Digital car switch panel

**November 2023**

**By**

**Thomas James Hammond**

**Student number 202112533**

**Word count: 1540**

Table of Contents

[Software 3](#_Toc152322720)

[Requirements needed from the software 3](#_Toc152322721)

[Primary 3](#_Toc152322722)

[Secondary 3](#_Toc152322723)

[User interface 3](#_Toc152322724)

[Requirements of the UI 3](#_Toc152322725)

[UX design 3](#_Toc152322726)

[Design Images 4](#_Toc152322727)

[Priorities of the Software 7](#_Toc152322728)

[Goals 7](#_Toc152322729)

[Holdbacks 7](#_Toc152322730)

[Hardware 8](#_Toc152322731)

[Requirements needed from the hardware 8](#_Toc152322732)

[Display types and choices 8](#_Toc152322733)

[Single board computer systems available 9](#_Toc152322734)

[Choices 9](#_Toc152322735)

[Processor 9](#_Toc152322736)

[RAM 9](#_Toc152322737)

[IO ports 9](#_Toc152322738)

[Pros and cons of each system 9](#_Toc152322739)

[Additional features 10](#_Toc152322740)

[Priorities of the Hardware 10](#_Toc152322741)

[Goals 10](#_Toc152322742)

[Holdbacks 10](#_Toc152322743)

# Software

## Requirements needed from the software

### Primary

* The software needs to be responsive and work in real time to ensure that the display is not lagged behind the real-world events; this is especially important for the gauges within the software as the user needs to see the values in real time.
* The software needs to be able to provide the user with a clear display of buttons and gauges which can be viewed and pressed.
* The buttons on the display should be large enough the user can see and use while moving without needing to look at the display for long periods of time.
* The software should be compatible with the major car groups and not vehicle specific.

### Secondary

* The software should be able to allow the reading of error codes which can then be displayed to the user.
* The gauges and switches can have the backgrounds changed to one of the users choosing.
* The software should start up in a reasonable time to ensure that the user is not waiting long periods of time before being able to use the device/system.
* The software could display a waring for when the car is moving.
* The software could darken when at night so it’s not too bright for the user.

## User interface

### Requirements of the UI

#### Primary

* Should be minimalistic so it is not distracting.
* Should have large buttons.
* Should have large gauge panels.
* Should be useable with minimal user training.
* Should update in real time.

#### Secondary

* Should darken when at dusk/night.
* Should warn the driver not to use when moving.

### UX design

### Design Images

#### Iteration 1

This is the first iteration of the design images which features a basic design but missing some key features such as menu buttons. This is a very basic display and has not taken into account any input data that would need to be used in order to display the gauge. This iteration uses old style analogy gauges to display the speed, RPM, Fuel, Coolant and Oil

A screenshot of a computer

Description automatically generated

A close-up of a speedometer

Description automatically generated

#### Iteration 2

This is the second iteration of the gauge cluster has now changed to a digital format which suits more modern cars and makes older cars look a bit more modern. The gauge cluster shows the same gauges as before and now has a logo in the top right which can be changed in the menu settings. The physical buttons (Fan and Kill switch) have been removed form the cluster as they were not very big and hard to press and having a kill switch in the main display could pose a safety concern if pressed accidently.

A digital gauge with numbers and a green screen

Description automatically generated with medium confidence

#### Setting Page Iteration 1

This is the initial design for the menu that will allow the user to change various features within the dash. It shows the ability to change the gauge faces, background, brightness, startup logo, haptic feedback and Bluetooth device.

A screenshot of a computer settings

Description automatically generated

#### Setting Page Iteration 2

The second iteration of the settings page was designed after researching possible menus and finding out what worked best for the display. Menu drop down boxes did not work well with the touchscreen display therefore had to be changed to clickable buttons for easier use. A update button was added to confirm the changes when the user was done and then a return to dash button so user can get back to the dash display.

A screenshot of a computer

Description automatically generated

## Priorities of the Software

* Display gauges
* Display switches
* Update in real time

## Goals

The goal of the software would be to provide the user with a display which can display digital gauges which update in real time as the car is moving such as showing the current RPM of the engine. The software should also be able to allow the user to click virtual buttons which can then control various systems in the car such as an AUX fan switch. The system should also be able to be used on multiple vehicles not just one brand which will make it almost universal and gives it the ability to be used by anyone no matter what car they own.

