout #1: 27.175 seconds
Steering wheel Test Layout #2: 27.125 seconds
Steering wheel Test Layout #3: 41.600 seconds
Steering wheel Test Layout #4: 31.088 seconds
Steering wheel Test Layout #5: 45.147 seconds

Steering wheel Test Layout #6: 30.663 seconds

The 4 triangle based layouts, TEST LAYOUT (T.L.) 3, 4, 5 and 6 were all the least desirable and hardest to use by testers. Overall, they thought that the layouts were not very intuitive and that using them did not correlate well with the buttons on the screen. It was hard for the users to intuitively figure out which buttons correlated to the on screen buttons. These T.Ls had the longest execution time. T.L. 4 and 6 were the easiest of the 4 to understand, as they were relatively similar to T.L. 2.

The T.L. 1 layout was the easiest to understand and use as it complied perfectly with the on screen display. The only downside was that it felt uncomfortable to press the buttons at certain times, especially for thumb positioning for the bottom button.

The T.L. 2 layout was the most favorable in all aspects tested. It was easy to use, comfortable, intuitive and did not give the user any strain or discomfort. This was the highest recommended layout.

Both T.L 1 and 2 had the overall fastest execution times by most testers, with T.L. 6 and 4 being the next fastest. T.L. 3 and 5 had by far the slowest times, and were not recommended to be used, since they weren't intuitive. Based off of user testers' responses, it was recommended to us to use T.L. 2 or 1 and make sure that the button positions were relatively close to hand positions of 9 and 3 on the steering wheel. It was also brought up that when turning the steering wheel, the user may accidentally bump the buttons, so to make sure they are flush or even slightly indented with the steering wheel.

Another recommendation that was made, though this is outside the scope of our project, is to have some sort of indication on the screen when the user is "hovering" over a button, so they know they are pressing the right one. It was also recommended to have some sort of glow or indicator as to which button was pressed on the screen, as in a color change or something of that sort as some testers were unclear as to which button was pressed.

These user tests were really useful in determining our button placement on the steering wheel, as well as giving us some pointers on how to make the interaction for the user easier to use. Not only did it give us orientation layout results, it also gave us interaction results with our user interface.

5. Project Execution Performance Evaluation

Timeline

Original Timeline

October 30: Mid-Level Requirement Document Completed

November 8: Technology Review & Technical Implementation Plan Completed

November 10: Operating system setup and running on hardware.

November 15: Test code working on Intel hardware, basic framework for building user interface complete.

November 24: Basic user interface functionality prototype completed

December 1: Protocol mocking complete

January 10: Steering wheel and button control prototypes

January 24: Basic vehicle hardware integration completed

February 15: User studies and analysis of completed design, possible improvements

March 20: Feature Lock

May 7: Cornell Cup Competition

Timeline Adjustments

The primary adjustments in the timeline center around the hardware integration. The sourcing of proper screens proved to be difficult. The original plan was to be able to run three small, separate screens off the competition hardware. However, after further investigation and confirmation when we received the competition hardware, we realized this would not be feasible. Our task then switched to find a screen with the proper aspect ratio that would fit in a vehicle.

Another adjustment was made by focusing on using a simulator, rather than actual vehicle setup and associated protocol and mocking integration allowed us to focus on the key innovative features, rather than work around protocol conversions. There was also an unexpected approximately two week delay in getting the simulator data available for use, due to documentation inconsistencies with actual program behaviors

Revised Timeline

October 30: Mid-Level Requirement Document Completed

November 8: “Technology Review & Technical Implementation Plan” completed.

December 18: Speedometer with basic speed feedback implemented.

January 13: Basic menu system completed.

January 29: Mid-term review

February 6: Begin development on Terasic hardware.

March 3: Basic steering wheel integration with simulator software.

April 2: Dashboard resolution updated for new screen size.

April 14: Racer software simulator data output connected to the screen.

April 14: Dashboard size screen working with Kontron LVDS.

April 16: Audio player integration working.

April 21-May 2: Finalize implementation.

