**ÇANKAYA UNIVERSITY**
**FACULTY OF ENGINEERING**
**COMPUTER ENGINEERING DEPARTMENT**

# CENG 407

# SOFTWARE DESIGN DESCRIPTION REPORT


| Prepared By | ID |
|---|---|
| Aleyna DEDE | 201711019 |
| Ali BOZDOĞAN | 201717009 |
| Eylül ERDOĞAN | 201711025 |


**27.12.2021**

# Contents

# 1. Introduction

This Software Design Description Report provides comprehensive information on what kind of system software Autonomous Vehicle should include. This document includes the working principles of the methods that we will develop. Moreover, it mentions knowledge regarding designs of the android application. In our project, we will build our system using both hardware and software techniques. For software part, we will be using C/C++ to move our vehicle in both manual and self-driving modes. For that purpose, we use Arduino IDE. To be able to operate our car in manual mode, we aim to develop our own software which receives real-time video and capable of moving car from remote control. This software application may be written in Java, C# or HTML/CSS. For real-time object detection, we decided to use YOLO algorithm which provides fast results. In addition, accuracy of this algorithm satisfies the requirements, hence, it is preferred for our project. For hardware part, we will use variety of modules and tools which are detailed in 2.4.Hardware Design section.

In the future process of our development, these techniques can be changed based on test results we received. Therefore, we will decide step by step as we receive outputs of our system.

## 1.1. Purpose of This Document

This Software Design Document provides a description of the technical design for Kaşif UGV (Unmanned Ground Vehicle). The main purpose of this document is to explain the technical vision of how requirements in SRS document will be fulfilled. Additionally, it will provide an overview of the system’s design arhitecture. This document explains what is to be built and how can we built the system. Therefore, document contains all the interfaces, diagrams and interactions that we used during the development process of the project.

## 1.2. Scope of the Project

This project is aimed to design a fully autonomous vehicle and to develop it in order to protect the security of life of law enforcement officers and soldiers from external dangers. Our project has both software and hardware. It is aimed to produce a final product containing hardware and software by making simulations in appropriate environments. It needs to be equipped with a powerful processor and a good camera to detect hazards.

Our project has the following features:

- Object detection and tagging
- Notify the host when a malicious object is detected
- Ability to work with 24/7 patrol method
- Fully autonomous
- Ability to work synchronously with the main system
- Emergency response

In order to be successful in the project, the software and hardware parts must work in sync with each other. The final product needs to be tested in a real environment after the simulation phase.

## 1.3. Glossary (Definitions, Acronyms, and Abbreviations)

| TERM | DEFINITION |
|---|---|
| Actor | An actor can be a user, or another software system that interacts with the system. |
| Android | A name of the mobile operating system made by an American company; Google. |
| C/C++ | C and C++ high-level and general-purpose programming language. C is a structured programming language while C++ is an object-oriented programming language. |
| Python | Python is another high-level programming language that is preferable when Artificial Intelligence algorithms are used. |
| IOS | A mobile operating system created and developed by Apple. |
| Software Design Description (SDD) | A document that provides a comprehensive description of the system's design, requirements, and conditions to be able to perform methods. This document is an SDD document. |
| User | Users will be students who are participated to competition using our online platform and they will find solutions for the given problem as teams. |
| Autonomous | Autonomous means independent. In our project, we call our system autonomous vehicle since it does not need any human intervention to be able to move. |
| Object Detection | Object Detection is a popular computer science technology. By using vision techniques, |

| | |
|---|---|
| | computer decides and identifies what the object is. |
| UGV | UGV refers to Unmanned Ground Vehicle. It is a vehicle that can move on the ground by itself. There are various of UGVs used for different purposes. In this project, we designed our UGV which will be for surveillance purposes. |
| GUI application | GUI refers to Graphical User Interface. It provides an interface in which users can interact with existing software. In our project, we are planning to implement a GUI application that visualizes the operations of our system. |
| Arduino | Arduino is an open-source microcontroller board. We use an Arduino board to upload UGV's movement-related code. |
| Arduino IDE | Arduino board must be programmed via its IDE which is called Arduino IDE. Therefore, Arduino IDE is an Integrated Development Environment for Arduino boards. |
| Spyder IDE | We use the Python programming language for many operations in our system. Spyder IDE is an open-source development platform for the Python programming language. |
| Local Area Network (LAN) | A Local Area Network is a computer network that can be used within a limited area. |
| JAVA | We will use Java to develop application which provides to control moving and display detecting objects. |
| YOLO | YOLO algorithm has good accuracy with faster run time. That's why we used YOLO to detect objects. |
| Android Studio | We developed an application on Android using Android Studio. This application shows the image of detected objects and provides a movement vehicle by using the arrow keys. |

