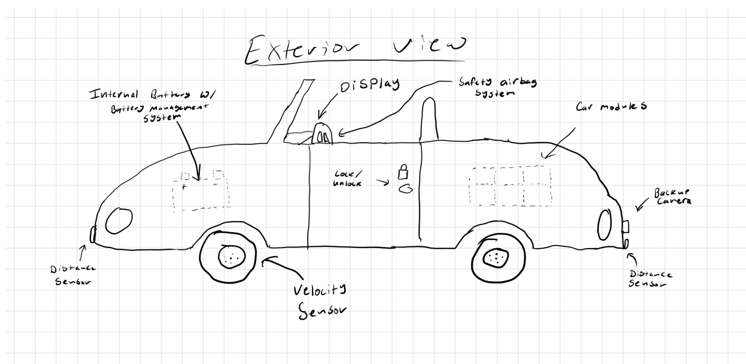
Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Fall 2023



CAN on Wheels
Modular EV

Destiny Rogers
Leonardo Ibarra
Carson Fabbro
Edwin Diaz
Hector Sosa

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.03.2023	CF	document creation
0.2	10.06.2023	CF, LI, HS, DR	complete draft

Contents

List of Figures

1 Problem Statement

A few decades ago, vehicles were just a means of transportation, and the mechanisms that made the vehicles run were relatively simple when compared to the modern counterparts. As the rise of concern for emissions and fuel efficiency rose, the simple mechanical systems that made vehicles run became insufficient. Therefore, the need for computers in vehicles arose in order to keep up with the rising demands of lawmakers and emission regulations. However, vehicles are now becoming more than just a means of transportation. They contain an infotainment system, sensors, drive assist, and more recently, a push for autonomous driving. With these recent demands, the need for more computers and high speed data transfer within the local networks of the car has risen. Just a decade ago, cars were running 15 million lines of code in their computers, but today that number has risen tenfold to 150 million lines of code. The current methodology for car manufacturers is to outsource the software development to third party companies who may not completely know how car manufacturers will combine their developed modules with others. With current trends, many car manufacturers must redesign the whole network architecture of their cars to be more focused on software and modularity. This would allow for added levels of abstraction of software, which will then allow for a platform for software developers to develop applications that tailor more towards the user, which will help developers from worrying about the hardware aspect of the vehicle.

2 Methodology

Our team will design a modular electric vehicle with plug-and-play hardware modules. This would serve as an educational tool for aspiring engineers and provide a method for rapid prototyping in the automotive industry.

3 Value Proposition

This vehicle platform will prove valuable to vehicle manufacturers because at the current pace of the market, they need to start transitioning more of their development time to the software since more and more the vechicle's value is dependent on the implementation of the software. If they have a simple testing platform of which they can put and take away modules at will, the development process will be more streamlined. Since this vehicle will be modular, when the time arrives to implement more computationally demanding features such as computer vision and autonomous driving, the platform provides a way to plan out the architecture that will fit the higher demands.

Another use case of this vehicle platform, will be for institutions can make use of this vehicle platform to future engineers, and have it as a platform of which they can conduct labs and activities for their courses. Employers desire experience over theory, therefore, if institutions adopt this electric vehicle platform, it will be a learning tool. Also, this also opens up the doors for research and development outside of the company itself. Hobbyist and other like minded people can go and develop their own modules for this platform.

4 Development Milestones

This list of core project milestones should include all major documents, demonstration of major project features, and associated deadlines. Any date that has not yet been officially scheduled at the time of preparing this document may be listed by month.

Provide a list of milestones and completion dates in the following format:

- Project Charter first draft - October 6th 2023
- System Requirements Specification - October 24th 2023

- Architectural Design Specification - November 17th 2023
- Completion of all individual modules - February 2024
- Detailed Design Specification - February 2024
- Finalization of vehicle chassis - March 2024
- Integration of individual modules - March 2024
- Demonstration of modular system - April 2024
- CoE Innovation Day poster presentation - April 2024
- Demonstration of hardware integration - April 2024
- Final Project Demonstration - May 2024

5 Background

Many popular vehicle manufacturers put a major focus on hardware rather than software, but as hardware innovation plateaus, consumer focus is shifting toward better software. With modular technology, giants in the automotive industry can transition to a more software-heavy approach like new companies in the automotive space. A small scale electric vehicle platform by which engineers, car manufacturers, and future embedded programmers can test their vehicle modules in various configurations would allow for manufacturers to program their modules in house and see their behavior instantaneously. If there is a specific recall, the vehicle which has specific configuration can be replicated in our module, of which it can have its modules replaced. This vehicle platform can serve to diagnose which of the modules is misbehaving via a port or screen. In the future, the internal communication protocol can be replaced to reflect the changes of the vehicle industry which demands higher data transfer rates of which ethernet can be capable to accomplish.