With our team consisting of a 3:1 ratio of software engineers to electrical engineers, we're extremely happy of the progress we did make. We anticipate some additional work after the competition to polish any loose ends (see next steps section). Also between the time this document is complete and the Cornell Cup competition there is continued development work happening, and final timeline and feature set is subject to change.

Budget and Expenditure Justification

[See included Excel spreadsheet for budget information - MetroSwiftBudget.xlsx]

Beyond travel expenses (airfare and hotel for the team), the most expensive hardware part of the project was the purchase of OLED screen buttons to embed in the steering wheel control. While we had difficulty getting these to work completely properly at this time, development on this feature will continue after Cornell.

We were also able to reuse a number of hardware components that we had often literally lying around available for use at the university. This includes a Kontron board currently being used to provide the correct LVDS output on the display screens. These were not included in our budgeted amounts.

Funding

Special thanks to our advisor, Kevin McGrath, for covering the outstanding expenses after Cornell and Oregon State EECS department funding.

Mid-Project Reviews

The mid-project review was a helpful checkpoint in the development process to explain our project to an outside resource, and be able to get feedback on both the design and the process.

As of the midterm, we had prototypes and plans for many of our features. Through the review, we were encouraged to apply actual user testing and metrics to our plans, in order to better understand how users actually will be helped by using our system over other designs.

We were also encouraged to perform integration testing as early and as often as possible, to make sure the various parts work together as intended. As of the midterm, we had prototypes and plans for many of our features. Through the review, we were encouraged to apply actual user testing and metrics to our plans, in order to better understand how users actually will be helped by using our system over other designs.

Process Understanding

The process we followed in completing this project was very insightful and we gained a lot of knowledge about various concepts and techniques implemented in our solution. We learned about useful tools for making user interfaces. The Qt framework and QML is really easy to use and develop in. This tool is something that we would use again in developing user interfaces, especially to expand upon this project. We also were able to expand our knowledge in using C++ and integrating it with QML. It was also really interesting to interact with an open source simulator. We decided on taking this route, of using a simulator instead of integrating it into a vehicle, especially after seeing Oregon State University's own vehicle simulator and how they integrated a digital display cluster into a real vehicle.

When it came to feature integration, we discovered the difficulty of developing various features by doing spikes. This technique really helped us in isolating and prioritizing which features would be plausible and which wouldn't be. There was a lot of time management involved in this project and figuring out how much time certain tasks would take to be completed. One of the biggest things that we learned is to always allow more time than you think you need. Since we didn't know exactly how difficult this project would be, nor did we know how long each component and feature would take to complete, we had to add time buffers to all completion dates.

6. Recommendations and Next Steps

There are a lot of features we would still like to implement in this project. Since time was our biggest restraint, we limited the features we implemented. If given more time we would have prioritized the features differently. For example, since portability was an issue for the Cornell Cup competition, we weren't able to actually implement this is a real vehicle.

Further Improvements

One of the biggest things we would want to continue with is the menu system of the user interface. There were a lot of features that weren't fully implemented, things such as entertainment, navigation, rear view cameras etc. These features were on the list of things that we wanted to implement but didn't have time. The hardware and software we used in this project was new to us, and so a lot of learning, experimenting and testing ideas were involved to give us an understanding of how to implement features.

We would like to have had a fully interactive menu system which would give the driver access to a navigation system based off of GPS. Our idea originally was to use Google Maps API for giving navigation data. We had done earlier tests with GPS and CAN bus Bluetooth devices which would transfer information from the vehicle to our laptop collection system. Using Bluetooth GPS data we were able to pinpoint our location. SO one of the key features we would have implemented was navigation using Google Maps API and Bluetooth GPS data. We knew we could gather the information; the hardest and next step would be to interpret that information and display it on our menu system.

One of biggest things we wish we could have done if portability wasn't an issue is fully integrating this system into a real vehicle, and gathering data from the on-board CAN bus protocols and send it to the display unit. That is the ultimate goal of the project, to replace the current display cluster in a vehicle. This would also involve more testing and making sure that the display is accurate, and reliable.