## 1.4. Overview of This Document

In this software design document, we aimed to provide a comprehensive design description of Kaşif UGV. Therefore, we stated design statements of our system. In 1. Introduction section, we provide a summarized explanation regarding our system and shared the basic features of our system. We categorized 1. Introduction section as 1.1. Purpose of This Document, 1.2. Scope of the Project, 1.3. Glossary, and 1.5. Motivation subsections. In 1.1. Purpose of This Document section, we stated what this document aims to explain. In 1.2. Scope of the Project, we stated what must be achieved to create our system, and shared features of our system. In 1.3. Glossary, we provide a definition of all special terms that we used in this report. In 1.5. Motivation section, we shared why we implement such system, and reason behind our methodologies. Afterwards, in 2. System Design section, we shared comprehensive design architecture of our system. It includes subsections which are 2.1. Architectural Design, 2.2 Decomposition Description, 2.3. User Interface Design and 2.4. Hardware Design. In 2.1. Architectural Design, we stated different diagrams of our system and explain what is the context of the design idea behind the system. In 2.2. Decomposition Description explains general structure of our system. We provided a hierarchy between among the modules, and the structure. In 2.3. User Interface Design section, we stated designs for our mobile application of the project. These designs are given as figures with brief explanations. We stated which features can be controlled from the application and stated how the GUI will be designed. In 2.4. Hardware Design, we shared the design arhitecture between hardware components, and state how we actually assemble the system. In 3. Requirement Matrix section, we provided a table format which indicates the connection between components and requirements which are stated in SRS document. For that purpose, we generated a link between system components and requirements by explaining which requirement is satisfied by which component. We checked each requirement is attached to the necessary component. Therefore, this document will serve as a guideline for soldiers, technician, commander and development team for understanding the general design arhitecture of Kaşif UGV.

## 1.5. Motivation

Today, technology is advancing very quickly. One of the technologies of our age is autonomous vehicles that can make decisions on their own, supported by artificial intelligence. The usage areas of autonomous vehicles have started to increase day by day because they provide excellent convenience to our lives. Most importantly, when used in military areas, it reduces the loss of life to 0 and provides the security of the area it is located 24/7. As a group that closely follows technological developments, we determined that the future is in this field and decided to develop autonomous vehicles. We aware of the fact that we will be competent engineer candidates experienced in this field when we receive the final product, and we continue our work and move forward with the goal of doing the best. The contribution we will make to our country and ourselves is our greatest source of motivation.

# 2. System Design

This system design section includes the Architectural Design of the system, the definition of the problem, which technologies are used, User Interface Design, and Hardware Design. Also, it contains various diagrams such as Sequence Diagram, Activity Diagram, Data Flow Diagram, Class Diagram.

## 2.1. Architectural Design

In this section, we described the problem description, technologies used and different diagrams to provide better explanation of our architectural design. As we explained in previous sections, our system will include 3 main tasks. Hence, we will generate an architecture which contains these tasks. In Figure 1, Kaşif UGV’s main structure is given.

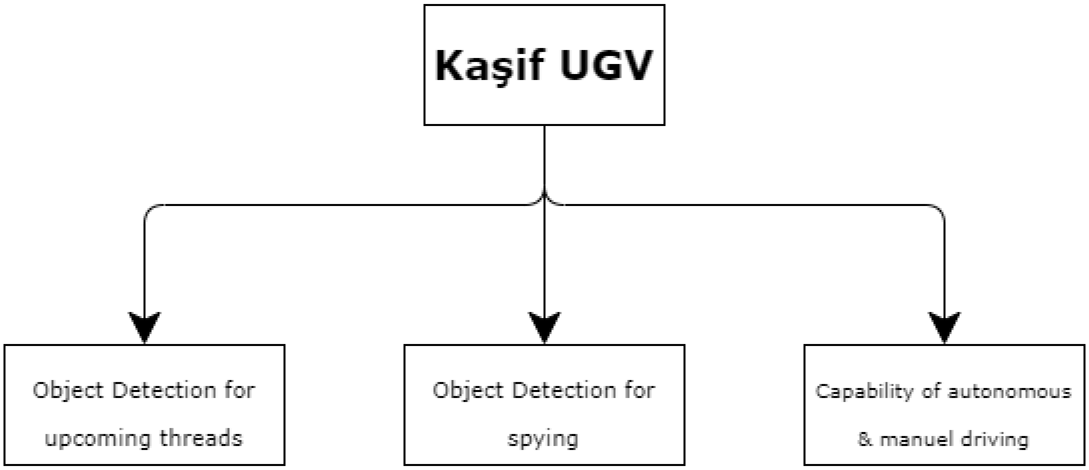

*Figure 1: System's general tasks*

### 2.1.1. Problem Description

Due to the fact that our country is in the middle east geography, it presents a dangerous situation. Our soldiers and security forces are responsible for protecting our country on a 24/7 basis, but when the human factor is involved, there is of course no personnel who can work fully efficiently on a long scale. Thanks to the autonomous vehicles that can make decisions with the high processor and artificial intelligence support developed, it is possible to provide security in a smooth and reliable way and to reduce the loss of life to 0. The project is being developed with the aim of both ensuring the security of our country and preventing possible loss of life.

### 2.1.2. Technologies Used

There are all the hardware products we use in our system below:

- Arduino Uno Board
- Robotic Car Cit (4 Wheels, 4 motors, chassis, screws, and AA battery enclosure)
- HC-SR04 Ultrasonic Sensor
- HC05/HC06 Bluetooth Module
- L298N Motor driver
- FTDI Programmer
- ESP32-CAM module
- Additional wires, breadboard, and connector cables
- Power Supply

There are all the software tools we use in our system below:

- YOLO

  We used the YOLO algorithm because it is faster and more advantageous than Faster R-CNN and other object detection algorithms.
- Arduino IDE version 8.1.12

  We use Arduino Uno board to be able to run our code on the vehicle.
- Python

  We use Python programming language for many operations in our system.
- Spyder IDE

  Spyder IDE is an opensource development platform for Python programming language.
- C/C++

  We used C/C++ on Arduino to enable the vehicle to do manual driving, autonomous driving, guard mode detection and spy mode detection.

