Next Generation of Touchscreen Interface

Mitchell Loeppky
UW CSE
1216 NE Ravenna Blvd
Seattle, WA 98105
1 (360) 393-8400
mitchell@loeppky.com

Vinod Rathnam
UW CSE
4235 12th Ave NE 203
Seattle, WA 98105
1 (425) 894-4133
vinodr@uw.edu

Nathaniel Steinbock
UW CSE/ME
8423 NE 123rd Place
Kirkland, WA 98034
1 (202) 643-3334
stein3@uw.edu

Michael Mallory
UW CSE/EE
5219 Brooklyn Ave NE 102
Seattle, WA 98105
1 (360) 393-6843
malbot@uw.edu

ABSTRACT

EVE is a touch-screen interface that has been created by students at the University of Washington to address current issues with infotainment technology in cars. One issue is that of safety. With the introduction of new, extravagant center console interfaces, simplicity has been lost and drivers find it difficult to refrain from starting at the screen instead of the road. The other issue is how efficiency is addressed. The current emphasis is on making vehicles more efficient, but there is not enough feedback given to drivers on how they can improve their driving habits. We decided to explore these problems by creating a touch-screen interface that has applications developed and designed to not sacrifice safety for beauty. Haptic feedback and gesture sensing was used to further encourage drivers to keep their eyes on the road. The interface was also designed to provide feedback to the user on how they can improve the efficiency of their driving through more specific information that can only be seen in park, as well as through real-time data that is still designed not to minimize their safety. The interface was created and developed in cooperation with the University of Washington EcoCAR 2 team for the GM EcoCAR 2 competition. The results of the infotainment system are yet to come, but we are confident that the results will illustrate the usefulness of driver feedback and that the move towards touch-screen interfaces in car need not be related to minimal safety for those within it.

Keywords

Infotainment, Touchscreen, Haptic Feedback, Gesture Recognition, Centerstack, Automotive, Human-Interface

1. INTRODUCTION

As the amount of computing resources and technology has increased in modern vehicles, the use of touchscreens has become more commonplace. However, the lack of buttons and knobs makes it difficult for the user to interact with the screen without looking at it, which is dangerous for a driver, even for a couple seconds. The NHTSA (National Highway Traffic Safety Administration) create guidelines last year that a single action should take less than 2 seconds, and a maximum of six screen touches in 12 seconds, all to reduce the amount of time the driver is not looking at the road [1]. This can be improved, by replicating the effects of knobs and dials on a touch screen. This provides the best of both worlds, the safety aspect to both the driver and other vehicles from knobs and sliders as well as the creative aspect and unique abilities provided by the touch screen, as well as integrating a gesture system, so that the drive does not have even use the touch screen for a handful of basic commands. Older touch screens fall way below this standard, with some features, like GPS or climate control, taking a significant amount of time and focus to accomplish simple tasks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission.

CSE 477 – Final Report

2. RELATED WORK

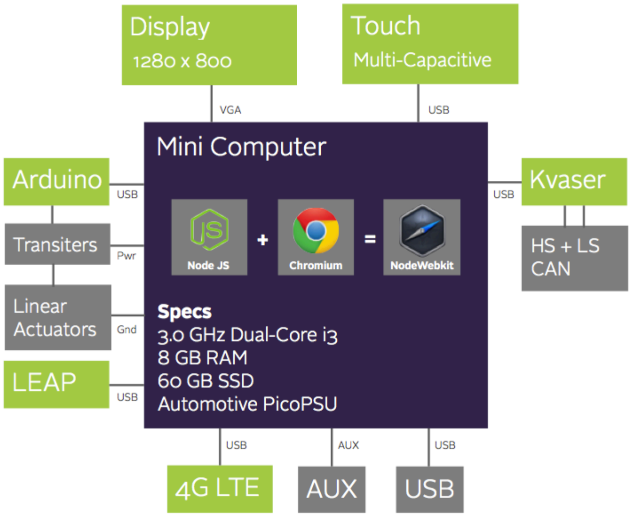
There is a wide range of touch screen or simple non-touch screen interfaces for cars, a large number of vehicle manufactures have their own designs, and even some non-vehicle companies like Apple Inc, have started to produce their own vehicle screens. Like ours, Apples patents incorporate tactile feedback, where the screen vibrates, simulating the feel of physical controls, which, like ours, attempts to reduce the amount of time the driver is looking at the screen and not at the road. This patent is an update from an older patent, which uses a project instead of a LCD display. With the current advances in technology, it would be surprising if Apple stuck with the projector screen, but it gives a good idea of what Apple wants to accomplish with their CarPlay centerstack [2].

