

SYSC-5709–W

Software Programming in C



Carleton
UNIVERSITY

SOFTWARE CONTROLLER FOR VEHICLE

Github Repository: <https://github.com/PCoser/Software-Programing-in-C>

Group members

- Pedro Coser
ID: 101126424
pedrocoser@cmail.carleton.ca
GITHUB ID : PCoser
- Rahul Saidha
ID: 101164212
rahulsaidha@cmail.carleton.ca
GITHUB ID : rahulsaidha
- Tarun Dwivedi
ID: 101152667
tarundwivedi@cmail.carleton.ca
GITHUB ID : Tarund95
- Unnati Thakkar
ID: 101164481
unnatithakkar@cmail.carleton.ca
GITHUB ID : unnati9979

a. Problem Statement

Thinking about the development of a new series of vehicles, an automobile company reached us out in order to ask our services for designing the software controller and the integration with all the devices of the vehicles. This series will be designed with the concept of All Drive-by-Wire technology, meaning that there will not be any mechanical linkage between the devices, all commands from the driver will be sent through communication protocol. Moreover, some autonomous functionality can also be added.

The project includes the software of the central controller, the interface of the sensors of the wheels, steering wheel, break and gas pedal, motor, gear and break. The project does not include the software controller of the motor or gear.

From our perspective, we should develop a simulator to test the system without the real vehicle.

b. Requirements

The software should be built to meet the clients needs. Some client's need concerns to the functionalities of the systems and this is included in the first release:

Throttle, break, steer, shift commands from the driver should be read by the sensors and executed for the central controller or other devices.

Other needs in this industry is concerned to the reliability of the system and will be part of the second release:

The central controller should be able to communicate with every device of the system in real-time. For that, the client requires a watch-dog function, where the central controller can track the time between requests and replies.

The devices should be able to return a fault message if the command could not be executed.

The devices should periodically send a "life" signal to the central controller to ensure it is "alive".

The client put some functionality to be added latter, this is part of the third release:

The central controller should execute an autonomous break according to the information from a front range sensor.

The central controller should execute an autonomous correction in the trajectory and alert the driver if, from the information provided by a front vision system, the car hit the edge of the lane.

Summarizing, the first release will cover the basic functionalities; the second release will cover the reliability requirements; the final release will cover additional functionalities concerned to autonomous driving.