## Holdbacks

The system that is used to communicate data from the car to the device would need to be coded to work for various different vehicles as each system is different and requires different things to access the data.

For the system to adjust between a light and dark mode it would need a light sensor to be implemented on the hardware side where a CAN signal is not used.

# Hardware

## Requirements needed from the hardware

#### Primary

* Double Din size screen.
* Processor speed to keep up with can system.
* Display should not have glare during the day.
* Safe shut down for the system when power is lost.

#### Secondary

* Display should be able to have a controllable brightness.
* Haptic feedback

## Display types and choices

The display needs to meet some key requirements that have previously been identified. To help meet these requirements the display can be chosen to ensure it meets these needs.

To ensure that the display is visible during the day and the glare dose not blind the user the screen should have an anti-glare layer installed which comes on most displays; alternatively, I could use a OLED display which would be even better in the day light and visible at any angle or light level but these are very high cost products.

The screen should be no larger than a double din radio screen which would ensure that it would fit in the desired vehicle and can be used to help with mounting as the double din radio is standardised. The standard double din is 7 inches by 4 inches, but this can be slightly bigger if you go outside the socket.

The display needs to have a controllable brightness setting as when its dark the system will need to dim the screen to ensure it doesn’t distract the driver or cause any other issues. Most displays come with this feature, but some don’t so will need to make sure the desired screen includes this feature.

## Single board computer systems available

The computer that needs to be chosen needs to ensure it can keep up with the constant traffic of the Can system within the car as if it’s too slow this will cause the system to be non-responsive and data will be laggy. The circuit board also needs to be powerful enough to support the desired screen size of 7x4 inches. The circuit board will also need to have a built-in system to ensure that when the power is lost it can still shut down safely and not corrupt the system files.

### Choices

* Arduino
* Raspberry Pi

### Processor

The raspberry pi has a quad core CPU that is running at 1.8GHz which is a good base and will allow the system to read CAN messages as well as being able to have enough processing speed and CPU cores to be able to multitask and run any other software that might be needed.

The Arduino has a CPU a that is running at a speed of 16 MHz which allows it to complete around 16 million clock cycles per second. Although this may seam like it is running much slower the Pi it does not have to run an operating system as all the functionality is run directly from the chip

### RAM

The raspberry pi has the ability to have varying amount of RAM depending on the requirements. They are available in 1GB, 2GB, 4GB and 8GB versions which provide the project with many different variations and the ability to choose the best one available for the requirements.

The Arduino has 2,048 bytes of RAM which might seem like a small amount, but this is because it isn’t required to run a operating system like the pi.

### IO ports

The raspberry Pi has a 40-pin system which allows for multiple devices to be connected and these pins are not only able to accept input and output data but can also provide power and ground to the external devices.

The Arduino has 14 pins on the board which can be used for input or output devices this can also be expanded via external components which would allow even more input and output devices to be connected.

### Pros and cons of each system

#### Raspberry Pi

Lots of aftermarket parts such as screens, motors and other addons which would make it easy to add extra features as well as improve upon the original ideas and make it even better than expected. Similar projects have also been completed with the raspberry pi so there are plenty of potential resources to help aid development and build new ideas along the way. The Pi is also relatively small so can easily be fit into the small space behind the screen which is important due to the cars limited space. The Pi also uses a version of Linux which allows it to be very versatile and adapt to the needs of the project.

#### Arduino

The Arduino is very good for adding extra components such as lights, sensors and other exterior parameters but it lacks in the fact it has no full operating system and is primarily used to control exterior devices which would mean a lot of effort would need to be put into the project to get it working as intended. There are not a lot of resources that have been released about using an Arduino for cars and therefore would make the project longer and potentially have less features.

## Additional features

Some additional features can be added such as dimming of the display to allow it to be used easier in the dark and adding a feedback motor to allow for haptic feedback to be enabled so the driver is less distracted. A light sensor could also be added to enable the switching of day and night mode where a CAN signal from the car isn’t present.

## Priorities of the Hardware

* Low cost
* Double din standard size
* Safe shutdown

## Goals

The goal of the hardware would provide the user with a double din sized screen which can easily be read at multiple angles. The circuit board should also be small enough to fit behind the display and be no deeper than the original head unit installed. The whole system should be easily available and be budget friendly.

## Holdbacks

Some parts that are required such as the display can have defects which may require them to be sent back to the manufacturer which could delay the project.