Another aspect that we would like to do is do more testing to try and improve this system for more than just the general driver. For example, our system assumes that the driver is not colorblind. Looking at corner test cases was outside the scope of our project but would definitely be something that should be looked out in the future. Something we could implement in that case would be to either have a different type of glow, with different contrasts, or use sound to inform the driver when they have exceeded the speed limit, i.e. an increasing decibel buzz or beep.

Also when it comes to usability, it would a good improvement to have more tactile feedback when it comes to using the buttons on the steering wheel. For example, when a driver lays their finger on a button, without pressing it, an indicator on the dashboard may be informative as to which button they are about to press, instead of them pressing it and then realizing they pressed the wrong one, and having to look down to make the right selection. That was one of the testing criteria for the layout of the buttons, is making sure they felt comfortable, and intuitive for use.

Something that would be really interesting to add is a mobile application for a passenger. We discussed how the interaction between our system and a passenger's mobile which could have many use cases. We did a couple spikes on how difficult it would be to develop a Windows phone app or an Android app. A passenger could use their app to interact with the navigation system, and upload a destination to the driver's main display. The app could also be an extension of the display and steering wheel rig, enabling the passenger to change vehicle settings, gather information about the vehicle, or change the entertainment settings. This would decrease the driver's distraction level even lower.

Necessary Resources

To fully implement this system into a vehicle would require a lot of time, work and effort. Firstly, we would need to change our data input from being data from Racer over the UDP port, to actual CAN bus data. Secondly, we would have to replace the current gauge cluster dash of a vehicle with ours, and have it wired up correctly. We would also need to connect the controls for climate, entertainment etc. into the controls for the vehicle which would take a significant amount of time, depending on the vehicle at hand. To implement and integrate our system into a vehicle would require a vehicle, CAN bus adapter which could stream data to our system, hours of accuracy testing, making sure our system is producing correct data, as well as reliability and robustness testing, to make sure that the system still works under extreme conditions, since it is the primary and only control system for the vehicle.

To implement the entertainment and camera aspects of the project would require us to rework our code for QT Quick 2.0, which would take a significant amount of time. We did a spike to see how much effort it would take to update our code, and given the scope of our given project, we did not see it plausible at this current time. We would also need to figure out a way to integrate the 1 or 2 webcams into the system. We were originally thinking 2 Xbox Kinect cameras since there are a lot of open source development projects available with this technology. To implement the cameras would require some sort of API which allowed for video display in the program. Since this type of system has been integrated into other cars successfully, we believe this was outside the scope of our solution. When doing earlier spikes on the features, we discovered a significant amount of resources for Google APIs supporting these features. It would just take some time to learn and implement this into our QML coded system.

To implement the navigation system, this would be not only more applicable for an actual vehicle, where navigation would be necessary, but it would also require more understanding of Google maps API. It was our original idea to use this information, especially since our speedometer would use that information to indicate to the driver when they are exceeding the speed limit. An early spike of this feature would not take too much effort especially since there is support for Google Maps API in QML, it would just take time integrating this into our system. The navigation feature would also require us to use some sort of Bluetooth GPS unit to inform the program of our current location. These are readily available devices, but integrating it into our system may require some time and effort.

To develop and build a smartphone application for a passenger to use in the vehicle would require a significant amount of time, not only to develop, but also to test and integrate into our system. It would be really useful especially today, since a lot of people have some form of a smartphone. It would also be necessary to develop an app on all major mobile operating systems (Android, Windows and iOS). This would make an excellent software focused follow-up project.

We anticipate some of these projects may end up being done as other capstone or class projects here at Oregon State, and we look forward to seeing what future project teams might be able to develop using the framework that we have created.

7. Glossary

CAN bus protocol:

Control Area Network is a serial message based protocol system used primarily in automotive applications. Enables on board sensors to communicate with the central computer system.

FPGA:

Field Programmable Gate Array, an integrated chip designed to be programmed after being manufactured, by the customer, especially useful when doing complex logic.

