MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Potato Integrated Technologies

Conceptual Design Report

Project: Devices Trying to Score in Each Other's Goals

Duration: 01/10/2018 – 01/06/2019

Estimated Cost: $200

Design Studio Coordinator: Arzu KOÇ (Section 5)

COO	CMO	CTO	CEO	CFO
İrem COŞKUN	Berkay GÖKSU	Furkan Bahadır ELİK	Fatma Nur ARABACI	Aycan BEYENİR
2030419	2030757	2030518	2030047	2030328
+90 539 414 5969	+90 505 635 8145	+90 554 963 9410	+90 533 602 7527	+90 539 330 0570

Table of Contents

1 Executive Summary

This report presents details of the project for a teleoperated robot trying to score in opponent's goal and defend its own meanwhile. Remotely controlled robots are widely used in many aspects of our life. Main idea behind this project reflects a real-world problem in occasions where the user has no view of the robot. Therefore, a two-way communication with the robot and user's computer should be established. Reliability and efficiency of this communication is vital since intervening a distant robot which can be unreachable at that instant, by the user can be challenging.

This design will be accomplished by building an efficient communication, a robust mechanical design and reliable electronic system. We aim to achieve satisfactory performance in speed and accuracy. Our final product is to detect the ball on the field, shoot the ball towards the opponent's goal, defend its goal and stay in its half field doing so. Several solution methods are sufficiently described in the following sections of this report.

Our company is composed of five shareholders with different specialization fields and backgrounds. Therefore, each team member came up with a solution from a different point of view. We merged these perspectives in order to construct solution approaches at this point of our design process.

Even though our base knowledge is similar, each team member chose to study in different fields. Therefore, at certain levels of the design process and solution offering, each member has a different idea and experience. Ms. Arabacı is more experienced in controller design and system modelling which will help her to guide and inform the team, Ms. Coşkun and Mr. Göksu will lead the team with their mechanic & analog design and integration skills. Mr. Beyenir's skills & background in image processing will help us in programming stage and last but not least Mr. Elik will take an active role in building a durable and stable communication system.

Final product will be delivered in best way with a cost of 200$, at the end of 7 months by PITECH engineers. Once the customer purchases the final product, they will own the final action robot consisting of mechanical subsystems, camera, sensors and drivers. In addition to these, a user manual, a warranty document, required software tools, four batteries, two battery chargers, three game field walls, two balls (one is extra) and a dummy robot are also provided.

2 Introduction

In the last few decades, robots are gaining more complex abilities, thanks to improvements of technology, that they substitute for humans in many fields of industry. This progress enables us to handle things easier and, in more time, efficient way since their performance is better than ours in many aspects of our daily life such as personal, professional life etc.

Being a newly founded company with five highly motivated, young engineers from different specialization fields such as electronics, control, computer and telecommunications; our aim is to develop a teleoperated robot that can play hockey which includes trying to score in opponent’s goal and also defend its own goal. Apart from the specifications defined above we intend to come up with the best featured robot possible.

In this project, our main purpose is to build a robot that we can control from a specified distance with a remote controller. In order to fulfill this requirement, we found an efficient way to transfer data from our robot to the main computer so that we can improve our chance to score a goal and win the round. This is a two-way communication since we will send directions to the robot so that it can move with respect to these commands. In addition, the mechanical structure of the robot should also be robust so that it can endure possible encounters with the ball. While developing the design, it should be noted that both mechanical and electrical solutions support each other. To sum up, this project and its solutions may contribute to the areas where teleoperated robots are used for many different purposes.

In this report, a detailed analysis of this project is presented. Detailed problem statements are defined, solution approaches for each subsystem is discussed, team organization, tentative cost-budget analysis and time plan of the project is introduced.

3 Overall Description

3.1 Description of Individual Subsystems

- **Power Supply Subsystem:** This subsystem is for distribute power towards other subsystems at required voltage level and with specified current limitations.
- **Communication and Telecontroller Subsystem:** This subsystem provides communication between the robot and controller subsystem. This system is used for sending commands and sending the data received by robot and detection subsystem.
- **Motion Subsystem:** This subsystem consists of wheels and motors driving these wheels by which we provide capability of motion to our robot in the game field.
- **Detection Subsystem:** This subsystem detects the boundaries of the field, the position of the opponent robot & the ball. This subsystem also locates itself in the field.
- **Main Processor Subsystem:** This subsystem is the main robot computer onboard which controls other subsystem with respect to feedbacks sent from these subsystems.
- **Shooting Subsystem:** This subsystem is responsible for sending the ball to the opponent's goal with high precision to score a goal.

3.2 Block Diagram

Block diagram of the robot side is as in Figure 3.2.1.