6 Related Work

As opposed to removable modules, most hardware components in the automotive industry must be tested with breadboards and external circuitry [2]. This can be time-consuming and inefficient as opposed to the modular idea we are proposing. Although older manufacturers put more focus on hardware, newer companies put more focus on software [3]. Software integration is more prominent now more than ever, and as a result, older manufacturers may find it necessary to rapidly experiment with and expand upon their own software. With a modular design, this should be much simpler and streamlined. Finally, current methods of inter-processor communication in vehicles are becoming obsolete and can no longer carry enough data for modern vehicles [1]. With our platform, replacing such outdated modules should be highly efficient. A faster, more efficient network can be installed in new vehicles quickly.

7 System Overview

To solve the issues that car manufacturers are facing today, we plan to create a modular electric vehicle, that can be used as a platform for testing various modules. While our project itself will not be extremely valuable to these manufacturers, the concept could be extremely useful. This is because car manufacturers outsource 90% of their software, which include the actual modules within the car themselves. As a result of this, car manufacturers are abstracted away from the software that run these modules, and face compatibility issues between the components themselves. This is where our project idea could really prove useful.

The main focus of our electric vehicle, will be in its modular design. We plan to utilize a Power Wheels car as a base, and intertwine its functionality with our devices. Additionally, we plan to make these devices as plug-and-play modules, that will allow for easy insertion and deletion of functionality. Finally, to tie it all together we plan to have a center screen for interfacing with these devices, which will provide vehicle information, test tools, module status, current time, and potentially power consumption. By doing this, the car will serve as an excellent testing platform, that is very user friendly and adaptable.

Some of the major features we plan to add to the car are: a center display, front and rear object sensors, rear backup camera, security-lock system, immobilize-security system, collision detection sensors, collision safety, battery management system, and a speedometer. Additionally, we want all of our devices to communicate through the CAN communication protocol. This will simulate how a real vehicle operates, which will prove to be an extremely useful testing and learning tool.

The way that a user would interface with our project, would be very hands on. It would require inserting and removing modules from the chasis, and scrolling through the digital display. Ideally, it will be very user friendly, and easy to understand.

Finally, our system will be designed to be adaptable. We want it to be very receptive to external systems, making it seamless to implement new modules. The diagrams of the project are shown below.

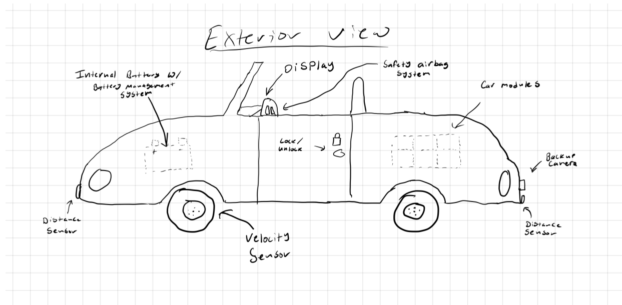


Figure 1: Exterior view of our EV

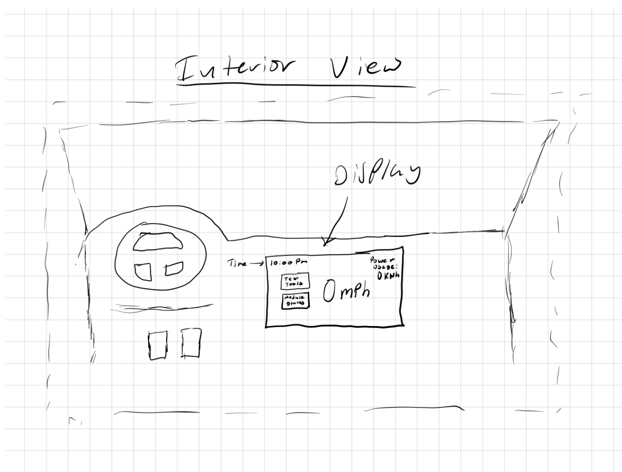


Figure 2: Interior view of our EV

8 Roles & Responsibilities

The stakeholders of our project is composed entirely of our sponsor, UTA. They are the ones funding this project, and the outcome of our project will effect only UTA and ourselves. The team members of this project include Carson Fabbro, Leonardo Ibarra, Hector Sosa, Destiny Rogers and Edwin Diaz. We are all computer engineers, so our strengths are very balanced between hardware and software.

Destiny, Carson, Leonardo, and Hector are all taking a course in Real Time Operating Systems, while Edwin has taken a course in Wireless Communication Networks. Additionally, Hector, Carson and Destiny are taking a Mechatronics course as well. This makes them very desirable when it comes to designing the battery management system, motor drivers as well as the real-time operating system that will be used to manage the systems in the car. Leonardo is currently taking a Robotics class, which makes him more equipped to manage the vehicle movement systems and software. Finally, Edwin is very passionate about the infotainment of our design, and will be working on the signal processing, and potentially battery management of our vehicle.

