Retrofit Car High-Level Design

Nathan Campeau, Freddy Feliciano, Josue Guerra, John Healy, Wesley Simpson

EE 41430

TABLE OF CONTENTS

1 Introduction

Most old cars need more features to catch up to modern technology. Some people may replace their working vehicles in favor of one with newer and increased features. Rather than buying a whole car, another option is to modify older cars by adding newer features. This project will create an all-in-one user interface integrating multiple sensors that can be installed in an older vehicle. This is more environmentally friendly and cost-effective than simply buying a whole new car.

2 Problem Statement and Proposed Solution

Newer cars come with many features that make the driving experience safer and more enjoyable. For example, proximity sensors and ultrasonic sensors can help reduce the risk of collisions. However, buying a new car with such features may be too expensive for some people, and someone might have a perfectly working older car that they wish they could transplant such features onto.

Our solution is a system involving multiple sensors that will provide data featured on a touch-enabled display. Our sensors and screen display will get power from the 12V cigarette lighter outlet on the vehicle. The sensors will have wired connections to feed data into a microcontroller, which will process the data and send it to the display. The display will have general metrics including outside temperature, the time, and the current speed, with advanced options that measure trip mileage or enable ultrasonic sensors. The system will also be able to connect to a smartphone over Bluetooth, and it will have an auxiliary cable to connect to the car's audio system.

3 System Requirements

Feature Requirements:

- The system should be powered by the 12V cigarette lighter outlet.
- Sensors should be able to withstand outside-of-car elements if they're outside, or be able to withstand high temperatures if near the engine. The sensor signals must be able to be sent (and have information integrity maintained) over the length of a car.
- The main microcontroller must have (or be connected to) a Bluetooth module, allowing us to wirelessly send audio from a phone or similar device.
- The UI display must be able to toggle between different layouts.
    1) The default layout will display the outside temperature, time, the speed of the vehicle, and the trip length in miles of the vehicle.
    2) The audio layout should show information from the Bluetooth source about the media being played.
    3) The proximity sensors must be toggleable on the touchscreen.
    4) The UI display should have easily adjustable brightness levels, as well as an option to turn the screen off.

Installation Requirements:

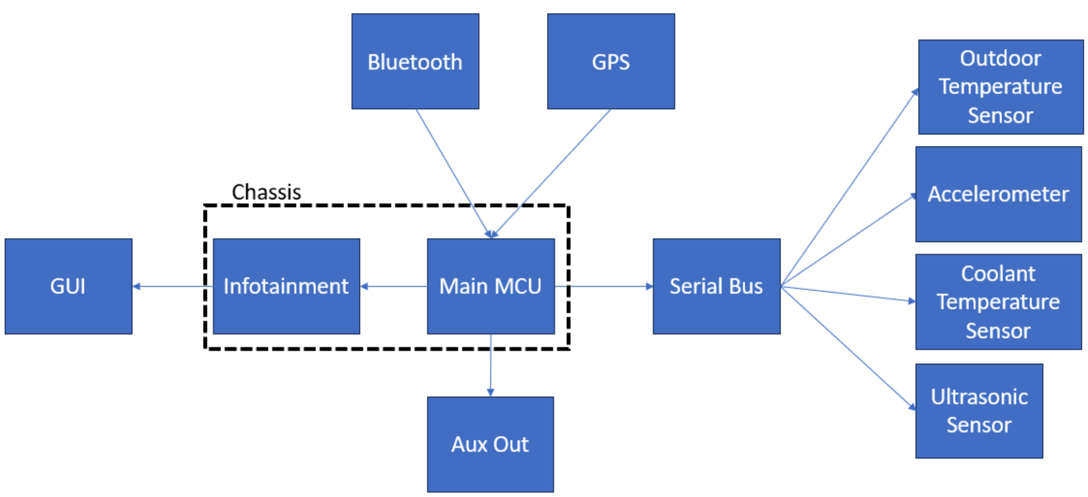
- Sensors (and their wired connections) will have to be snaked through the car. Wires must have sufficient length and ideally are easily substitutable if adjustments to wire length need to be made. Connectors should be idiot-proof so as to not be accidentally plugged in the wrong direction.
- Sensor location placement must balance cable management considerations with sensor efficacy considerations.
- The touchscreen must have a chassis that secures it somewhere over the center console. The chassis design will need to be tailored to a specific vehicle model.
- System software must be able to check whether or not sensors are connected and update the GUI appropriately if a sensor is missing.

Safety Requirements:

- Since 12V is being used to power multiple devices in this system, extra care will be required to make sure that components (wires, PCB trace width, and SMT packages) are appropriately rated.
- Car modifications must not negatively impact the car's safety and operability. Airbags should not be interfered with, internal car wiring should not be substantively modified, and the system must not cause the car battery to die.
- The touchscreen must be easy and intuitive to use in order to pose a minimal distraction to the driver. The touchscreen should be easy to click, and the information on the display should be big and easy to read.

4 System Block Diagram

4.1 Overall System:



*Accelerometer sensor is a possible future addition that may or may not be integrated depending on time constraints.

4.2 Infotainment Subsystem and Interface Requirements:

The infotainment subsystem is contained within the UI screen and it is the main bridge between the data acquired and the user. It must display all the data to the operator of the vehicle and be simple to traverse. Ideally, the operator will not have to stare at this screen while driving but only glance. The infotainment subsystem must be able to display the following information: speed, coolant temperature, exterior temperature, and humidity. If time allows, other features may be added. The infotainment UI will have 2 separate displays that can be switched by pressing a (virtual) button on the screen. One display is for speed, temperature, and normal driving needs. The other display is a music display which will show the name of the current song and allow the user to skip or pause. The infotainment subsystem requires the data to be accurate and real-time. It must also have a way to dim the screen or brighten depending on the amount of ambient light.