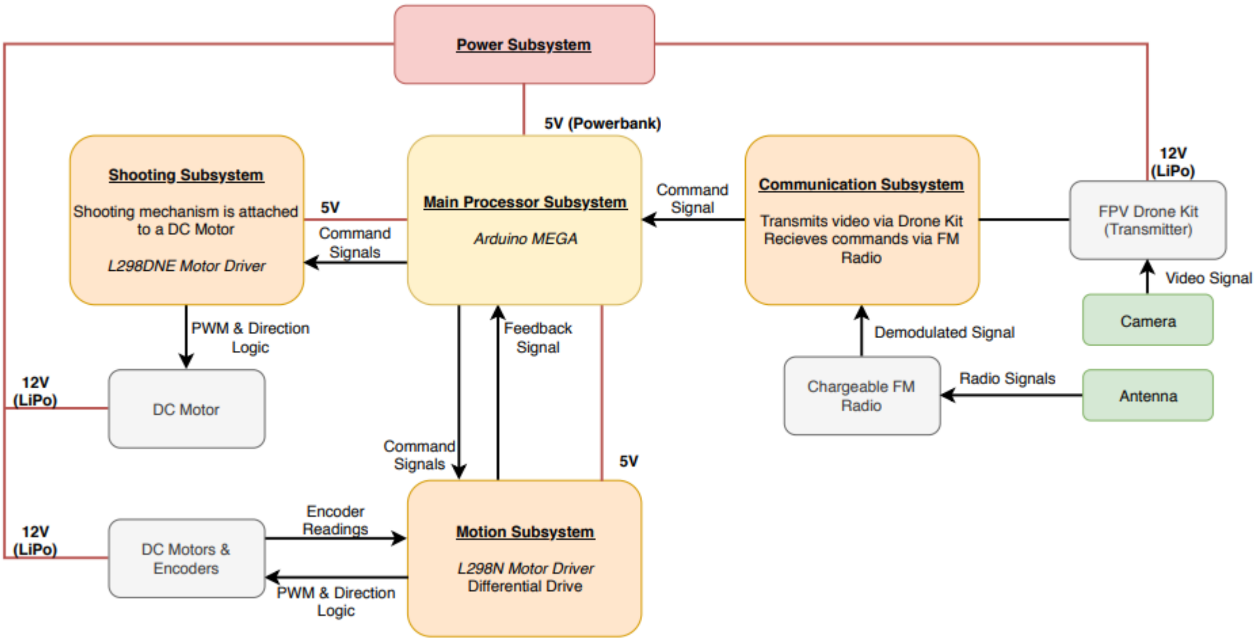

Figure 3.2.1: Block Diagram of Robot Side

Block diagram of the telecontroller side is as in Figure 3.2.2.

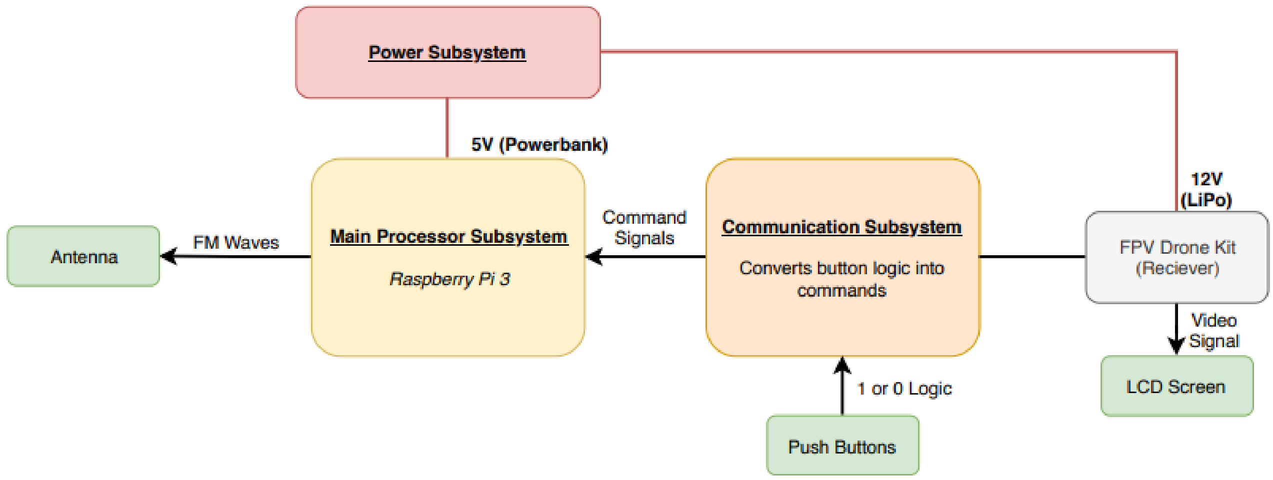

Figure 3.2.2: Block Diagram of Telecontroller Side

3.3 System Level Flowchart

Flowchart for Drone Kit Solution is as in Figure 3.3.1.