Lexus has developed an interface dubbed Remote Touch. Instead of using a traditional touch screen, the interaction is all enabled via an input device lower on the center console. This device is similar to a computer mouse; however it adds a spectacular level of haptic feedback. The mouse has the ability to snap to on-screen elements in order to aid the driver in finding and selecting different options. This allows it to be used as either an on-screen pointing device, or as a bumper to "tap" between adjacent options [4]. Other approaches to driver-interface interaction includes Audi

3. TECHINAL DETAILS

3.1 Overview

The touch screen interface incorporated multiple different hardware components as well as software to be able to perform all of the functions that where accomplished. The main piece of hardware that drove the system was the Mini Computer, which ran a Node Js and Chromium blend called NodeWebKit for the application that the user interacts with on top of a Linux operating system. All of the other hardware and software supported these either by giving extra functionality or enabling the system to communicate with the rest of the vehicle. Below is a block diagram of how everything is incorporated into the touch screen interface that replaced the center stack of the 2013 Chevy Malibu.



3.2 Theory of Operation

The primary goal of EVE is to improve the safety of operation. To achieve this goal, several features and design decisions were made utilizing multiple human sense and intelligences.

The most important sense that conventional touchscreens do not utilize is touch. This sounds counter intuitive, but it is because a conventional touchscreen takes in touch input, but does not send any touch signals back to the user. In the case of a conventional temperature control dial, the user passively receives feedback that they are holding the dial and can feel a slight click as they rotate to the next temperature level. By adding our custom bezel, the user can feel when their finger is in the location of a button or slider. Additionally, by adding haptic feedback, the user is actively informed when they increment through a temperature list.

Conventional touchscreen interfaces attempt to be very stimulating and provide as much information for the user to be able to view as possible. EVE is designed to make sure only the user needs to see is on the screen at anytime. Additionally, the user should always be able to look in the same spot to find the information they are looking for. This is all with the purpose of minimizing the amount of time that the user has to remove their eyes from the road. For this reason, the design team created a custom icon set that is very minimalist and easily recognizable.

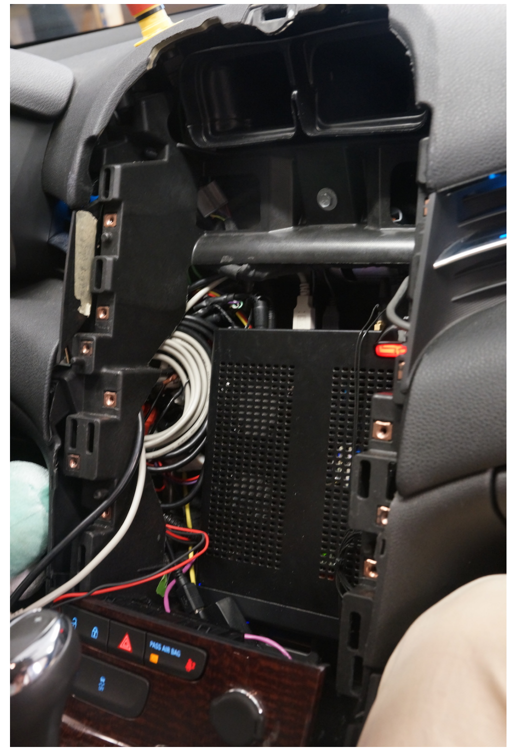
Finally, the gesture recognition does not require the use of any senses, but instead relies on the driver's memory to remember 4 simple actions he or she can take.

3.3 Implementation Details

3.3.1 Hardware

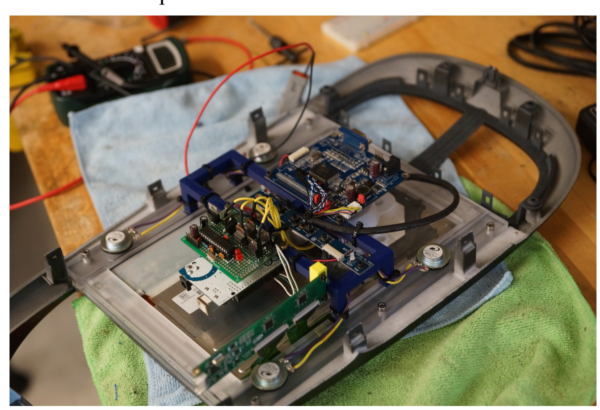
3.3.1.1 Mini Desktop PC

Controlling the majority of the front-end is a full-fledged desktop computer. Consisting of an Intel i3 Processor, 8GB of RAM, and a solid-state drive, it has plenty of power to handle the CAN-interfacing and display. The desktop is hard-mounted into the vehicle behind the dashboard, and it is powered off of a relay controlled by the supervisory vehicle controller.