LVDS:

Low Voltage Differential Signaling, an electrical digital signaling standard used especially in LCD displays for laptops, tablets, TVs, automotive infotainment systems and communication devices.

OLED:

Organic Light Emitting Diode, used to create digital displays on mobile devices.

QML:

Qt Markup Language - A declarative scripting language as part of the Qt framework which allows for intricate graphical user interfaces.

Qt:

Qt is a cross-platform application framework that is used to create graphical user interfaces.

Racer:

Freely available open source racing simulator software available from http://racer.nl

SPI:

Serial Peripheral Interface bus is a synchronous serial data link which communicates using the master/slave mode.

UDP:

User Datagram Protocol for fast IP communication. Does not provide any reliability guarantees. Used for communication between simulator and dashboard.

Wunderboard:

Oregon State University’s microcontroller that features an Atmel 8-bit controller, LED display and USB ports.

8. References

[1] United States. Dept. of Commerce. *Statistical Abstract of the United States, 2012*. Washington, D.C.: U.S. Census Bureau, 2012. Web. <http://www.census.gov/compendia/statab/2012/tables/12s1103.pdf>.

[2] Millikin, Mike. "Green Car Congress: Fuel Consumption at Higher Speeds." *Green Car Congress: Fuel Consumption at Higher Speeds*. BioAge Group LLC, 22 May 2009. Web. 20 Apr. 2013.

9. Acknowledgments

We would like to thank D. Kevin McGrath for his help, support and vision on this project. He encouraged us to undertake this project and enter into the Cornell Cup to represent Oregon State University, and also providing funding and resources to make it a reality. We would also like to acknowledge and thank Team Mojo2 for their help with using the Wunderboard and SPI bus protocols for our OLED buttons, and for being great office mates!

Special thanks to Dr. Mike Bailey for his guidance during our project and helping us with our presentations, and making sure we were keeping to deadlines. We would like to thank the sponsors of the Cornell Cup, especially Intel for their customer support and helping us understand the capabilities of the hardware they supplied.

We would like to thank Reese Russell for sharing his knowledge of FPGA's and providing assistance with debugging and implementation, and Joey Conrad for his Solidwork's expertise, and helping design our steering wheel adapter.

10. Survey Results

Initial Survey Results:

Research Questions and Goals:

- Is text easier/quicker than icons to understand what a menu represents at first glance?
- Is information easier to understand and less distracting in front of the driver or on the center console?
- Is touch screen more distracting than physical buttons?
- What features are used more than others? What needs to be easiest for the user to navigate to and what can be put into a sub-menu?
- How many button is too few? How many is too much?
- Do users prefer to be notified about important things via audio, onscreen alerts, or some other means?

Interview Questions

- How many years have you been able to drive a car?
- What three things do you use most often inside the car? (Purposefully vague to see whether the subject emphasizes driving functions like steering, lights, and wipers, or auxiliary functions like a/c, radio, etc. We clarified from there to ask about what entertainment features are most used, if the subject went another direction.)
- In the center console of the car, what is your most often used action?
- How long is your most frequent trip in the car?
- What do you do during that drive?
- When you're driving with a passenger, do you or the passenger control the radio, navigation, etc.?
- How often do you have passengers in your car?
- If you were to buy a new vehicle today, what features would you expect in the entertainment system?
- In what way do you expect information to be displayed in your car? (Left vague to see different responses. e.g. warnings vs text messages vs trip information vs navigation instructions).
- Do you want audio or on-screen/in dash alerts? Are there certain types of alerts you'd prefer e.g. audio vs onscreen? (check engine, lane exit, incoming call, lane exiting, etc.)
- What is the most frustrating feature of your vehicle?
    - Why is <u>*feature x*</u> frustrating?
    - If given enough information, we asked how they would improve it.

- Tasks the subject was asked to complete in their own car:
    - Play song 7 on a cd.
    - Turn on the front window defrost.
    - Change the radio to AM 550.
    - Change the clock to 24 hour mode.