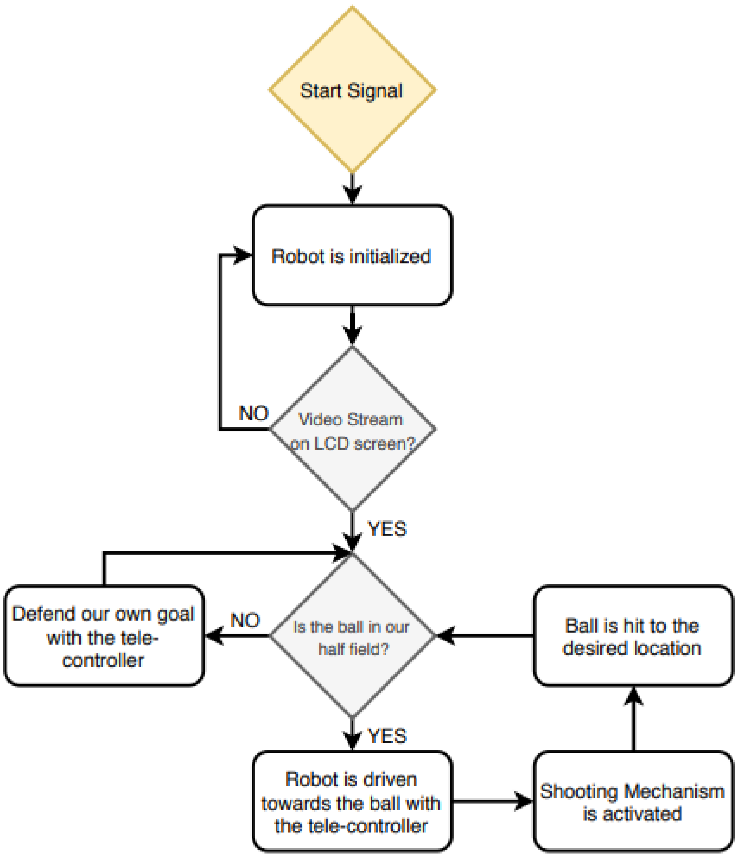

Figure 3.3.1: Flow Chart for Drone Kit Option

Flowchart for Image Processing Solution is as in Figure 3.3.2.

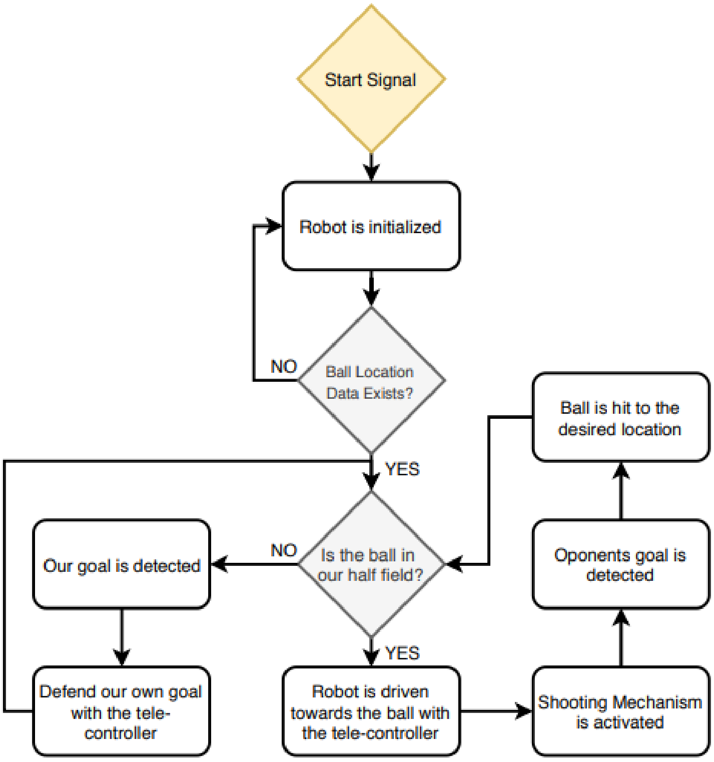

Figure 3.3.2: Flow Chart for Image Processing Option

4 Restatement of the Problem and Requirement Analysis

The goal of this project is to design and construct a teleoperated robot (controlled from a distance up to at least 30 meters) which can compete with a similar robot in shooting and scoring to opponent's goal. This part is restatement of the problem and requirement analysis. We already state the problem for proposal report. This is the updated version of the previous statement and requirement analysis. Functional, physical and performance requirements of the project are as follows:

4.1 Functional requirements:

- Detect the start signal
- Monitor the surrounding
- Process the monitored data
- Encode the processed data for communication
- Transfer the encoded data to the teleoperator

If the ball is at players half-field and far away from the robot:

- Transfer the movement direction command given by the teleoperator, to move toward the ball
- Perform the move operation respect to the command transferred from teleoperator
- Move robot to the ball until ball is in the shooting range
- Transfer the hit the ball command given by the teleoperator
- Perform the hit the ball operation given by teleoperator

If the ball is at opponent's half-field:

- Transfer the movement direction command given by the teleoperator, to cover the goal, given by the teleoperator
- Perform the move operation respect to the command transferred from teleoperator
- Move robot to the own goal to protect it from the incoming shoot
- Protect the goal respect to the commands from teleoperator

4.2 Physical requirements:

- Goals must be at least twice as wide as their defenders' lateral dimensions.
- Robots can hit, push or otherwise drive the ball but not grasp, scoop or otherwise carry it. So, robot cannot have a grasping or scooping part.
- The playfield also has some physical requirements:
    - The playfield should be regular hexagon on a bare floor, with center-line and goal lines marked by "masking tape".
    - The playfield constructed from 6 sidewalls of 70-75 cm length each and two goals snugly fit at the opposite corners, while preserving symmetry.
- The robot should fit in a cylinder to measure the maximum dimension.
- In order to move fast enough our motors should carry the weight of the remaining parts.
- Weight should be aligned in center for steady and controllable movement.