3.3.1.2 ATMega328 Microcontroller Board

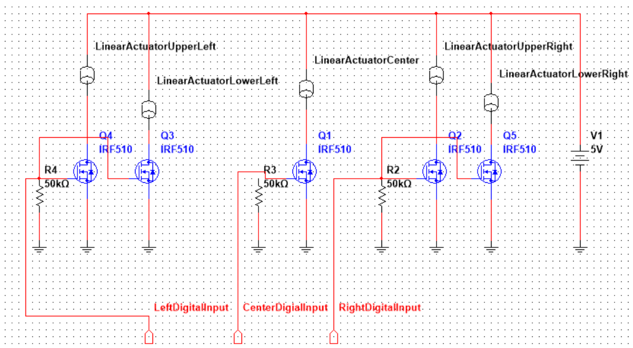
The haptic feedback feature of our centerstack system is a key component of the eyes-free use of our interface. To control the haptic feedback, there is a custom-built microcontroller board. This board takes advantage of an Atmel ATMega328 processor. The board is connected to the mini PC via a USB to serial interface, and to the haptic-feedback linear resonant actuators via a custom build amplifier.



3.3.1.3 Linear Resonant Actuators

In order to provide haptic feedback to the user of the interface, five linear resonant actuators (LRA) were strategically placed along the back of the screen. Four 3-watt actuators were placed along the borders of the user-interaction area at the bottom half of the screen in order to weight the vibration to the left or right side of the screen. An additional, larger LRA is located central to the screen and provides additional force for the menu "clicks" felt when scrolling up and down through settings.

The LRAs are all amplified via a custom-made transistor-based amplifier. Using digital inputs from the ATMega328, the transistors turn on and off the flow of current through the LRA. The transistors used are IRF510, which are fast switching, MOSFET, and n-Channel, chosen because of their availability and size, as well as their current and voltage capacity. Because of the low resistance of the LRA, the transistors had to be able to handle over 3 Amp of current and up to 12 volts, which the IRF510 could easily handle [3]. It was experimentally determined that this much current and voltage was more than the LRA could handle, and although smaller and cheaper transistors could have been used, the IRF510 were already acquired. Below is a schematic of the LRA connected with the transistors.



3.3.2 Software

3.3.2.1.1 Application Architecture

EVE is a web-application that is used through NodeWebKit, an app runtime based on Chromium and Node.js. Chromium is Google's open source project for their web browser Chrome. Node.js is a platform used to build scalable applications on Chrome's JavaScript runtime, V8. So, applications were written in JavaScript and HTML following the Marrionette.js MVC design pattern Marrionette.js simplifies the creation of large-scale JavaScript applications.

3.3.2.1.2 Visual Structure

Safety was highly considered throughout the development of EVE and its applications. This was a factor in the consistency of the overall structure of the different screens in the interface. The top half of the screen was designed to display the visual content that the user will have no tactile interaction with. The user's main controls are the five input zones in the bottom half of the screen. These 'sliders' were used in two different ways. One: by touching and dragging to make a selection from a list or range of options. Two: -- by tapping the edges with symbols to select an option such as front-defrost in climate control or to move to a new screen or application in EVE. By keeping this visual structure set through all the screens, the driver will not need to remember as much information regarding the location of his interaction with the interface. Furthermore, for the first time user, knowing whether a slider is touch-drag or tap may be difficult to ascertain. So, we included a nice feature so that when a touch-drag slider is tapped, green dots travel across the path of the slider indicating it must be touched and dragged. We believed this would help drivers build intuition of the system.

3.3.2.1.3 CAN bus

For many of the applications that EVE supports, information and access to controls in the car is necessary. To achieve this, a Kvaser USBCAN II tool is connected to the minicomputer on which the apps are run. The CAN (Control Area Network) is a bus protocol used for data communication between the different devices in the car. The GMLAN low speed network, accessed through the single wire (SWCAN), operates at a baud rate of 33.333kbps, whereas the high-speed bus operates at a baud rate of 500 kbps. The Kvaser USBCAN II comes with a C++ library with which code was developed to write and receive CAN messages for the functionality of different applications such as climate control and driver feedback. This C++ code was then made into a Node.js module that made integration to the JavaScript simple and clear.