- We asked the subject to identify the functions of a number of buttons and switches in the vehicle (Trying to observe learnability/memorability of features)

Interview Process

We conducted interviews & observations with two subjects. The interviews were conducted in an actual vehicle, which is the “field” for this setting. This allowed for interview questions to be asked about specific features of the car and allowed the subject to be observed while they operated the vehicle. It also helped the subject to remember features about their vehicle that they liked or disliked since they could see their dashboard. A semi-structured interview format was used when asking questions.

Triangulation of conclusions was attempted by using the same list of starting questions for each of the subjects interviewed to see if we could get similar results. The interview contained introductions of the team, explanations of the process, and a few general warm-up and cool-down questions which are not recorded here. During the interviews, each team member observed a particular portion of the context (people, objects, environment) or the responses.

Answers to Research Questions

Is text easier/quicker than icons to understand what a menu represents at first glance?
It didn’t seem to matter whether a button or menu was represented with text or icons. What was more important was that the text or icon made sense with the function of the menu and was easy to understand after a short glance.

How many years of driver experience do they have?
Our subjects had between 4 and 7 years of driving experience, but our sample was also of young drivers. Other drivers will have many more years of experience.

Is information easier to understand and less distracting in front of the driver or on the center console?
The location didn’t seem to matter as much the consistency of where information was displayed. Subjects complained about information being displayed in too many places, which caused distraction as they looked for the information. One subject did request that information relating to the health of the car be in whichever instrument contained other car-related information.

Is touch screen more distracting than physical buttons?
One subject was able to use her iPhone’s touchscreen without looking, but this required the ability to hold it in her hand and take her attention off of the road. This wouldn’t be an option with an in-dash touchscreen, which implies that a touch screen would be distracting.

What features are used more than others? What needs to be easiest for the user to navigate to and what can be put into a sub-menu?
Aside from driving-related features like wipers, headlights, and shifters, both subjects frequently used the radio. Neither seemed to use CDs much, as they just used their phones. Navigation was also important. As for driving related functions, it depends on the level of knowledge the driver wants to have about the car. Our subjects had varying degrees of automotive knowledge, so they wanted varying amounts of information displayed. Some users may be happy with a check engine light, while others want a detailed error message so they can fix the problem themselves.

How many button is too few? How many is too much?
It’s unclear how many is too few or too many, but it’s important that they be easily accessibly to the user. One subject really likes the buttons on his steering wheel because it means he doesn’t have to go looking for them. The learnability of these buttons would be worth researching.

Do users prefer to be notified about important things via audio, onscreen alerts, or some other means?

Users seemed to want to be notified of most things on-screen, and weren't sure about audio alerts. More research will have to be done into this area. One subject mentioned she appreciates being able to have vocal directions from passengers and vocally ask them to skip songs, but this may not carry over into navigation and entertainment systems. Another user mentioned that they want more options regarding how alerts are presented, specifically navigational alerts.

Other Insights

- Small touches are important. Things like the radio fading in and out instead of abruptly turning on or off make the user more relaxed and comfortable.

- Users want to feel in control of their car. The feeling can be affected by the amount or type of information provided to the user, as well as the intrusiveness of driver assist functions and what the car allows the user to do while in motion.

- The amount of attention demanded by an alert should be proportional to the severity of the alert. A alert about the outdoor temperature, for example, should have a lighter warning than a flat tire. That should definitely be a flashing, annoying alert.

- Setting up of devices must be simple and fast. It happens infrequently, but is often the first experience a user has with the car.

- Media related interfaces need to be accessible by the passenger, in case the driver wishes to delegate.

- Systems designed for the passenger to control can be more involved, even when the vehicle is in motion, because they are not concerned with driving.