4.3 Performance requirements:

- Ball should be transferred to opponent's half-field in no more than 20 seconds from the project description. For our team for successful operation it should be less than 15 in order not to have any unforeseen violations. So, consistency of operation in this case is important. It should transfer the ball to the opponent field less than 15 seconds at least 55% of the time and other times it should be less than 20 seconds because of project description.
- The operator remotely controls the robot from a distance up to at least 30 meters.
- The transmission between the robot and receiver should be less than 1 second for successful operation. So, less than 4 points for transmission delay objective is unacceptable.
- In order to have successful operation aiming precision of the robot should not be less than 45%. So, less than 4 points for precision objective is unacceptable for our company.
- The robot's throwing mechanism should be strong enough to throw the ball to opposing goal.
- The robot's structure should not be affected by hitting the ball fast to the robot, so, durability of the robot cannot take less than 4 points.

Constraints:

- The robot is not allowed to cross the center-line
- The ball must be transferred to the opponents half field no more than 20 seconds
- The communication between robot and teleoperator must happen from a distance at least 30 meters
- Carrying, grasping and scooping the ball is not allowed. Robot can only hit or push the ball.

Objectives, metrics and related objective trees are given in the next section of this report.

5 Solution for Subsystems

5.1 Power Supply Subsystem

The one thing to remember about battery selection is that there is no such thing as a perfect battery that works for every application. Selecting the right battery for our application is about identifying the most important battery metrics and trading these off against others.

For this selection, we use the comparisons in Figure 5.1.

A Quick Guide for Common Battery Chemistries

	PROS	CONS	COMMON APPLICATION
LEAD ACID	Rechargable, Extremely Common, High Power Density, Durable, Wide Temperature Range	Low Energy Density, Very Heavy, Large in Size	High Current Demand Applications, Car Batteries, Large Scale Battery Banks
ALKALINE	Extremely Common, Cheap, Decent Power Density, Low cost	Susceptible to Natural Rupture, Generally Non-Rechargable, Short Lifetime	Non-Rechargable Consumer Electronics, Flashlights, Toys, Household Items
NICKEL-METAL HYDRIDE	High Current Ability, Less Susceptible to Memory Issues, Lower Cost	Short Storage Life, Susceptible to Overcharge	Power Tools, RC Airplanes and Drones, Portable Systems
LITHIUM ION	Very High Energy Density, Limited Memory Effect, Long Life, Low Maintenance, Rechargable	High cost, Vunerable to Stress (and Exploding!), Require Lots of Protection	Space Constrained Products, Weight Constrained Products, Cell Phones, IoT Devices, Electronic Watches

Figure 5.1 Quick Guide for Common Battery Chemistries [1]

Lithium is the lightest metal in the periodic table and has a specific capacity of 3860 mAh/g. Lithium also has an electrochemical reduction potential of 3.045 V against 1.22 V for NiMH (i.e. a lithium-based battery provides a battery voltage of 3 V or greater). Some of them can be seen in Figure 5.2. The combination of these two properties results in very high energy densities for lithium-based batteries. [4]

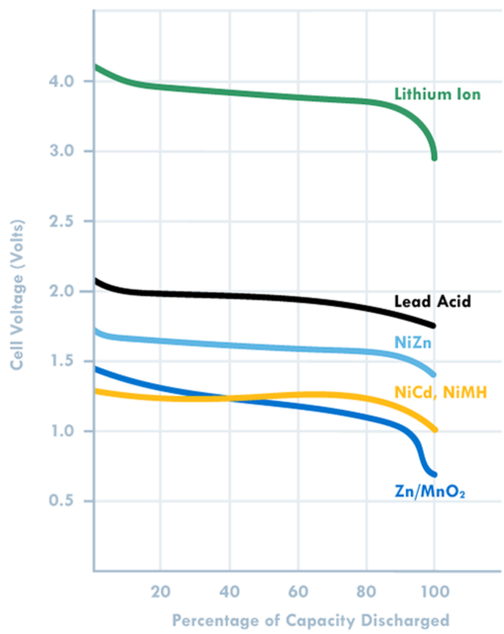

Figure 5.2 Graph of one cell voltage capacity [2]

Li-Po batteries are lightweight, can be produced in desired size, have high capacity and power, also have fast charging and discharging features provides us to choose the Li-po battery.

