Group Member NAMES: Ryan Sauer, Demetrius Johnson, Jonathan Schall & Olivia Pellegrini

Software Requirements Specification

Table of Contents

1.0 Introduction

1.1 Goals and objectives

The main objective of our project is to develop a drone-car collaborative model that can be used to obtain data on drone-car collaboration. The model will use a drone camera to record the surroundings of the car and notify the car of potential hazards in real time. Some basic goals of our model are:

- Accurately detect a potential hazard in multiple environments using sensors from the car and drone, particularly the vision systems (camera).
- Minimize car stop-response-latency by optimizing the communication protocol between the drone and the car.
- Maximize drone battery life.
- Develop a working product that is open source and reproducible through our documentation to serve as a future reference/baseline for further research.

1.2 Statement of scope

General Requirements of the model

- Car driving (both driverless and with driver)
- Video feed from drone and car
- Drone following car
- Analyze objects in video feeds (i.e. obstacles, cars, colors of such obstacles and cars)
- Send data from drone to car (and vice versa)
- Store system analytics on the drone and the car

1.3 Software context

We will be applying software to a drone and car for research purposes. It is not intended to be publicly accessible and only used by researchers for data creation and analysis. It is also meant to be used in a controlled environment with simple scenarios for gathering data.

1.4 Major constraints

The project needs to be completed by August 15th, 2023. This gives us approximately six months to complete the model. However, after the model is finished, we would like to use it to collect research data of our own. In order to do this the model would need to be finished before August.

Currently we are working with the PiCAR-x and the Clover Drone

We will be obtaining a few extra Raspberry-Pi’s in order to give everyone on the team the ability to work with one. Outside of this, there will be a very limited budget and it is unlikely the remaining budget can afford extra drones, cars, or expensive technology.

Additionally, drone flight regulations from University and state policies make finding and developing a more comprehensive test environment more challenging and limited.

2.0 Usage scenario

2.1 User profiles

There will be 3 main types of users:

Vehicle Administrators (Operators):
The vehicle administrators will manage the function of each vehicle, ensure its safety, and make any physical or software changes as requested by an experiment facilitator.

Data Analyst
The data analyst will recover information from the vehicles after an experiment, process the data, and create visualizations for the results.

Experiment Facilitator
The experiment facilitators will make decisions based on the data analysts findings on what parameters should be set for further experiments. They will ensure all the constants remain and variables are set and documented. The experiment facilitators should have contact with the vehicle administrators to make changes are applied to the vehicles.

2.2 User stories

As the **Vehicle Administrator**, I need to use the car and drone to run experiments. These experiments will allow my team to collect data on drone-car collaboration.

As the **Data Analyst**, I need to obtain specific information on both the drone and the car, including the drone battery life and the network status. This data can be used to further research done using the model.

As the **Experiment Facilitator**, I need to be able to modify the parameters used for the car and drone. This will allow the Vehicle Administrator to run experiments customized for the research.

2.3 Special usage considerations

- The experiments run using the car and drone must be done in a controlled environment.
- All data retrieved will be for in house use and doesn't need to be sanitized.
- The parameters set for the car and drone along with the resulting data must be kept on record for accurate analysis.
- University policy is strict around drone use and we must adhere to those policies during testing and execution.

3.0 Data Model and Description

3.1 Data Description

3.1.1 Data objects

3.1.2 Relationships

3.1.3 Complete Data Model

An UML Class model for the software is developed

3.1.4 Data Dictionary

Class	Description	General Overview of Responsibilities	Collaborators/ Relationships

4.0 Functional Model and Description

4.1 Description for Function n

Use Case: Running an Experiment
Actors: User, Car, Drone
Preconditions: The input will be a video frame captured from the drone. There is no real output, but after analyzing input a command may be sent to the car.
Triggers: The drone should always be capturing video frames and constantly be analyzing the frames.
Scenario Description: To successfully run an experiment, the drone must be able to capture video frames and analyze a frame. Once a target is detected, a command should be sent to the car to perform an action.
Post Description: We have limitations on the environment our group can work in. This is due to university and state policies about handling drones.
Exceptions: Our drone and car should be able to function in all circumstances and constantly collect metrics and analyze captured video frames.