The application which will be responsible for operating vehicle will be developed by a programming language which can combine above solutions. It can be Java, HTML/CSS, or Python. In future process of development, it will be decided.

## 2.1.3. Data Flow Diagram

In figure 2, data flow diagram is stated.

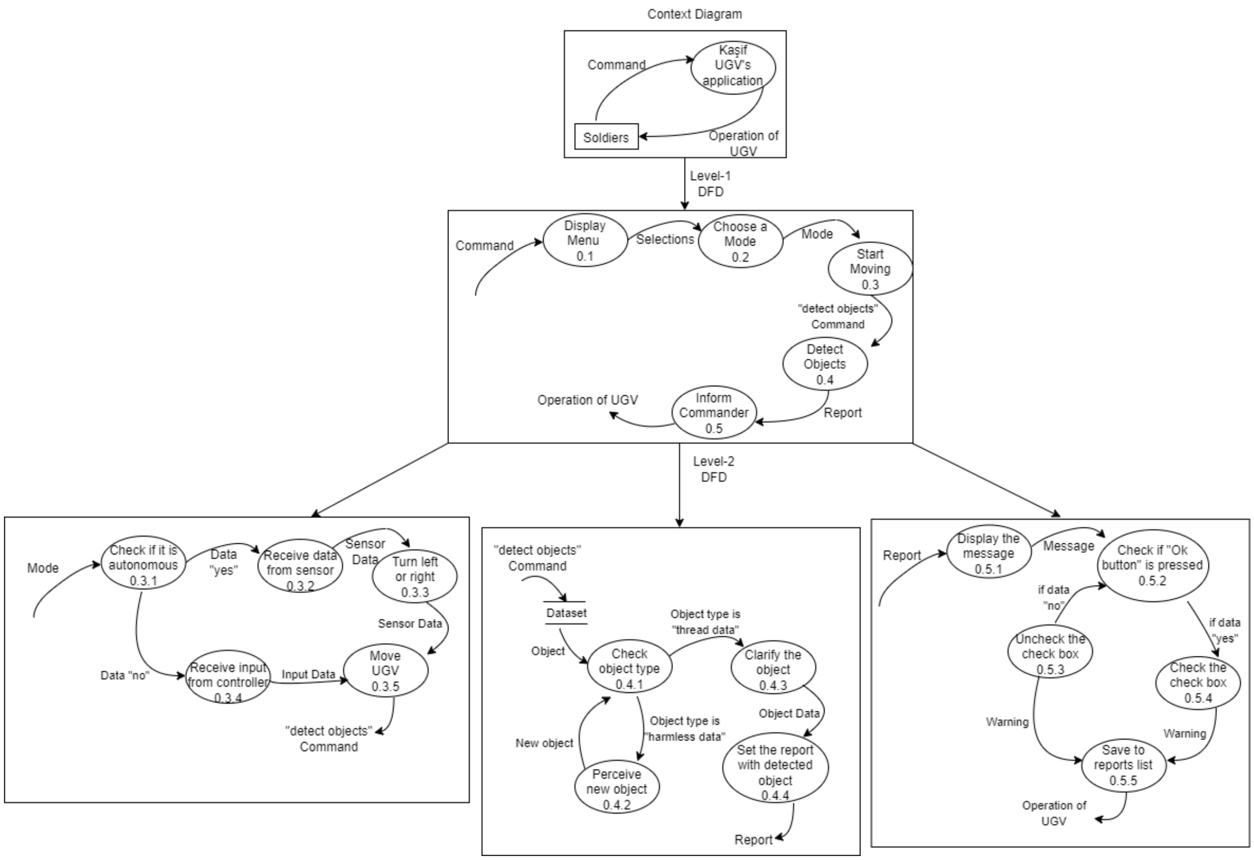


*Figure 2: Data Flow Diagram*

### 2.1.4. Activity Diagram

In figure 3, activity diagram is stated.

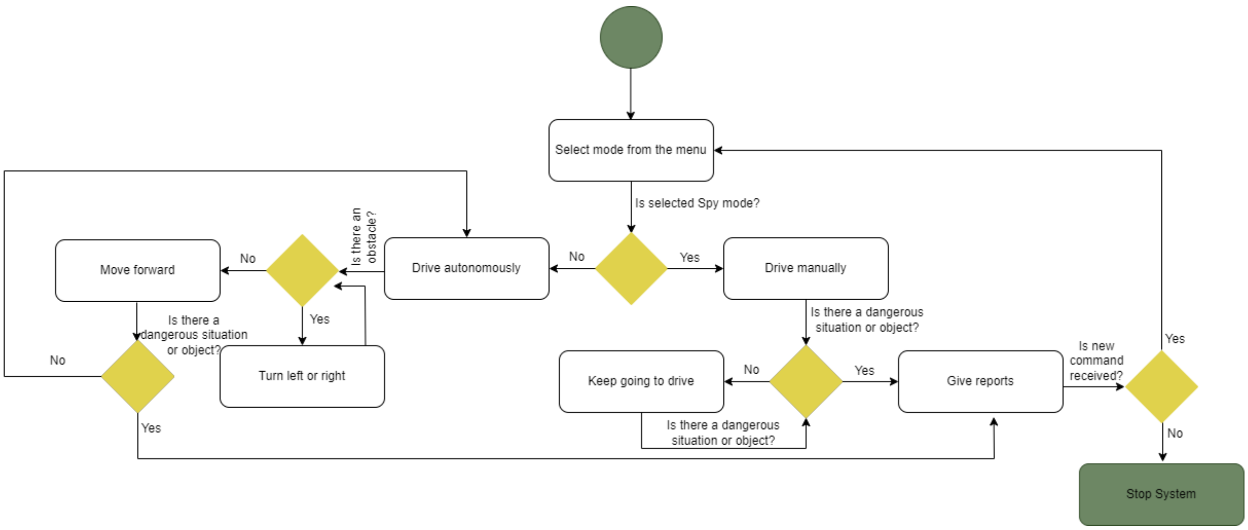


*Figure 3: Activity Diagram*

### 2.1.5. Class Diagram

In figure 4, class diagram is stated.