We decided to use Li-Po to drive motor, but we decided to use powerbank for controllers (arduino, raspberry). Because they have to drive with 5V. For this reason, we can also use a voltage regulator with a Li-Po battery for controllers as a backup plan.

After deciding to use Li-Po, the next step was to choose which type of Li-Po should be used. Since we need 12V, we decided to use "3S" (3 * 3.7V = 11.1V. In Figure 5.3, the one Li-Po cell can be seen.

Figure 5.3 :Example of 3 cells Li-Po Pack [3]

The power sources that we should use in the project can be seen below:

Robot Part:	Drone Transmitter	LiPo – 12V
	2 DC motors for movement	LiPo – 12V
	1 DC motor for shooting	LiPo – 12V
	Arduino Mega	Powerbank – 5V
	Radio	Li-ion (inside itself)
Controller Part:	Drone Receiver	LiPo – 12V
	Raspberry Pi 3	Powerbank – 5V

5.1.1 Level Risks Assessment

There are few risks of using the above-mentioned processors.

- The first risk is these systems heats a lot which can create a risk for other parts of the system.
- Power supplies are one of the heaviest part of our whole system, so they can create alignment problems and cause immobilize.

5.1.2 Error Sources

Some internal and external sources can cause an error at this subsystem. The possible error sources are as follows:

- Environmental temperature conditions can be a source of error. To eliminate this effect fan can be used or power is drawing can be reduced.

5.2 Main Processor Subsystem

Processor subsystem consists all the processors in different subsystem.

5.2.1 Solution and Relevant Algorithms:

A processor is needed to identify the command and give an output respectively. We have already tried different processors for this task. At first, we tried TIVA Board for identifying the frequency. It worked but with TIVA board we needed another module of PWM converter. So, we search for different and easier ways to identify a signal. For the demonstration we used

Arduino MEGA. For a cheaper option we also tried Arduino UNO but for demonstration it failed to give enough current for our system.

5.2.1.1 Plan A

Our main plan for main processor subsystem is to use Arduino Mega. Figure 5.4-5.5 below shows an Arduino Mega and its specifications.

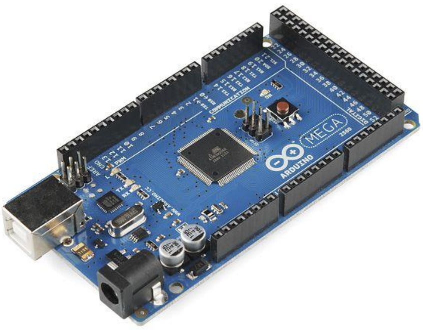

Figure 5.4 Arduino Mega [5]

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Figure 5.5 : Arduino Mega specifications [5]

Our main plan is to use Arduino Mega for main processor subsystem it is chosen because it has better DC current than UNO. Also, Arduino has a variety of online sources that are easily reachable which makes it easier to use from alternatives like TIVA. The Figure 5.6 below shows the code for code for receiving sine input and identify the frequency, then output from different GPIO pins.

```
//#include <LiquidCrystal.h>

//LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

int Htime;                //integer for storing high time
int Ltime;                  //integer for storing low time
float Ttime;              // integer for storing total time of a cycle
float Ttime1;
float Ttime2;
float Ttime3;
float frequency;          //storing frequency
void setup()
{
    pinMode(41,INPUT);
    pinMode(13,OUTPUT);
    pinMode(53,OUTPUT);
    Serial.begin(9600);
    //lcd.begin(16, 2);
}
void loop()
{
//    lcd.clear();
//    lcd.setCursor(0,0);
//    lcd.print("Frequency of signal");

    Htime=pulseIn(41,HIGH);      //read high time
    Ltime=pulseIn(41,LOW);        //read low time
    Ttime1 = Htime+Ltime;
    Htime=pulseIn(41,HIGH);      //read high time
    Ltime=pulseIn(41,LOW);        //read low time
    Ttime2 = Htime+Ltime;
    Htime=pulseIn(41,HIGH);      //read high time
    Ltime=pulseIn(41,LOW);        //read low time
    Ttime3 = Htime+Ltime;
    Ttime = (Ttime1 + Ttime2 + Ttime3)/3;

    frequency=1000000/Ttime;    //getting frequency with Ttime is in Micro seconds
//    lcd.setCursor(0,1);
//    lcd.print(frequency);
//    lcd.print(" Hz");
    if( frequency < 4100 && frequency > 3900)
    { digitalWrite(13,HIGH);
    }
    else{
      digitalWrite(13,LOW);
    }
    if( frequency < 2100 && frequency > 1900)
    { digitalWrite(53,HIGH);
    }
    else{
      digitalWrite(53,LOW);
    }
    Serial.println(frequency);
    delay(500);
}
```

Figure 5.6: Code for GPIO control of Arduino

From the Figure 5.6 above it can be seen that it is easy to program Arduino. This code both reads the input and give output from different ports. Output of this system will be connected to the motors of motion subsystem.