Our team plans to rotate the role scrum master, while keeping Hector as the product owner because he is the one created the idea. We believe that this will allow us to be more adaptable, and further enhance the experience for us all. We plan to change the role of the scrum master every sprint, unless we find that one person fills the role exceptionally well.

9 Cost Proposal

For this project there are a few major expenses to note. The first and most major expense will be the vehicle frame itself. For this, we will be purchasing a Power Wheels, which is miniature car made for children. These cars come out of the box being able to drive and steer. Buying this vehicle will save us a lot of money, as well as time. Additionally, we will need to buy a display, to create an interface for the user. Furthermore, we will need to purchase distance sensors, a small camera, many micro-controllers with CAN communication, CAN trancievers, electronic components, PCB's for the modules, and collision sensors, in order to realize the functionality of our project. Finally, we will need to purchase a battery to power the car, a raspberry pi to run the display, and whatever miscellaneous parts we need along the way (screws, wires, connectors, etc.).

9.1 Preliminary Budget

Item	**Budget**
Vehicle Frame	$250
LCD Display	$50
Raspberry Pi	$100
Backup Camera	$40
Micro-controllers	$100
Sensors	$75
Battery	$50
CAN Transeivers	$10
Electronic Components	$30
PCBs	$10
Miscellaneous	$85

Table 1: Preliminary budget for our project

9.2 Current & Pending Support

The following is a list that shows all of the support given for our project:

- The University of Texas at Arlington - $800

10 Facilities & Equipment

For our project, we will need a moderate testing ground space where we could test the vehicle's movement and collision detection. Gaining access to one of the 24 hour laboratories in the Engineering Research Building will be necessary for testing. However, it would be optimal to work outside and design a pseudo testing ground for experiments and testing. Additionally, we would need a cloud-based work environment to upload and work on our software. Services such as GitHub and Jira would be the best choice depending on whether or not the GitHub repository will be private or public. For specific equipment, the laboratories and would provide multimeters and soldering stations for soldering and battery testing. In addition, special training will need to be done to access any power tools from the laboratory. Regarding the specific chassis to use for the vehicle, a power wheels frame would have to be purchased and deconstructed in order to save time and money. For a printed circuit board, there are multiple free PCB design software online that are offered. Finally, light manufacturing like 3D-printing parts will need a 3D-printer, which is offered at the FABLab at a low cost.

11 Assumptions

- A suitable outdoor testing location will be available by the instructor no later than the beginning of sprint 4
- All electronic components of the project will be available to use by the end of sprint 4
- Modular EV will be deconstructed into a workable frame by November 17, 2023
- Specialists and mentors will be offering guidance in part recommendations and features to include on the Modular EV
- Every team member will save time in their schedules for general meetings and work days on the project

12 Constraints

- Possible sponsors and donors will respond to our requests at a reasonable time
- Data transfer rate is capped at 1 Mbps for the Modular EV
- Rough design layout of Modular EV must be done by November 17, 2023
- Total development costs must not exceed $800
- Modular design must be at most one cubic yard in size

13 Risks

Risk description	Probability	Loss (days)	Exposure (days)
Availability of Raspberry pi due to manufacturing delay	0.50	20	10
Suitable testing spaces not available	0.20	10	2
Unknown faults such as producing a faulty PCB prototype	0.05	10	0.5
Delays in shipping from overseas vendors	0.10	20	2.0
Sensor durability and the chance of faulty sensors	0.15	7	1.05

Table 2: Overview of highest exposure project risks

14 Documentation & Reporting

14.1 Major Documentation Deliverables

14.1.1 Project Charter

The Project Charter will be updated every two sprints and after a major feature or setback has occurred to reflect development of the project. The initial version will be submitted by October 6th, 2023. However, the final version will be submitted by the end of the semester which is dated to be the first week of December 2023.

14.1.2 System Requirements Specification

The initial version of the SRS document will be delivered, October 24th 2023, and it will be updated as the modules are created and the projected remaining time becomes cleared. The document will be edited collaboratively using CoCalc. Some items may take longer than others. The final version will be be delivered by the first week of December 2023.

14.1.3 Architectural Design Specification

The Architectural Design Specification will be done in the second half of the development cycle once all the hardware has been attained. It will be updated when there is a realization that the current demand for work the project is too much and will be edited collaboratively using CoCalc. The initial version will be delivered November 17th, 2023. The final version will be delivered in the first week of December 2023.