Button Layout Results:

Below are the raw results from our user tests for figuring out the ideal layouts for the OLED buttons on the steering wheel. All times given below are in seconds (s). The time listed for each test layout is the total time

User Test #1:

Test Layout #1-->27.4s
Test Layout #2--> 27.4s
Test Layout #3--> 40.4s
Test Layout #4--> 30.3s
Test Layout #5-->49.7s
Test Layout #6--> 29.9s

Recommendations, comments and answers to questions:

The layouts for 1 and 2 were the easiest to use. I preferred 2 because it felt more comfortable since the steering wheel was circular. The triangle ones did not seem normal at all, at least for me, but 4 was definitely the easiest of the triangle ones. The hardest part was figuring out what the buttons did. It was hard to tell which button I was pressing, since there was no feedback. If I had any recommendations, it would be that maybe have some feedback on button presses so that the driver knows what they pressed. Also make sure the positioning is relative to where the hands on a steering wheel would be, around 9 and 3.

--//--

User Test #2:

Test Layout #1-->27.9
Test Layout #2-->27.4s

Test Layout #3-->41.9s

Test Layout #4--> 31.6s

Test Layout #5-->50.6s

Test Layout #6--> 30.7s

Recommendations, comments and answers to questions:

This was really fun. I liked it. I would really enjoy having this in my car. I would prefer to have the vertical layout ones, the triangle ones were weird, and I had to look down too much at the wheel to see which were which, plus I have small hands so I can't reach too far. So I like the up down ones.

---//--

User Test #3:

Test Layout #1--> 25.1s

Test Layout #2-->26.2s

Test Layout #3-->39.7s

Test Layout #4-->29.8s

Test Layout #5-->48.5s

Test Layout #6-->29.5s

Recommendations, comments and answers to questions:

I didn't like the triangle layouts, the 3rd, 4th, 5th and 6th were horrible. Otherwise it was cool to use, its just like using cruise controls except expanded. It would interesting to have some sort of feedback for the buttons, like vibrations or on screen blinkers or something. Also, what happens if the person using this is color blind?

---//--

User Test #4:

Test Layout #1-->26.9s

Test Layout #2-->27.3s

Test Layout #3-->46.6s

Test Layout #4-->33.3s

Test Layout #5-->52.5s

Test Layout #6-->31.6s

Recommendations, comments and answers to questions:

This is very strange to use. Steering wheels keep getting more buttons, and I am surprised that these buttons do so much. It will be cool to see this in a real vehicle someday. I like the vertical layouts the best though the 4th layout and the 6th made sense. I don't understand the point of the 3rd or 5th. This is a really interesting idea, though I do like my knobs for changing volume and climate stuff, so it would take some time to get used to.

---//--

User Test #5:

Test Layout #1-->27.6s

Test Layout #2-->27.6s

Test Layout #3-->40.3s

Test Layout #4-->30.8s

Test Layout #5-->52.9s

Test Layout #6-->30.1s

Recommendations, comments and answers to questions:

I had some trouble getting used to this, 5 minutes was definitely not enough. A new driver would have to take a while to get used to this, though after using this for a while they would easily be able to use this. I think the buttons that keep the wheel curvature work the best in my opinion, though they might be hard to reach for people with small hands. Maybe you should have some visual feedback on the screen for the button presses.

---//--

User Test #6:

Test Layout #1-->27.4s

Test Layout #2-->27.9s

Test Layout #3-->40.3s

Test Layout #4-->30.8s

Test Layout #5-->54.4s   Test Layout #6-->32.9s

Recommendations, comments and answers to questions:
I don't really like the buttons for this. It feels weird. I like having buttons and knobs on the center dash for this sort of thing. Granted I've been driving for over 40 years so I am used to old vehicles with less technology. I feel it could get really distracting. Personally, I wouldn't use any of the control layouts, but I can see how newer drivers would find them easy to use. I would get frustrated with it though, especially if I did something wrong, and I have no feedback to reverse what I did.

--//--

User Test #7:

Test Layout #1-->27.1s   Test Layout #2-->26.5s
Test Layout #3-->39.7s   Test Layout #4-->30.8s
Test Layout #5-->49.9s   Test Layout #6-->30.3s

Recommendations, comments and answers to questions:
This is wicked cool. I want this idea in my vehicle, it's like having a game controller in my steering wheel, I feel like I am playing a racing game. As a gamer I like the triangle ones because the buttons are closer together but this could get confusing for people. The vertical ones would be easy if this was the first time driving. I would be able to get used to all of these so I don't care which one would be used. They would all be fairly easy to get used to.