5.2.1.2 Plan B

Our plan B for this subsystem is to use another processor for example TIVA board, Arduino Uno or Raspberry Pi can be used for this purpose. All of these have different advantages and disadvantages. These are going to be covered in 5.2.5 Comparative Analysis section. TIVA board has already been tried for frequency control and outputting. Arduino UNO works with the same principle as Arduino Mega there is no need to change the code above in Figure 5.6. In addition to that Raspberry Pi also has GPIO ports that can be used for this purpose, but it is not as efficient as Arduino Mega.

5.2.2 Level Risks Assessment

There are few risks of using the above-mentioned processors.

- The first risk is these systems heats a lot which can create a risk for other parts of the system.
- Processors are one of the heaviest part of our whole system, so they can create alignment problems.
- Processors have their built-in fault algorithms which causes them to shut down at unforeseen situations.

5.2.3 Error Sources

Some internal and external sources can cause an error at this subsystem. The possible error sources are as follows:

- Environmental temperature conditions can be a source of error. To eliminate this effect heat sink or fan can be used.
- Overflow can be an error caused by increasing the current coming to the system. To eliminate this a voltage comparator can be used so that we should be sure that we are not feeding the systems a voltage higher than 3.3V.
- For higher frequencies sampling rate can cause discrepancies. To eliminate this kind of error we are going to work on comparatively lower frequencies.

5.2.4 Test Results

We conducted a test to find error rate of the Arduino Mega frequency counter. We experimented the frequencies between 0-20kHz. The results in Figure 5.7 below showed approximately 2% error for these frequencies which is acceptable in many cases.

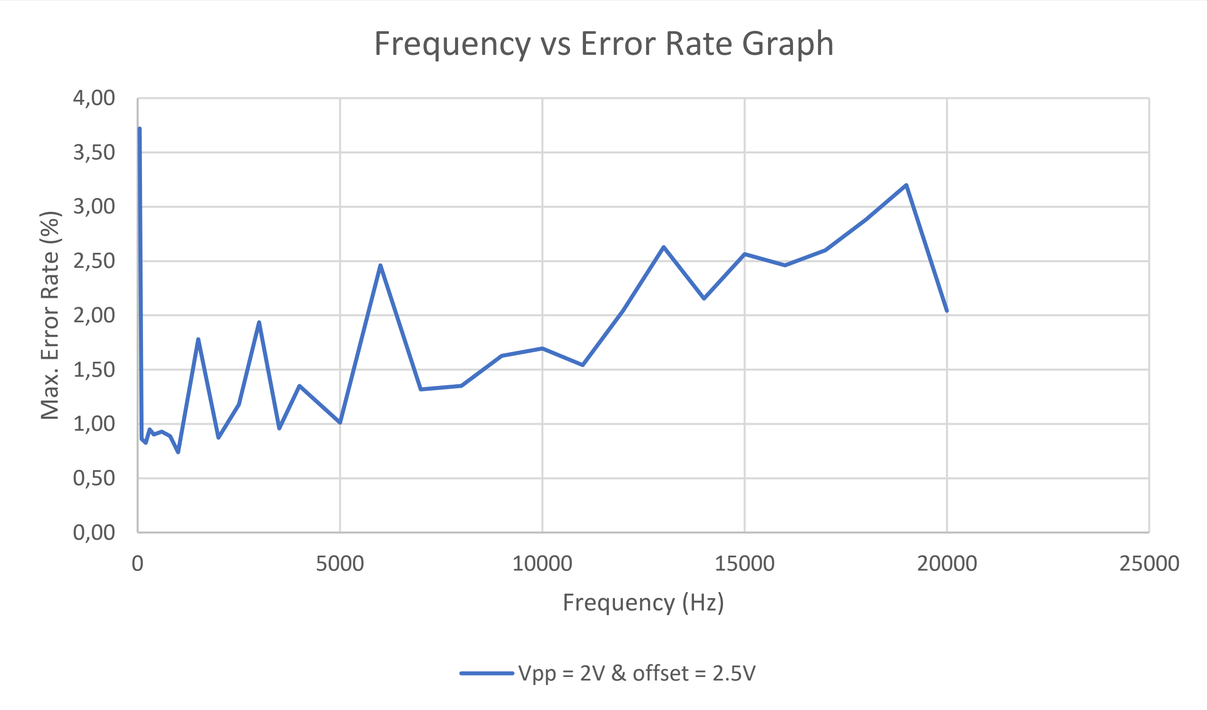

Figure 5.7: Frequency vs Error Rate for Arduino Mega frequency counter

5.2.5 Comparative Analysis

In order to choose our main solution, we compared possible solutions. Comparative analysis can be seen in Table 2 below.

Table 5.1: Comparative analysis for main processor

	Arduino Mega	*TIVA Board*	*Arduino Uno*	*Raspberry Pi*
Price ($)	38.5	14	20	5-35
Analog compatibility	Yes	No	Yes	No
Max DC current(mA)	50	40	40	40
Weight(g)	37	20	25	23

Even though, Arduino Mega has the worst values for price and weight it can has higher maximum DC current rate and analog compatibility. Analog compatibility and DC current are the most important specifications for selection, so we selected Arduino Mega even though it is the heaviest and priciest option.