Use Case: Change Car and Drone Parameters
Actors: User, Car, Drone
Preconditions: The input is parameters from the user. There is no real output, but the drone and car will receive these parameters.
Triggers: In order for this use case to take place, a user must intervene and change the parameters using the command line.
Scenario Description: The user will change parameters using the command line to run experiments with the drone and car.
Post Description: Commands must be correctly formatted to properly function.
Exceptions: System should only take valid parameters.

Use Case: Retrieve Data
Actors: User, Car, Drone
Preconditions: The input will be metrics received from the drone and the output is the information that will be recorded in the files.
Triggers: Log files will be updated when actions are completed by the drone and car. Data files will collect updates about the battery life of the drone and the network status.

Scenario Description: Metric should be taken from the drone and car constantly. Data will be sent to the log file when an action is executed by the drone and car.
Post Description: There are no design constraints for retrieving data in our project.
Exceptions: Due to limited battery capacity of the drone, the drone may need to choose between longer flight time and the accuracy and timeliness of the data. To avoid this, our project aims to offload the time-insensitive tasks to car-mounted computers to avoid draining the drone’s battery.

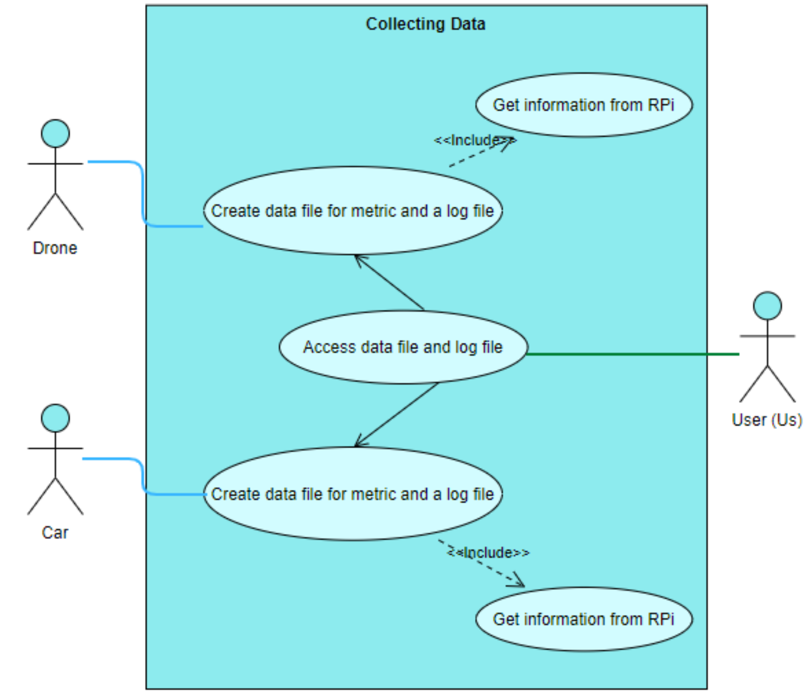
4.2 Software Interface Description

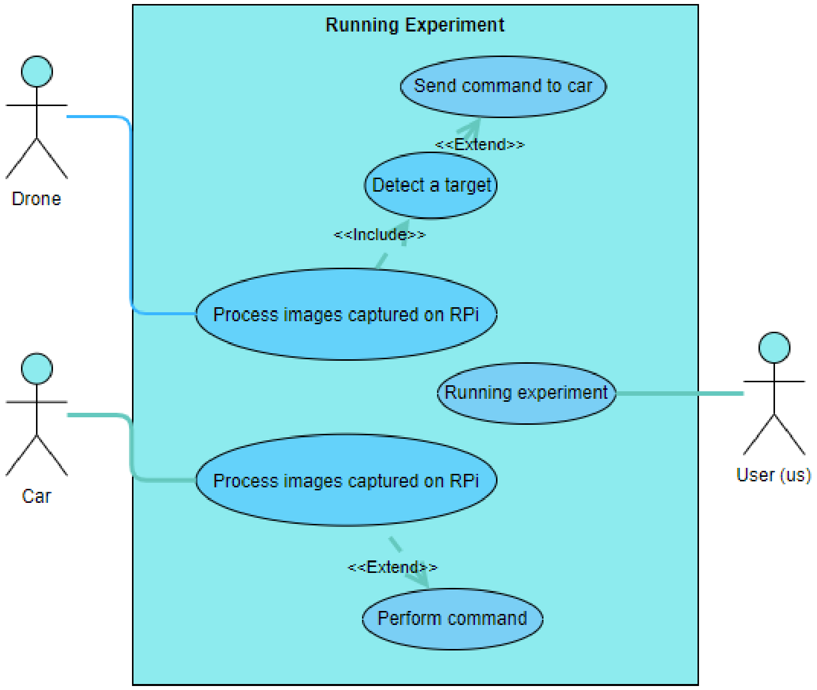
4.2.1 External Machine Interfaces

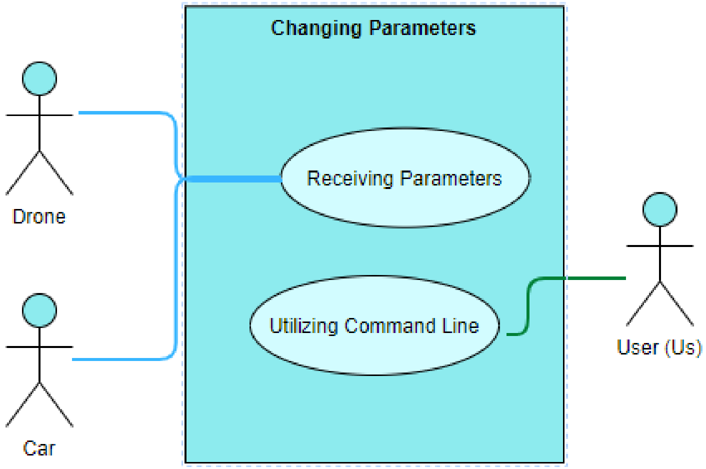
4.2.2 External system interfaces

4.2.3 Human Interface

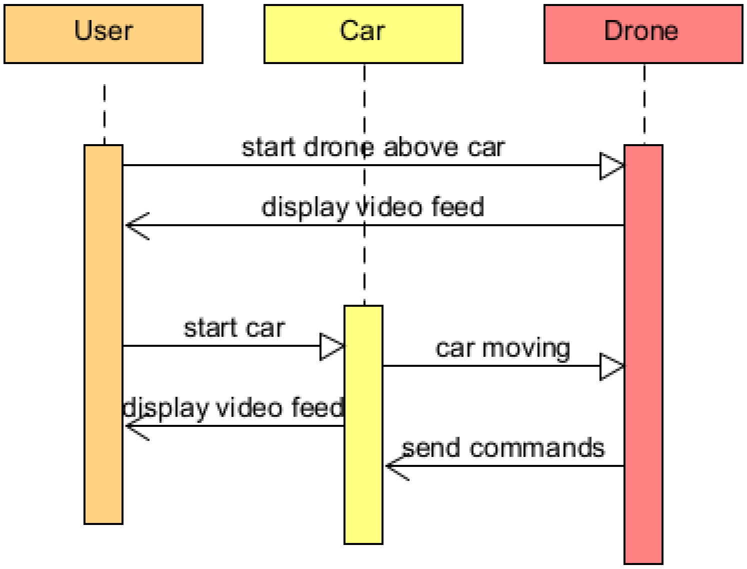
4.3 Use Case Diagrams

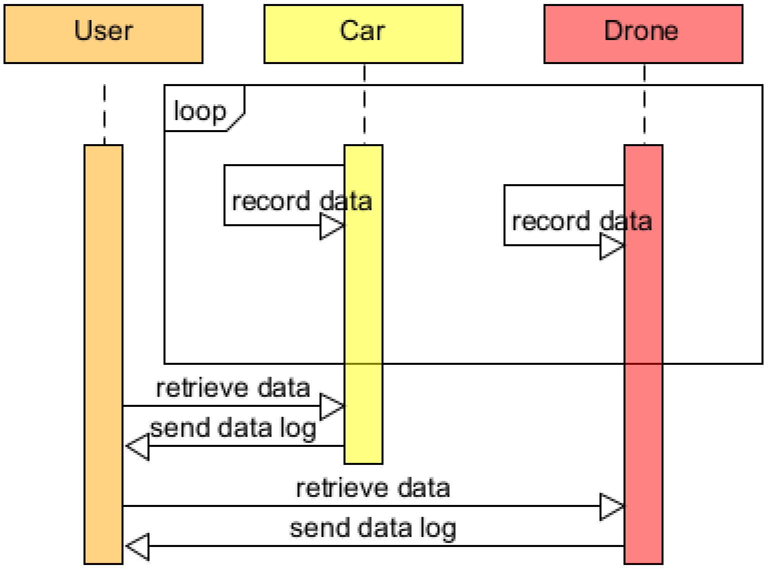

Collecting Data
Get information from RPi
<<Include>>
Create data file for metric and a log file
Drone
Access data file and log file
User (Us)
Car
Create data file for metric and a log file
<<Include>>
Get information from RPi

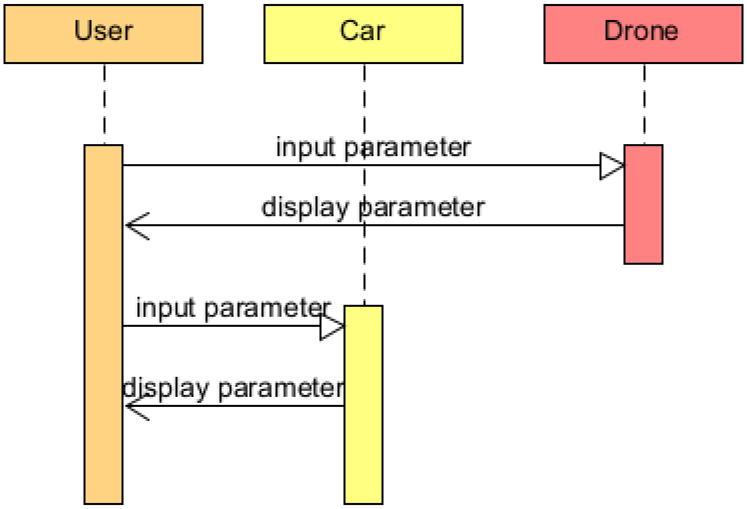



Changing Parameters
Receiving Parameters
Utilizing Command Line
Drone
Car
User (Us)