4.3 MCU Subsystem and Interface Requirements:

This subsystem, at a minimum, must be able to do the following: It must handle power distribution in the system as a whole, read data from the serial bus, process data from the serial bus, push data to the infotainment and interface subsystem, upkeep Bluetooth, upkeep GPS, and output Audio. For power distribution, this subsystem will be where power enters the system in totality. It will take the unregulated power from the vehicle and set up 12V, 5V, and 3.3V power lines for each part of the system that needs power. For serial communication and data processing, this subsystem will take in data over CAN and then process the data to push it over TTL serial to the infotainment and interface subsystem. For Bluetooth and GPS, this subsystem will host the components for connectivity on each of these devices and handle all of the software upkeep necessary for this functionality.

4.4 Sensor Subsystems and Interface Requirements:

This subsystem will contain information regarding the sensors that will feed into the MCU. The sensors that would fall under this subsystem are the outdoor temperature sensor, the coolant and header temperature sensors, and the ultrasonic sensors. The outdoor temperature sensor will require 3.0 to 5.5 V for power/data. It will also require only one digital pin for communication. The coolant and header temperature sensors will be similar to the outdoor temperature sensor, so they will have the same voltage requirements. Finally, the ultrasonic sensor operates between 2.5 V to 5.5 V. These sensors will require a connection to the UI of the system to communicate the data that is desired to be displayed to the user.

4.5 Future Enhancement Requirements

Future potential enhancements include an accelerometer and cellular connectivity (with the ability to make emergency calls from the vehicle). Another potential enhancement would be a mobile application through which one could store the car’s data (transmitted to the phone over Bluetooth). It would also be nice (although difficult) to integrate a backup camera.

5 High-Level Design Decisions

Infotainment Subsystem

We will use a Nextion touchscreen which comes with its processor to handle the screen's GUI. This should take some burden off of our primary MCU and make it easier to design the GUI. The touchscreen will receive data (to update the display) from the main MCU over the TTL Serial. The Nextion touchscreen that we are considering comes with NEXTION Editor software to create a GUI for the Nextion touchscreen. A chassis will be designed and 3D printed to hold the touchscreen in place. The Nextion touchscreen has mounting holes in each corner that can be utilized to secure the screen to the chassis. The chassis will likely be secured to the car by having clips that can latch onto the air vents near the center console. This part of the chassis will be designed based on the car we choose to use for our demonstration, but ideally, the chassis will have some element of adjustability in it. A digital encoder may be mounted into the chassis (and wired to the MCU system) to adjust the touchscreen's brightness.

MCU System

We will use an STM microcontroller as the heart of our system. The STM microcontroller linked in the components section supports interfacing with a CAN bus. CAN will be used to receive data from the sensors because CAN is resilient against signal disruption and is standard in the automotive industry. The microcontroller will be responsible for taking in sensor data, making calculations as appropriate, and transmitting data to the touchscreen or the audio output. The STM microcontroller will be placed on a PCB which will also be attached to the chassis holding the Nextion touchscreen. This PCB will also contain a Bluetooth module and a GPS module, as well as the power intake from the cigarette lighter and voltage regulation. Connectors will be selected and soldered onto the PCB to connect our sensors to the CAN bus. The PCB will also have an audio jack (out) to connect to the car's audio system, and the PCB may have a piezo on it (which will be used to alert the driver in accordance with the proximity sensors).

Sensor Subsystems

The proposed sensors include an outdoor temperature sensor, a coolant temperature sensor, and (2) ultrasonic sensors. These sensors will have wires running back to the main MCU to provide power and to convey signal information (over CAN). Each sensor may need to have its own microcontroller nearby on a custom PCB to convert the sensor's output into a format which can be transmitted over CAN. These microcontrollers would likely be programmable via an external debugger (with header pins) and would also have to be CAN-interfaceable.

6 Open Questions

Below are questions and problems that are intended to be addressed:

- How do we unpack the Bluetooth data and find a way to display the song information on the GUI? The song data would include the name of the song, the artist, the album, and the album cover (if applicable)
- What will be the most efficient and aesthetically pleasing way to snake the wires through the car? What kind of wires will be needed to successfully bring our idea to fruition?
- How do we use the CAN bus protocol and interface the sensors with it?
- What organization system will be best for the group's success? Examples include an agile method or a Gantt chart.

7 Major Component Costs

technology	link/info	cost
Touch Screen	Screen w/ its own upkeep	$99
Development Board (for testing)	Dev Board	$11.10
MCU	STM32F042G6U6TR	$3.16
ST-LINK	Debugger	$22.84
GPS	GPS sensor from Amazon, is in-stock	$13
Bluetooth	Bluetooth	$13.98
ultrasonic sensor (x2)	This module has a long range of about 21 feet. There are cheaper options on Digikey and Amazon which don't extend as far. Further testing is needed to determine the appropriate distance and degree. Will probably use 2 sensors on	up to $30 (x2)

	car.	
temperature sensor	many sensors are available (such as this) that can withstand high temperatures	$10

8 Conclusions

This project aims to be a streamlined package that can be added to a vehicle to enhance its features. The retrofit system will help motivate people to extend the use of their older cars, which will help them save money. For testing and demonstration, we will install this system into an actual car.