5.3 Shooting Subsystem

There are two issues that stand out in this part of the project. The first one is to determine the mechanical design of the shooting part and the other to choose the motor to be used for this shooting. When determining the mechanical design for the shooting system, it was necessary to select the one that would provide the best shooting experience. For this reason, we tried to develop different solutions by making some 3D drawings. The program which is called “Rhinoceros” is used for 3d modeling of the shooting subsystem and “Keyshot” is used to get the end product image of the system.

For shooting mechanical design part, our main solution axonometric perspective can be seen in Figure 5.8.

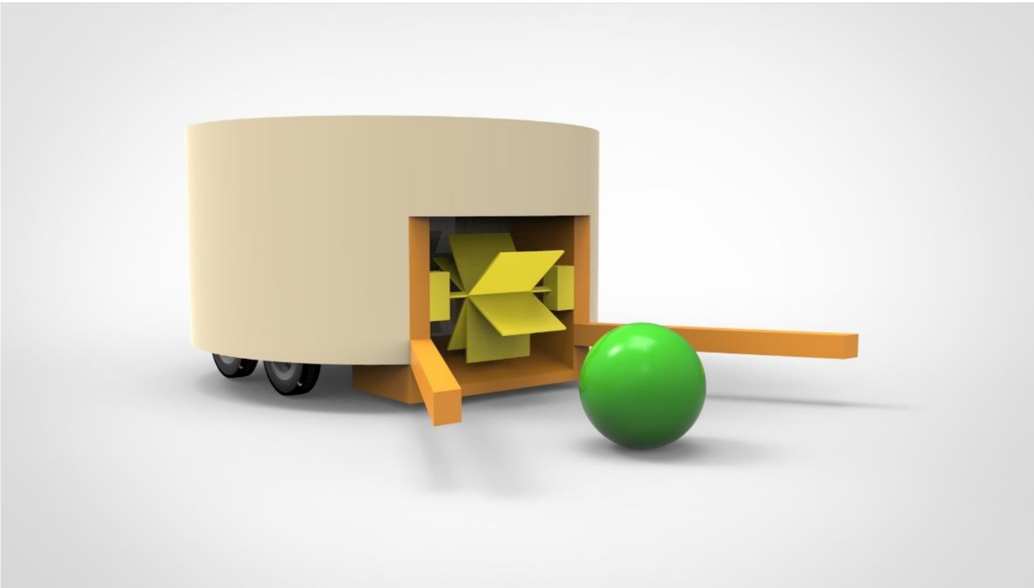

Figure 5.8: First shooting subsystem Design Axonometric View

According to our experiences, there should be two angular sticks that are parallel to ground for holding the ball closer to rotational shooting part.

However, the size of the goal in our project will be determined by the size of the widest part of the robot, the decision to use these sticks will be made after the demo matches. Therefore, it was decided not to use the sticks in backup design while keeping the sticks in the main solution.

In the Figure 5.8, the orange ones are representing these sticks. The radius of the ball is 4 cm and it gives reference for the rest of the model. The length of the sticks are 6 cm and the radius of larger cylindrical part is around 20 cm. The yellow parts represent the rotational shooting system.

For better understanding, this design can be examined in two different aspects in Figure 5.9 and Figure 5.10.