14.1.4 Detailed Design Specification

In the second phase of project development, the Detailed Design Specification will be completed and edited collaboratively using CoCalc. The initial version will be delivered February 2024.

14.2 Recurring Sprint Items

14.2.1 Product Backlog

The product owner will be be able to add items to the product backlog as the project progresses. These items will be prioritized by functionality and dependencies. The software used to keep track of dependencies has yet to be determined. Jira will be used to keep track of the product backlog.

14.2.2 Sprint Planning

There will be 5 sprints from Aug. 21st to Dec. 6th and 5 sprints from Jan. 16th to Apr. 30th. Each sprint will be 2 weeks long. We will meet at the end of each sprint to reflect and plan for the next sprint.

14.2.3 Sprint Goal

The Scrum Master, which will be rotating every sprint, will decide the sprint goal. The customer will be involved by informing them of our goal via email or in person meetings.

14.2.4 Sprint Backlog

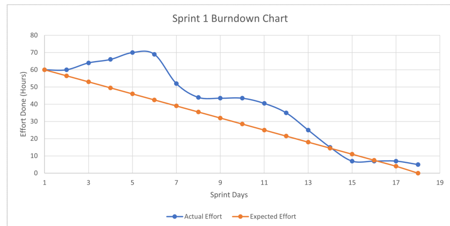
The product owner will decide which items from the product backlog make it to the sprint backlog for a given sprint. The backlog will be maintained and hosted through Jira.

14.2.5 Task Breakdown

Individual team members will volunteer from a list of broken down tasks during scrum meetings and sprint planning. We will keep track of tasks in a backlog, and task will be purposely broken down into small feasible work.

14.2.6 Sprint Burn Down Charts

The scrum master will be in charge of the burn down chart for each sprint. They will be able to make note of the time stamps given by Jira and the messages sent via our communication platform to keep track of our activity. The burn down chart will be a simple graph with two lines showing the ideal versus the actual effort.



Days	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18
Actual Effort	60	60	64	66	70	69	52	44	43.5	43.5	40.5	35	25	15	7	7	7	5
Expected Effort	60	56.5	53	49.5	46	42.5	39	35.5	32	28.5	25	21.5	18	14.5	11	7.5	4	0

Figure 3: Example sprint burn down chart

14.2.7 Sprint Retrospective

Mondays, which is when the sprint ends, will be the day to have our meeting to review our progress of the sprint. The documentation will be created as a group of individuals and will be turned in on the days presentations are due. The data will be compiled by the scrum master.

14.2.8 Individual Status Reports

Every Sunday before the closing of a sprint, we will meet to compile hours worked and conduct team member evaluations

14.2.9 Engineering Notebooks

There will be no engineering notebooks implemented but there will be a general set of notes compiled by the team members as the necessity arises.

14.3 Closeout Materials

14.3.1 System Prototype

The final vehicle prototype will contain all of the modules, and it will contain different types of configurations for each demonstration. The acceptance test will be performed by the customer to evaluate the overall performance of the project.

14.3.2 Project Poster

The project poster will include block diagrams of the overall design of the vehicle platform, features, and explanations of the protocols used. The final dimensions will be about a meter by half a meter. This poster will be delivered in the final week of April 2024.

14.3.3 Web Page

The website will contain all of the information the project poster contains, but in greater details. Examples include modules developed, the reason behind certain design details etc. It will be delivered at closeout which on the 3rd week of April 2024, and it will contain a link to a demo video.

14.3.4 Demo Video

We will demonstrate the system working by displaying the vehicle moving, inputted sensor data, safety features , and the plug and play functionality of a module.

14.3.5 Source Code

We will host a public GitHub repository, and it will act as a place to store our source code as we develop it. We will establish a step by step procedure of installation GNU, and single read me file.

14.3.6 Source Code Documentation

There will be documentation in the form of a PDF file. The standard to be employed and tools to create the documentation are yet to be determined and will be decided upon at a later date.

14.3.7 Hardware Schematics

We will be creating printed circuit boards for this project. The schematics for these circuit boards will be added to this document at a later date..

14.3.8 CAD files

It is anticipated that some 3D printed components will be necessary. These objects will be designed in OpenCAD and will be used to generate STL files.

14.3.9 Installation Scripts

Software will be provided to allow the development and use of custom plug-and-play modules.

14.3.10 User Manual

There will be a digital user manual of how to use the platform and its modules as well as how to develop modules for it. If enough time is available late in the project, there may also be a video showing how to use the modules that come with the product.

References

[1] Ondrej Burkacky, Johannes Deichmann, Georg Doll, and Christian Knochenhauer. Rethinking car software and electronics architecture, Feb 2018.

[2] Robert N. Charette. How software is eating the car, Mar 2023.

[3] McKinsey and Company, Aug 2021.