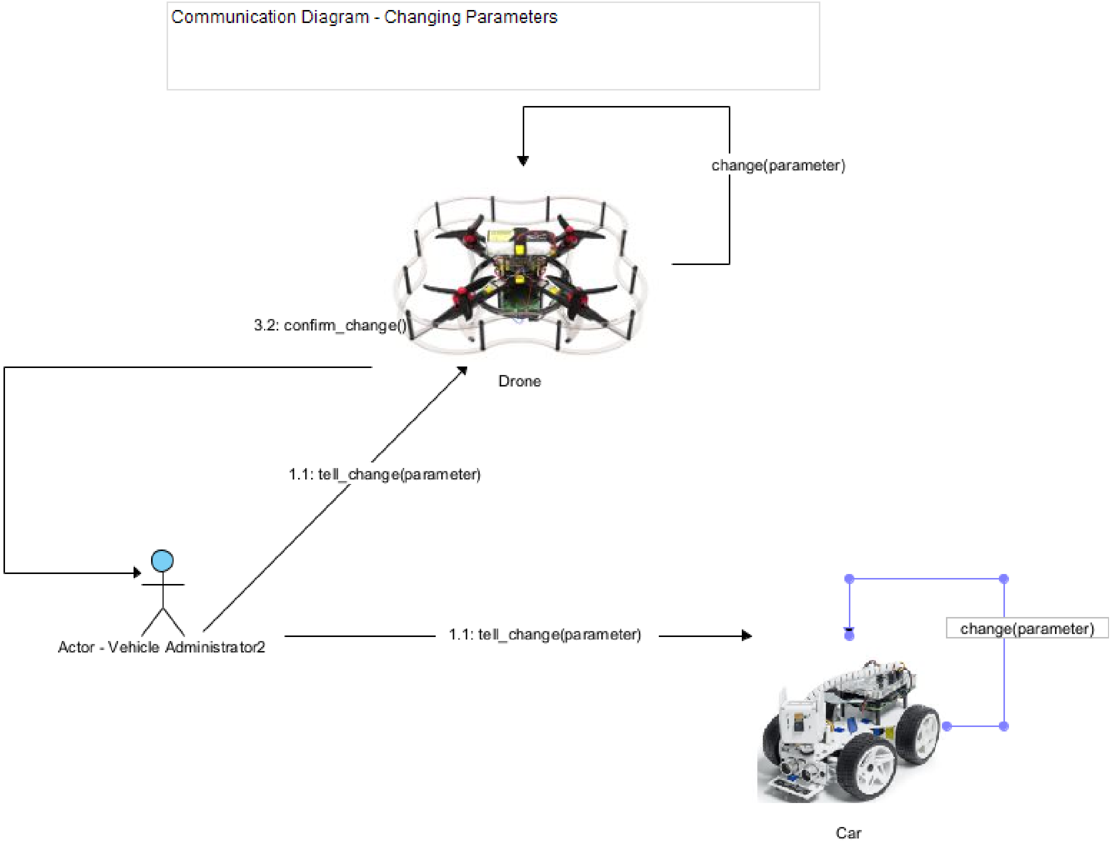
4.3 Sequence Diagrams



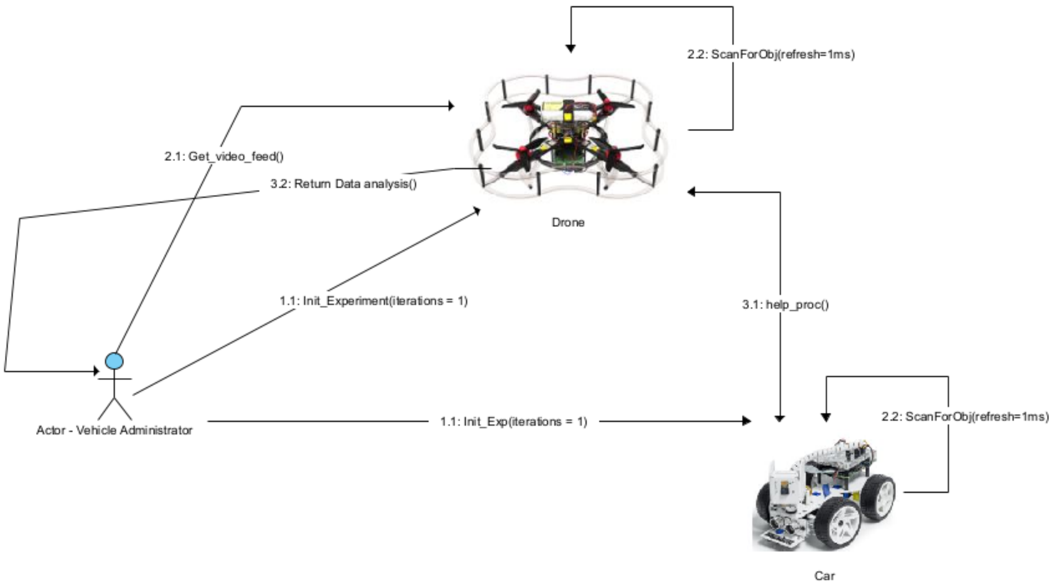




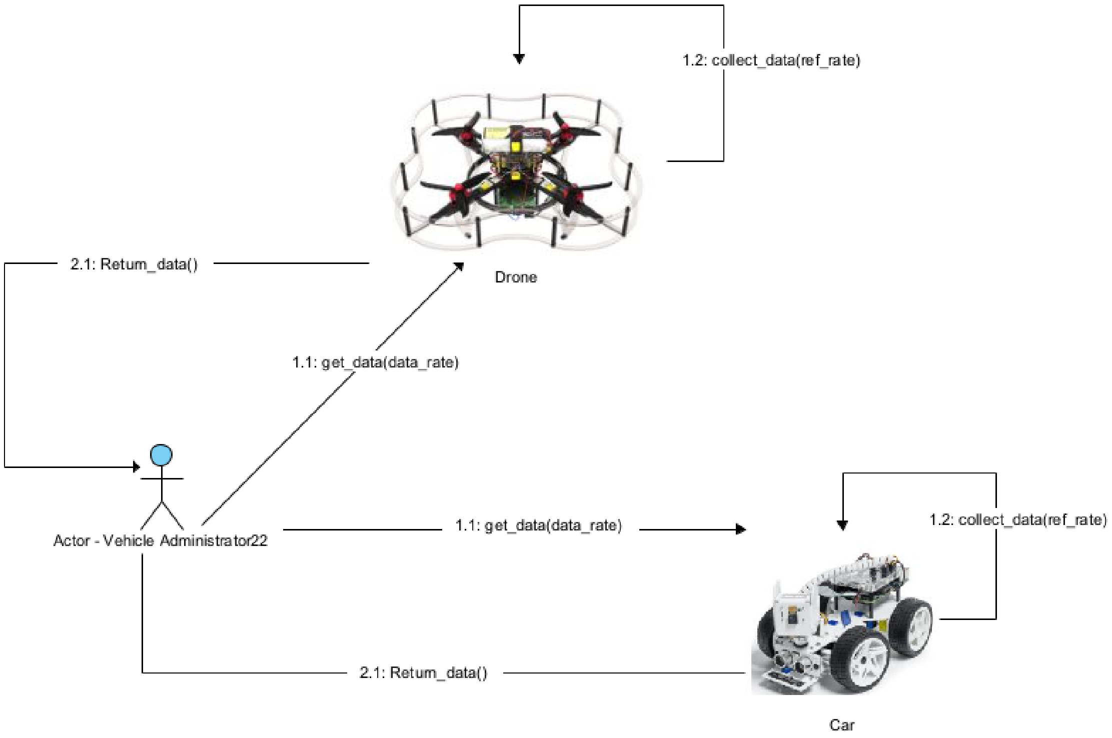
4.3 Communication Diagrams


Communication Diagram - Changing Parameters
change(parameter)
3.2: confirm_change()
Drone
1.1: tell_change(parameter)
Actor - Vehicle Administrator2
1.1: tell_change(parameter)
change(parameter)
Car

Communication Diagram - Running the Experiment


2.2: ScanForObj(refresh=1ms)
2.1: Get_video_feed()
3.2: Return Data analysis()
Drone
1.1: Init_Experiment(iterations = 1)
3.1: help_proc()
Actor - Vehicle Administrator
1.1: Init_Exp(iterations = 1)
2.2: ScanForObj(refresh=1ms)
Car

Communication Diagram - Collecting Data



5.0 Behavioral Model and Description

5.1 Description for Software Behavior

5.1.1 Events

A listing of events (control, items) that will cause behavioral change within the system is presented.

5.1.2 States

A listing of states (modes of behavior) that will result as a consequence of events is presented.

5.2 State Transition Diagrams

Depict the manner in which the software reacts to external events.