3.3.2.1.4 Climate Control

Climate control is the most fundamental application that needed to be included in the interface after the previous center console was removed. Features that are provided include modification of fan speed, airflow direction, front defrost, rear defrost, and the temperature on the driver's side of the vehicle versus the passenger's side. The operation of these features was achieved through the sending of the appropriate CAN messages in the low speed network.

3.3.2.1.5 Music

The second most important application for the centerstack is the music application. It provides the ability to play music over the surround-sound system via any of USB, aux-cable input, or Internet radio. When browsing USB, the sliders change to allow song selection via title, artist, album, and playlist.

3.3.2.2 Haptic Feedback

There were four goals for the code developed for the haptic feedback. The first as that due to time constraints, communication (both the method and the commands) between the Centerstack computer and the microprocessor had to simple and reliable. Second was that the frequency of the linear actuators could be changed via the communication method chosen. Once the screen and bezel was installed, this would allow for adjustments in the frequency without having to remove the microprocessor for re-programing, as well as making testing different frequencies significantly quicker.

3.3.2.2.1 Method of Communication

For communication between the microprocessor and the centerstack computer, RS-232 Serial was chosen. Commands were sent via two character commands and depending on the command, followed by a single or multi-digit ascii number, finished with an ascii period. The first character of each command is which group of linear actuators the command is for, with the second the actual command. Through these commands, different frequencies could be chosen, length of time a group would be "pulsed", as well as starting/stopping the linear actuators.

3.3.2.2.2 ATMega328 Processing Commands

On the microprocessor side, these commands are processed and the parameters updated accordingly. To be able to adjust the frequency of the linear actuators, a timer interrupt on a 5 ms period was used to increment a different counter for each group. The frequency commands from the Centerstack computer where computed to the number of 5ms intervals for a single change of the signal. Once the counter for a group hit this value, the digital out associated with the group would switch states (either high or low), and restart the counter. Pulse commands where converted to the number of state transitions before stopping vibration

3.3.2.2.3 ATMega328 Controlling Linear Actuators

When the digital outputs of the of the microprocessor changed states, to either high or low, the transistors would follow suit, either start allowing current to flow through the linear actuators or stop the flow of current. Originally the design called for the use of H-bridges for driving the linear actuators, but because H-bridges are designed to be driven by two separate inputs and our design necessitated the use of only one input, we had to take precautions so that the transistors would not short themselves. These precautions drastically reduced the effectiveness of the H-bridge, so we switched to using a single transistor. Current could only flow in one direction, but the vibrations caused by the LRAs were several orders of magnitudes greater than with the H-bridge, and this solution required a simpler and smaller design.

3.3.2.2.4 Computer Side Software

Because EVE runs on the NodeWebKit framework, the software developed for the haptic feedback system on the centerstack side was done with JavaScript, built into a NodeWebKit module. This involved only a series of commands that would send data over a serial port to the Atmega328, to set up the frequency for each linear actuator groups, as well as starting the haptic feedback.

3.3.3 Gesture Recognition – Leap Motion

EVE has simple gesture recognition to aid in controlling EVE without the driver removing his or her eyes from the road. A lot of effort went into make the interaction pattern for the EVE apps very simple and therefore the goal with the gesture recognition was to merely supplement the apps as opposed to providing a completely different interaction pattern the user would have to learn. Therefore, the gesture recognition interactions are very simple. When the user's hand enters a 3 dimensional cube, a square appears on the screen with a white dot that represents the position of the hand in the 3D space and an icon at each edge of the square to indicate the available gestures. When the user's hand (and the white dot) is mostly still for a short duration, the dot turns green. At this point, if the user moves their hand off an edge of the square, then the action represented by the icon at that edge is invoked. To make it very easy for the user to learn the gesture system, the same four actions are universal across all of EVE. Swiping left goes to the previous track, swiping right goes to the next track, swiping down plays or pauses the music, and swiping up enters EVE mode. In EVE mode, the background of the screen turns to a gradient from purple to green. When the driver is driving very efficiently, the screen shifts to be greener. When the driver is not driving efficiently, the screen shifts to be purple. "Efficiently" is determined by the vehicle's Hybrid Supervisor Controller.

The LEAP motion is installed in the sunglasses holder of the vehicle. This areal location provided simpler mounting, and also was minimally effected by obscure lighting conditions and the gear selector.