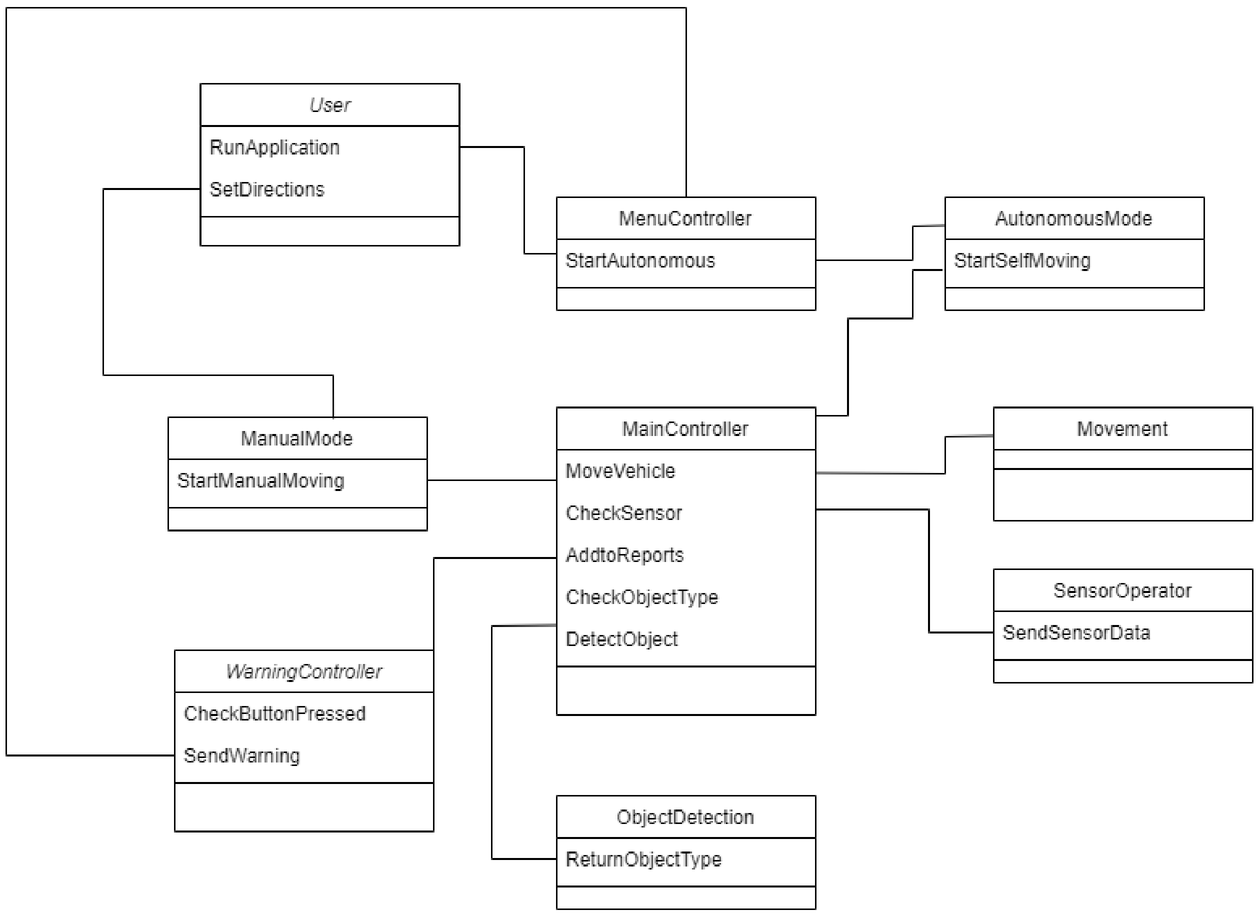


*Figure 4: Class Diagram*

### 2.1.6. Sequence Diagram

In figure 5, sequence diagram is stated.

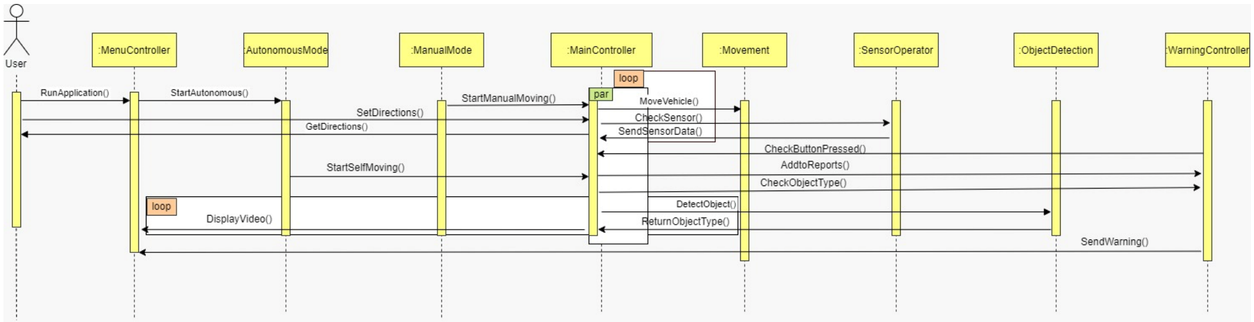


*Figure 5: Sequence Diagram*

## 2.2. User Interface Design

In this section, we will state our designs for possible use cases and explain what features are included for each UI screen.