5.3 Activity Diagram

Depict the manner in which the software reacts to internal events.

6.0 Restrictions, Limitations, and Constraints

7.0 Validation Criteria

Test Case	Test Name	Preconditions	Steps	Expected Output
1			1.	
2			1.	
3			1.	
4			1.	
5			1.	
6			1.	
7			1.	
8			1.	
9			1.	
10			1.	

8.0 Appendices

Presents information that supplements the Requirements Specification

8.1 System traceability matrix

A matrix that traces stated software requirements back to the system specification.

ID	Use Case	Requirements	Priority	Depends on (ID)
1	**Run an Experiment**	• **Car driving** • **Drone video** • **Drone following car**	**HIGH**	
2	**Retrieve and Analyze Data**	• **Develop API to collect data** • **Route the data to another API for analysis**	**HIGH**	**1,4**
3	**Change Car and Drone Parameters**	• **Develop API to change variable values**	**MEDIUM**	**1**
4	**Collect/track battery voltage of car and drone**	• **System voltage measurement interface**	**MEDIUM**	**1**

9.0 References

Hardware and Software Materials and Requirements

- **Raspberry Pi model 4B**
  - https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
  - Used as the onboard computing device for both the car and the drone (each has their own Raspberry Pi computing device) for autonomous driving/flying by taking input from the sensors and processing the information, and also for communications (networking) with between devices (vehicles), such as wireless communications, i.e. WIFI.
- **Raspberry Pi Ai Car Kit (PiCar-X) for Intermediate**
  - https://www.sunfounder.com/products/picar-x
  - https://docs.sunfounder.com/projects/picar-x/en/latest/introduction.html
  - Used as the physical device for the car including motors, frame, driving mechanism, and sensors.
  - The Raspberry Pi OS imager should be used to image the sd card that will serve as the nonvolatile memory unit the Raspberry Pi computer of the car:
    - https://www.raspberrypi.org/software/
  - Here is the Repository that is cloned onto the Raspbian OS image of the Raspberry Pi; it will contain all of the installation files needed to program and control the car:

        - https://github.com/sunfounder/robot-hat
    -

- **Clover Drone 4.2**
    - https://clover.coex.tech/en/
    - Used as the physical device for the drone including motors, frame, propellers, sensors, Electronic Speed Controllers (ESC), GPS, etc.
    - Includes Pixracer R15 Mini Pixracer Autopilot Xracer FMU V4 V1.0 PX4 Flight Controller
        - https://docs.px4.io/main/en/flight_controller/pixracer.html
- **Python Programming Language**
    - https://www.python.org/
    - We will use the Python programming language for both the car and the drone.
- **OpenCV-Python Library**
    - https://pypi.org/project/opencv-python/
    - This is a Python vision analysis library that has been adapted from a library originally written for C++
    - We will use it to analyze vision data collected from cameras on both the car and the drone.
- **Raspbian OS builds with Linux Kernel**
    - https://www.kernel.org/
    - Both the car and the drone have their own onboard computing device (Raspberry Pi model 4B) with a custom modified version of the Raspbian Operating System image that uses the Linux Kernel.
    - Additionally, the Clover 4.2 Drone uses the ROS robotic framework used for advanced robotic distributed systems.
        - https://wiki.ros.org/
    - Here is the image used for the Clover 4.2 Drone Raspberry Pi computer:
        - https://github.com/CopterExpress/clover/releases/tag/v0.23
        - Image features:
            - Raspbian Buster
            - ROS Noetic
            - Configured networking
            - OpenCV
            - Mavros
            - Periphery drivers for ROS (GPIO, LED strip, etc)
            - Aruco_pose package for marker-assisted navigation
            - Clover package for autonomous drone control
- **Q Ground Control**
    - https://docs.qgroundcontrol.com/master/en/
    - This is an open source software used to communicate with and calibrate and configure a drone's flight controller firmware. We will use this to calibrate the drone and manage the flight controller's parameters and how the flight system of the drone uses and responds to sensor data.
    - Here is the firmware image used for our flight controller:
        - https://github.com/CopterExpress/Firmware/releases/tag/v1.8.2-clover.13
- **Clover Drone Simulation virtual machine (VM) image**

  - https://github.com/CopterExpress/clover_vm
  - This is the virtual machine image used to run the simulation software used to simulate programmed autonomous flights for the Clover 4.2 Drone.
  - Image contains:
    - Ubuntu 20.04 Focal.
    - ROS Noetic.
    - PX4 autopilot, QGroundControl.
    - Preinstalled Clover and Clover simulation packages.
    - Shortcuts for running Clover simulator.
    - VSCode.
    - Useful robotics-related software.
- **Drone Simulation Environment (Using Gazebo software)**
  - The simulation environment is based on the following components: Gazebo, a state-of-the-art robotics simulator;
    - http://gazebosim.org/
  - PX4, specifically its SITL (software-in-the-loop) components;
    - https://px4.io/
  - sitl_gazebo package containing Gazebo plugins for PX4;
    - https://github.com/PX4/sitl_gazebo
  - ROS packages and Gazebo plugins
  - **Note:** all of the above components are installed on the Clover Drone Simulation VM in order to do simulation programming without the Raspberry Pi on board computing device of the drone. This allows for programming and simulation without needing the physical drone present.
- **Etcher - Flashing Software**
  - https://www.balena.io/etcher
  - This is the software used to flash the micro-SD card with the respective OS (drone or car) used as the nonvolatile memory unit for the Raspberry Pi computers.