--//--

User Test #8:

Test Layout #1-->28.0s   Test Layout #2-->26.7s
Test Layout #3-->43.9s   Test Layout #4-->31.3s
Test Layout #5-->51.9s   Test Layout #6-->30.0s

Recommendations, comments and answers to questions:
I prefer the vertical layouts, 1 and 2. The triangle ones were uncomfortable and not easy to use. It would be nice to know which buttons correlate to which control. I like the button idea, I really want this in my car. I only needed to look down at the triangle ones, otherwise it was easy to use. Though, I could see where people may accidentally hit one of the buttons, and cause something to happen that they weren't wanting, make sure they are inline with steering wheel depth i.e. flushed or indented.

--//--

11. Appendix

Appendix A: Button Layout Example Test

Thank you for participating in this study. Below are a series of commands to complete. The results of this study will help establish the best layout for our OLED buttons for our Cornell Cup project. This test is completely voluntary. If at anytime you feel uncomfortable you are welcome to take a break or stop. These exercises will be timed, so please try to do this as fast as possible. At the end of the test you are welcome to ask how fast you completed the exercise. We would also like you to include any comments and recommendations. We have included some question prompts at the end to help. This first test is a series of 3 exercises to help acclimate you to the button layout on the steering wheel. This will be timed, but these results are not going to be used in our study.

E1. Complete this sequence of Button Presses:

A. START:

OLED Button 1
OLED Button 2
OLED Button 3
OLED Button 4
OLED Button 5
OLED Button 6

FINISH:

B. START:

OLED Button 1
OLED Button 5
OLED Button 2
OLED Button 4
OLED Button 6
OLED Button 3

FINISH:

C. START:

OLED Button 6
OLED Button 3
OLED Button 1
OLED Button 4
OLED Button 5
OLED Button 2

FINISH:

We will now give you 5 minutes to explore the menus to help you understand where everything is in the display. The next section will ask you to complete a series of tasks and we will time you.
You may now begin your 5 minutes.

E2. Now that you have had time to explore the menu system. We would like you to complete the series of exercises as fast as possible:

A .START:

Beginning at the Home screen, increase the fan speed by 3 increments and increase the temperature to 80 degrees and return to the Home screen.

FINISH

B. START

Beginning at the Home screen, change the music selection from Radio to CD and return to the Home screen.

FINISH

C. START

Beginning at the Home screen, enter the Navigation screen, and select the Map option and return to the Home screen.

FINISH

D. START

Beginning at the Home screen, enter the Settings menu, and enter the Entertainment Settings menu. Then return to Home screen and change the music selection back to radio and return Home.

FINISH

Thank you very much for your participation. We would like to ask for any comments or recommendations about the layout of the buttons. Here are some prompts to help you. All answers are completely anonymous and you are not required to respond.

1. Did the button orientation seem comfortable?
2. Was it intuitive as to which button to press?
3. Did you feel compelled to look down at the buttons to know which button to press?
4. Was there any strain or uncomfortability felt when completing the exercise?
5. Finally, do you have any other comments, recommendations or questions about the study, the display, the button layout or the interactions you experienced?

Thank you for your time and participation. All results will remain completely anonymous.

Appendix B: List of Materials

Materials list as of April 21, 2013, subject to change as development is continuing.

Terasic DE2i-150 (Cornell Competition Board)

Sharp 12.3” TFT LCD Color Screen
    Model Number: LQ123K1LG03

Kontron MSMST
    Intel Atom board being used for LVDS support

Logitech Driving Force GT Steering Wheel

6 - NKK OLED Switches
    Model Number: ISC15ANP4

2007 Acura MDX Touring Edition Steering Wheel

Wunderboard / Arduino (For button/keyboard interface)

Appendix C: Kontron Data Sheet

See MSMST_Manual_V109.pdf in the zip file.

Appendix D: TFT LCD Monitor

See Sharp_LQ123K1LG03.pdf in the zip file.