### 2.2.1. Home Page


*Figure 6: Home Page Screen*

To be able to operate any feature of our project, user must need to choose which mode will Kaşif operate. Therefore, in home page, we prompt user to choose a mode which can be guardian&autonomous or spy&manual. When one of these options is selected, screen will change to the one the following screens.

### 2.2.2. Spy Mode Screen


*Figure 7: Spy Mode Screen*

This interface is the main interface of our mobile application. It includes UC-01 Control Vehicle and UC-11 Detect Objects use cases. To be able access this page, “Spy Mode” should be selected in home page. Commanders and soldiers can control the vehicle from here and observe the stream that is coming from vehicle’s camera. In the screen there are 7 buttons and two menu bars in total. The left-side menu bar is responsible for changing screens to display other features of the application. From right-side menu bar user can exit from the application, and reach general information related the application. By pressing “LED” button, user can open the led light of ESP32 camera which provides light. This feature can be used for night operations or surveillances. By pressing “STREAM” button, user will observe the real time video which is coming from the ESP32 camera. Therefore, user can watch the environment of Kaşif UGV. When user presses “DETECT OBJECT” button, our vehicle starts to detect objects around it. User will be able to observe that is detected at that moment. Sample visual is given in above figure.

### 2.2.3. Thread Captured in Spy Mode

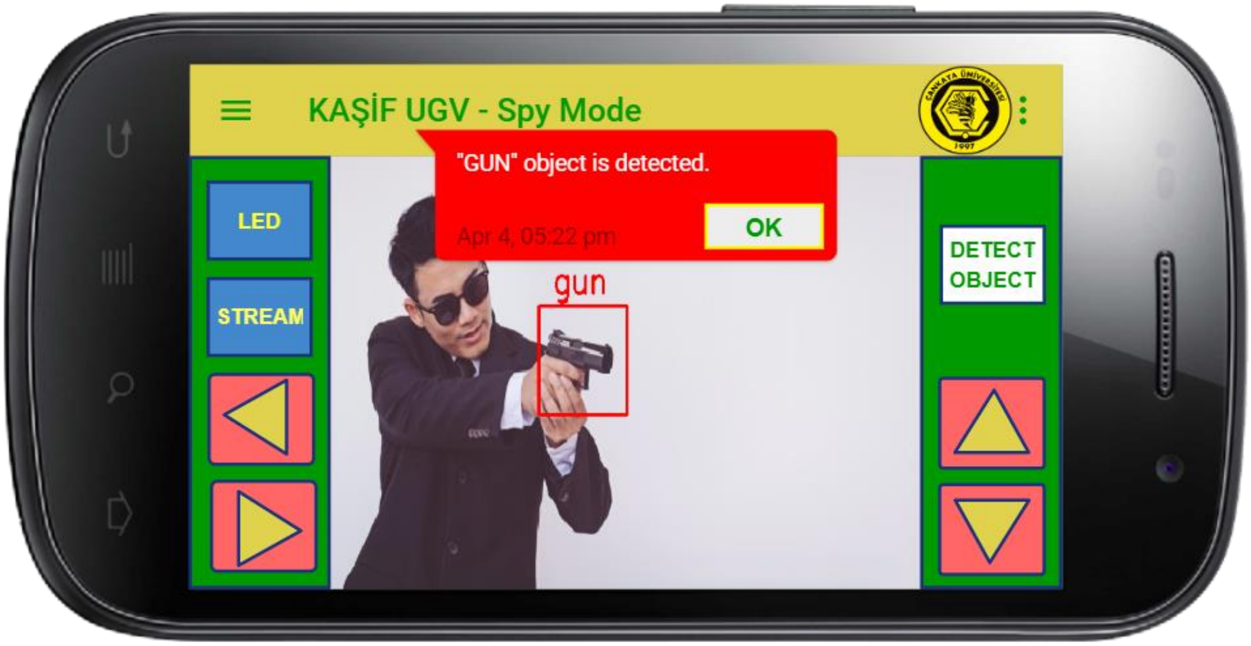


*Figure 8: Inform Commander UI*

This interface mainly includes UC-02 Inform Commander, UC-10 Drive manually use cases. To be able access this page, "Spy Mode" should be selected in home page. When the camera detects suspicious objects or threads, commander must receive a warning. In above figure, a warning can be observed. It indicates which and when the suspicious object is detected. There is also a "OK" button which will be used to check if commander received the message. If "OK" button is not clicked, in the Reports page, it will be seen as unchecked. Therefore, to get commanders' and soldiers' attention, warning box is colored as red.