Research Background Sources

1. S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J. -C. Cano and P. Manzoni, **"Evaluating UAV-to-Car Communications Performance: Testbed Experiments,"** 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 2018, pp. 86-92, doi: 10.1109/AINA.2018.00025.
   - **Abstract:** Vehicular networks are gradually emerging due to the expected benefits in terms of enhanced safety and infotainment services. However, outside main metropolitan areas, little infrastructure currently deployed, which may hinder these services. To mitigate this problem, Unmanned Aerial Vehicles (UAVs) are envisioned as mobile infrastructure elements, supporting communications when fixed infrastructure is missing. This way, in emergency situations, UAVs can offer services to vehicles including broadcasting alerts or acting as message relays between ground vehicles. Our work attempts to be a first step in this direction by presenting experimental measurement results regarding communications quality between cars and UAVs. In particular, we varied the altitude of the drone and its antenna orientation, and the car's antenna location to assess their impact on performance. Based on the experimental results achieved, we find that UAVs communicating in the 5 GHz band using IEEE 802.11 technology are able to deliver data to moving cars

within a range of more than three kilometers, achieving more than 0.5 of packet delivery ratio up to 2.5 kilometers under the optimal configuration settings.
   - **URL:** https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8432227&isnumber=8432202

2. J. Yoon, I. Kim, W. Chung and D. Kim, **"Fast and accurate car detection in drone-view,"** 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea (South), 2016, pp. 1-3, doi: 10.1109/ICCE-Asia.2016.7804775.
   - **Abstract:** With the development of drones, aerial images are used in a variety of applications. We propose a way to detect cars in drone-view fast and accurately. For this purpose we propose a feature called G-ORF for effective feature description. Also we designed a pose classifier and bin-specific weighted Linear Discriminant Analysis (wLDA) classifier for pose classification and binary classification of each pose respectively. Our method showed real-time performance in HD (1280×720) video in a PC environment with high detection accuracy.
   - **URL:** https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7804775&isnumber=7804716
3. Yildiz, Melih, Burcu Bilgiç, Utku Kale, and Dániel Rohács. 2021. **"Experimental Investigation of Communication Performance of Drones Used for Autonomous Car Track Tests"** Sustainability 13, no. 10: 5602.
   - **Abstract:** Autonomous Vehicles (AVs) represent an emerging and disruptive technology that provides a great opportunity for future transport not only to have a positive social and environmental impact but also traffic safety. AV use in daily life has been extensively studied in the literature in various dimensions, however; it is time for AVs to go further which is another technological aspect of communication. Vehicle-to-Vehicle (V2V) technology is an emerging issue that is expected to be a mutual part of AVs and transportation safety in the near future. V2V is widely discussed by its deployment possibilities not only by means of communication, even to be used as an energy transfer medium. ZalaZONE Proving Ground is a 265-hectare high-tech test track for conventional, electric as well as connected, assisted, and automated vehicles. This paper investigates the use of drones for tracking the cars on the test track. The drones are planned to work as an uplink for the data collected by the onboard sensors of the car. The car is expected to communicate with the drone which is flying in coordination. For the communication 868 MHz is selected to be used between the car and the drone. The test is performed to simulate different flight altitudes of drones. The signal strength of the communication is analyzed, and a model is developed which can be used for the future planning of the test track applications.
   - **URL:** https://doi.org/10.3390/su13105602
4. Barbeau, Michel, Joaquin Garcia-Alfaro, and Evangelos Kranakis. 2022. **"Research Trends in Collaborative Drones"** Sensors 22, no. 9: 3321.
   - **Abstract:** The last decade has seen an explosion of interest in drones—introducing new networking technologies, such as 5G wireless

connectivity and cloud computing. The resulting advancements in communication capabilities are already expanding the ubiquitous role of drones as primary solution enablers, from search and rescue missions to information gathering and parcel delivery. Their numerous applications encompass all aspects of everyday life. Our focus is on networked and collaborative drones. The available research literature on this topic is vast. No single survey article could do justice to all critical issues. Our goal in this article is not to cover everything and include everybody but rather to offer a personal perspective on a few selected research topics that might lead to fruitful future investigations that could play an essential role in developing drone technologies. The topics we address include distributed computing with drones for the management of anonymity, countering threats posed by drones, target recognition, navigation under uncertainty, risk avoidance, and cellular technologies. Our approach is selective. Every topic includes an explanation of the problem, a discussion of a potential research methodology, and ideas for future research.

- **URL:**
  https://doi.org/10.3390/s22093321