The gesture recognition code utilizes the LEAP JavaScript API to know the velocity and position of the hand. Then it is a simple JavaScript state machine for controlling the UI. The initial state is OUTSIDE ACTIVE ZONE. When the hand is detected as inside the active zone, the machine switches to INSIDE ACTIVE ZONE. If the hand leaves the active zone, it will return to the prior state. If the magnitude of the velocity falls below a parameterized threshold, the machine enters STOPPED INSIDE ACTIVE ZONE. If the velocity exceeds the threshold, it returns to the prior state. Once a specific time threshold has been met, the machine enters GESTURE MODE. Once the hand leaves the active zone, it fires an event specific to the direction with which the hand left the active zone. This design was very simple to implement is and is very robust for a single hand. When two or more hands (or hand like objects) are in the field of view, this system is much less effective.

4. EVALUATION AND RESULTS

Field engineers from Freescale Semiconductors Renato Frias, John Cotner and director of Electrification and Control System Integration of General Motors John Haraf evaluated EVE at the Year 3 EcoCAR 2 competition. They evaluated the centerstack based on its level of integration and the development process we used when implementing. However, we will not receive the judge's feedback and receive a placing until 8:00 PM Eastern Time on Thursday, June 12th, 2014.

Although the judges have not provided any feedback, during the demo they were able to successfully feel the haptic feedback in the screen and complete the gestures available.

The UW EcoCAR 2 team has been awarded the Best Vehicle Appearance award as voted on by the GM mentors of the competition. Although the exterior design of the car was likely the primary factor for the judges, the design and integration of the centerstack was probably also a contributor. Below are the visual results of the project, first is the 3D printed bezel and custom 12inch touch screen. Next is the integration of the Leap Motion into the sunglasses case in the ceiling of the car.

5. DISCUSSION

Overall, the touch screen and its peripherals works well, accomplishing all of the goals that where set out at the beginning of the project. Improvements could be made to the haptic feedback LRA, either getting ones that produce more vibration and less audible noise or completing a detailed frequency analysis of the touchscreen. If the resonate frequency of the touchscreen was found and the LRAs vibrated at this frequency, their effects would be amplified without having to get stronger LRAs. The 3D printed bezel over the touch screen could also be better integrated into the car. Only the part over the touchscreen was printed, the top section that covers the vents and the lower section that goes around the gear selector were cut out of the existing bezel and glued to the printed bezel. This meant that the connections between the three parts were weak and prone to braking. Further improvement would be to print the entire bezel in one piece. The touchscreen originally designed to be horizontal, not vertical. Though the viewing angle was changed to accommodate this new orientation, anyone wearing polarized glasses where still unable to see the screen, unless they looked at the screen sideways. While this is fine for an experimental vehicle, which the centerstack and touchscreen is currently in, for a production vehicle, this would have changed to accommodate sunglasses wearing drivers.

5.1 Subsections

6. CONCLUSION

Overall, the Eve centerstack has been a great success. Over two weeks at the EcoCAR 2 final competition, it has garnered over 10 interviews with reporters from various news stations and newspapers as well as automotive influencers. There has not been a single user of the interface to complain about or dislike any feature, and it is by far the most successful of all 15 teams of EcoCAR 2.

In addition, some drivers other than Eve team members were even able to get the hang of using the centerstack interface while driving and without looking. This shows the success of our screen-dividing bezel overlay and the haptic feedback. Spectators, passengers, and drivers alike have all been able to get the hang of the hands-free gesture control.

7. REFERENCES

[1] National Highway Traffic Safety Administration. (2013). *Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices* (NHTSA Docket No. NHTSA-2010-0053). Washington, DC: U.S. Government Printing Office.

[2] Campbell, Mikey. (July 9, 2013) Apple's safe automotive telematics patent mates touchscreen with tactile controls, *Apple Insider*. Retrieve from http://appleinsider.com/articles/13/07/09/apples-safe-vehicle-telematics-patent-mates-touchscreen-with-tactile

[3] Vishay Siliconix (21-Mar-11). *Power MOSFET IRF510, SiHF510*. Retrived from http://www.vishay.com/docs/91015/sihf510.pdf

[4] Goodwin, Antuan, (Nov 19, 2008) 2010 Lexus RX gets update, new Remote Touch controller, CNET. Retrieved from http://www.cnet.com/news/2010-lexus-rx-gets-update-new-remote-touch-controller/