### 2.2.4. Guardian Mode Screen


*Figure 9: Guardian Mode Screen*

This interface includes UC-09 Perform Self-Driving use case. To be able to enter this screen, Guardian&Autonomous mode must be selected from the home page. Therefore, once above screen open, our vehicle will start to move itself. As it can be seen from above figure, there are no direction arrows anymore. There are only “STREAM”, “DETECT OBJECT” and “LED” buttons to control ESP32 camera. In here, there will be a warning pop-up as well. The reports will be written into Reports page as checked or unchecked. In short, by using this page, without human invervention, we can observe objects in facility’s environment and receive a warning when there a thread is detected.

### 2.2.5. Display Reports

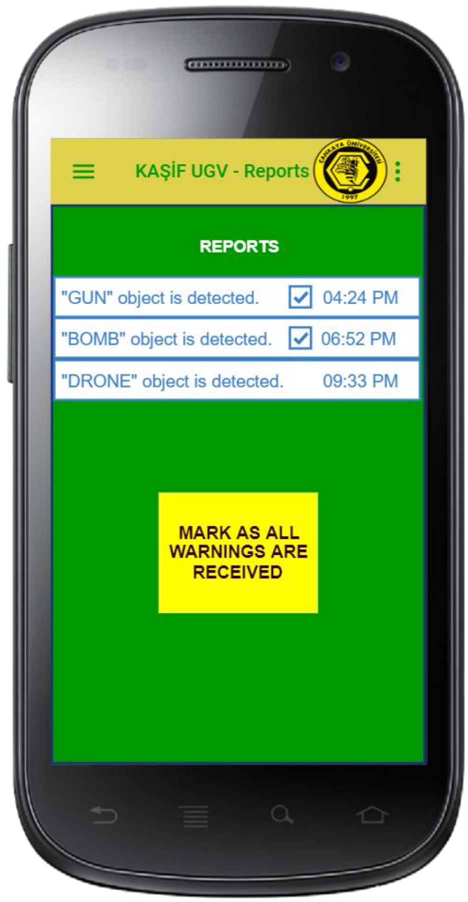


*Figure 10: Reports Screen*

This interface is used to display reports received from our vehicle. It includes UC-012 Give Report use case. To be able access this page, “Reports” page should be selected from menu page. In this page, warning messages will be listed with time information. From above figure, it can be observed that there is a checkbox at the near of each message. It refers to the “OK” button that we explained in *2.3.3. Thread Captured in Spy Mode* section. If commander or soldier presses OK button when the warning is received, then there will be a checkbox at the near of the message. This feature can be used to understand if any thread is detected when the commander or soldier was not available to see the warning. They can check the list and observe all warnings. After that, by pressing “MARK AS ALL WARNINGS ARE RECEIVED” button, they can check every warning to refer all of them are received.

## 2.3. Hardware Design

Since in our project, we will also generate the physical product of the vehicle, we need variety of hardware components. That is why, we needed regular and comprehensive design to make all components work properly. In this section, we will state our hardware design by categorizing it into five subsections which are ESP32 Camera – FTDI Module Design, Motors – L298N Motor Driver - Arduino Uno Board Design, HC05 Bluetooth – Arduino Uno Board Design, HC-SR04 Sensor – Arduino Design, and Power Supply Design. All the components which we explain the design architecture below are explained in SRS document in *3.1.2. Hardware Interfaces*.

### 2.3.1. ESP32 Camera – FTDI Module Design

We stated that we used ESP32 Camera to get real-time data via Wi-Fi with FTDI programmer module. In ESP32 Camera module we have a development board, and OV2640 camera. There is no USB programming board on ESP32 development module, that is why, we used external programmer FTDI. We assembled these two components as it can be seen from below figure[1].

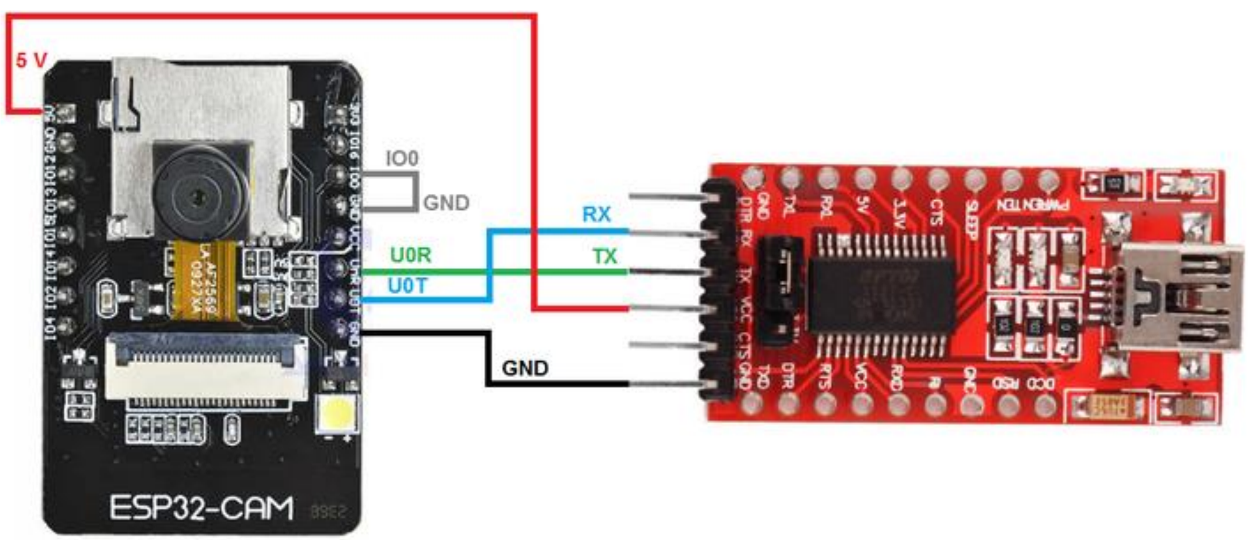


*Figure 11: Connections between ESP32 Camera and FTDI module*

As it can be seen from above figure, we supply board with exactly 5V power to use it with maximum performance. All the wire connections are stated in Figure 11. Additionally, there is one crucial step which we must perform during uploading process of our code. After upload is done and serial port is opened, we must press RESET button on ESP32 development board. After that, we opened serial monitor and disconnect GPIO0 from ground since code is already uploaded. After implementing these steps, our camera is ready to be find from external devices via Bluetooth and recording videos.

### 2.3.2. Motors – L298N Motor Driver - Arduino Uno Board Design

For each wheel, we used a DC motor. Therefore, there are 4 DC motors in our circuit. In below figure, only half of the connections to provide better understanding of connections.[2] On L298N motor driver, we plugged pins to set wheels' speed and direction of motors. By using the direction control pins, we can control motor spins either forward or backward.

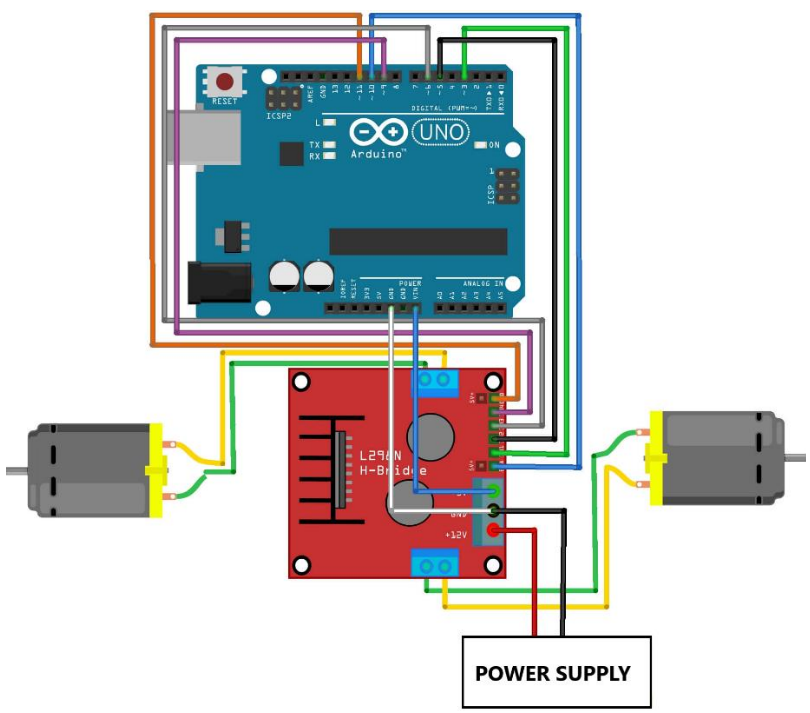


*Figure 12: Connections between Motor, motor driver and Arduino*

### 2.3.3. HC05 Bluetooth – Arduino Uno Board Design

To able to operate wheels from remote control, we used HC05 Bluetooth module which ıs connected to our Android smartphone. Using this module, we can control our vehicle manually via our smartphone using our application. HC05 will be blinking when it is in configuration mode. To connect HC05 with Android smartphone, Bluetooth settings must be switched to ON. Once the connection is satisfied, it stops blinking. The circuit that we performed is given at below figure[3].

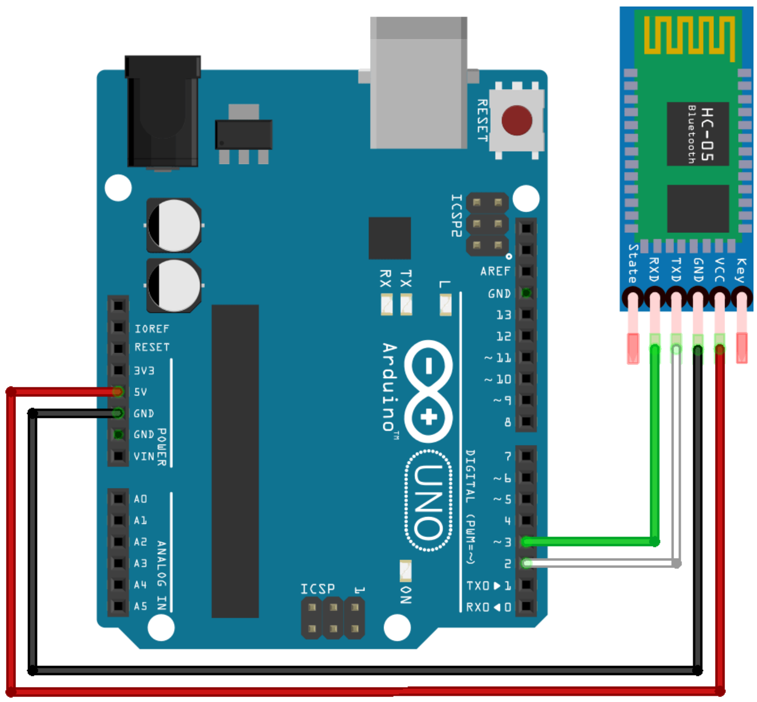


*Figure 13: Connections between Arduino and HC05*

### 2.3.4. HC-SR04 Sensor – Arduino Uno Board Design

Since our vehicle must be operated in autonomous mode as well, we used HC-SR04 sensor which measures distance. It emits an ultrasound at 40kHz to detect obstacles around it. Therefore, it calculates distance based on speed of sound[4]. To make HC-SR04 sensor work properly, we assemble the circuit as follows:

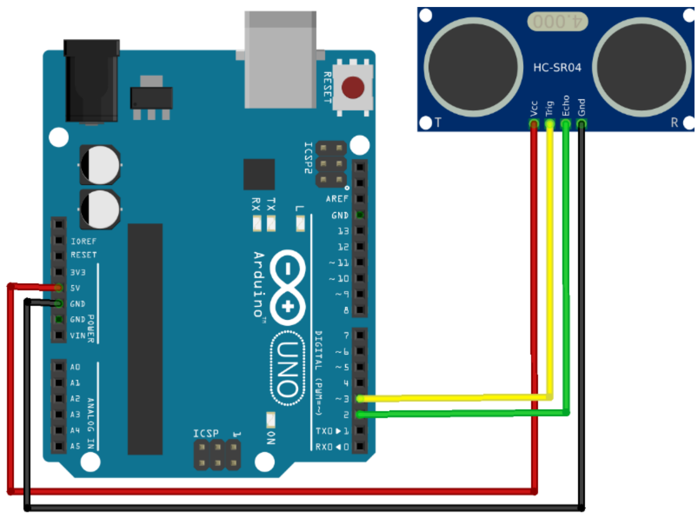


*Figure 14: Connections between Arduino and HC-SR04*

In below figure, how can distance be calculated using HC-SR04 is explained.

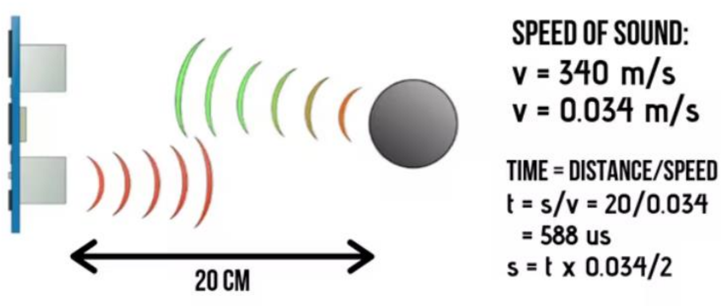


*Figure 15: Distance Calculation using HC-SR04*

Since sound wave needs to travel forward and bounce backward, we divided speed by two when calculating the time.

### 2.3.5. Power Supply Design

In this section, power requirements of crucial hardware components are given.

**HC-SR04 Ultrasonic Sensor:**

- **Voltage:** 5V
- **Current:** 15mA

**L298N Motor Driver:**

- **Voltage:** 5-35V
- **Current:** 0-36mA

**ESP32 Camera Module:**

- **Voltage:** 5V
- **Current:** 2A

**DC Motor:**

- **Voltage:** 3-12V
- **Current:** 95mA

**HC-05 Bluetooth Module:**

- **Voltage:** 1.8-3.6V
- **Current:** 50mA

# 3. Requirements Matrix

In Figure 16, requirement matrix of our system is stated.

| | Components | COMP-01 | COMP-02 | COMP-03 |
|---|---|---|---|---|
| Requirements | | | | |
| FREQ - 1 | | | | X |
| FREQ - 2 | | X | X | |
| FREQ - 3 | | X | X | |
| FREQ - 4 | | X | X | X |
| FREQ - 5 | | | | X |
| FREQ - 6 | | X | X | |
| FREQ - 7 | | | | X |
| FREQ - 8 | | | | X |
| FREQ - 9 | | | | X |
| FREQ - 10 | | | | X |
| FREQ - 11 | | X | X | |
| FREQ - 12 | | | | X |

COMP - 01 = Object Detection for upcoming threads
COMP - 02 = Object Detection for Spying
COMP - 03 = Capability of Autonomous & manuel driving

FREQ - 1 = Control Vehicle
FREQ - 2 = Inform Commander
FREQ - 3 = Attack when it is necessary
FREQ - 4 = Defend Facility
FREQ - 5 = Provide Service
FREQ - 6 = Observe Enemy & Enemy Battle
FREQ - 7 = Control Units
FREQ - 8 = Attempt Attack
FREQ - 9 = Perform Self-Driving
FREQ - 10 = Drive Manually
FREQ - 11 = Detect Objects
FREQ - 12 = Give Reports

*Figure 16: Requirement Matrix*

# References


[1] Assembly of ESP32 Camera, “Connections between FTDI and ESP32 Development Module” [Online]. Available:
https://www.instructables.com/ESP32-Cam-Programmer/
[Accessed 25 December 2021].

[2] How to use L298N Motor Driver, “Connections between motors and L298N” [Online]. Available:
https://create.arduino.cc/projecthub/ryanchan/how-to-use-the-l298n-motor-driver-b124c5
[Accessed 25 December 2021].

[3] How to use HC05 with Arduino “Circuit Diagram of HC05 and Arduino” [Online]. Available:
https://www.aranacorp.com/en/arduino-and-bluetooth-module-hc-05/
[Accessed 26 December 2021].

[4] How to use HC-SR04 with Arduino “Circuit Diagram of HC-SR04 and Arduino” [Online]. Available:
https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6
[Accessed 26 December 2021].