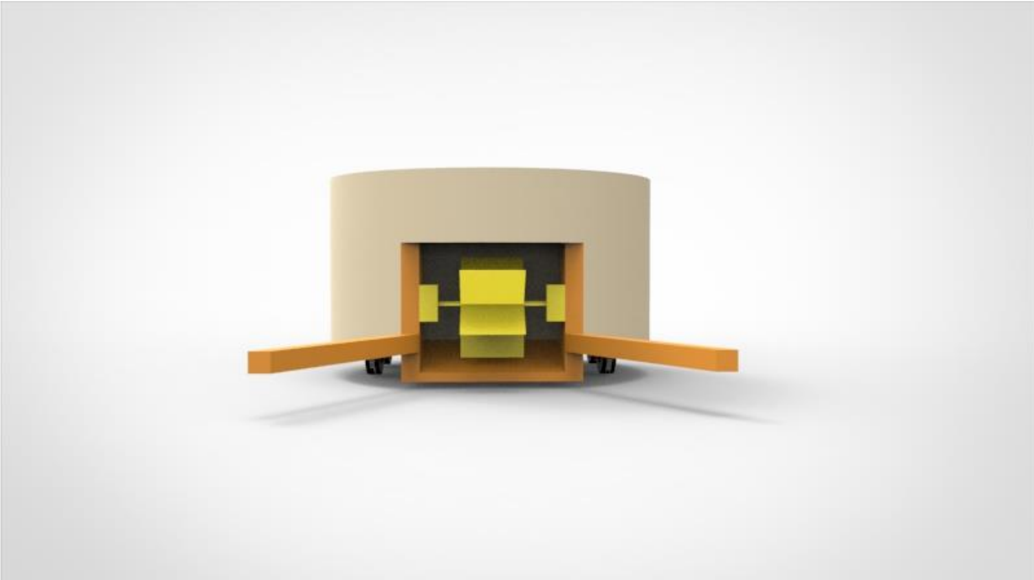

Figure 5.9: First shooting subsystem design front view

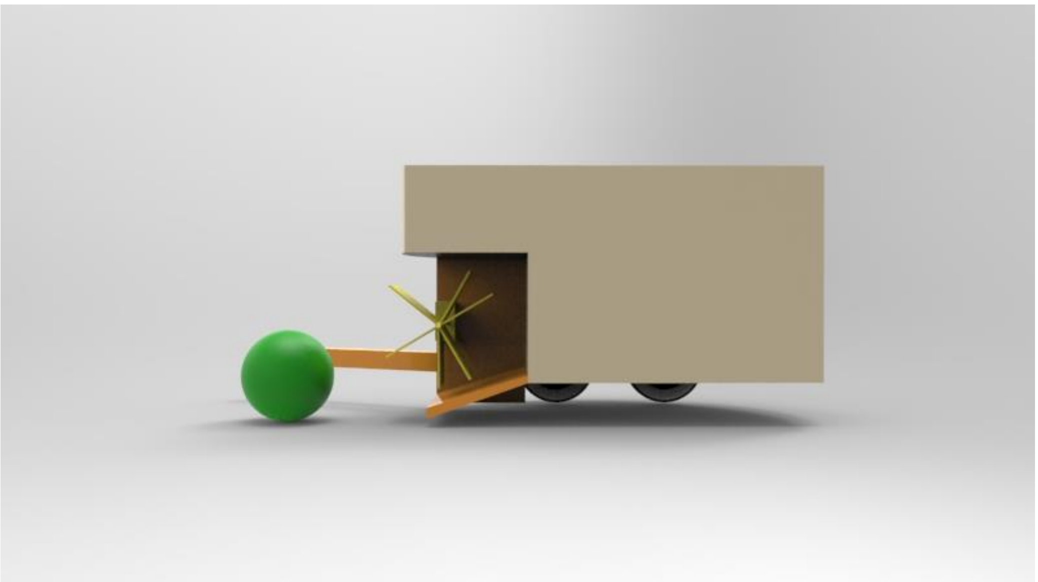

Figure 5.10: First shooting subsystem design side view

For shooting mechanical design part, our backup solution axonometric perspective can be seen in Figure 5.11.

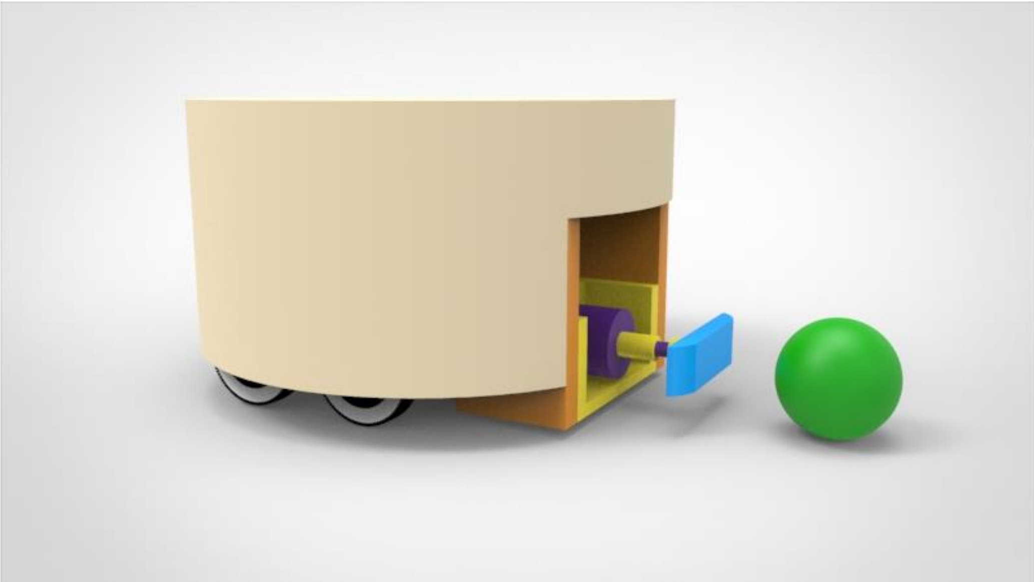

Figure 5.11: Second shooting subsystem design axonometric view

In this design, the part indicated by the blue part comes out of the robot and hits the ball. It is compressed by the spring system inside and then it is released so that the ball is hit.

For this application, it was decided to construct a spring actuated shooting device.
To wind the spring up, a spindle with nut is used. The nut presses against the spring which is thereby compressed. A lock and release mechanism hold the plunger in place and releases it when needed.

The time needed for the winding of the spring is the most important problem for this design. Therefore, the determination of the spring type and maximum force of the motor will be the most important stages.

For better understanding, this design can be examined in two different aspects in Figure 5.12 and Figure 5.13.

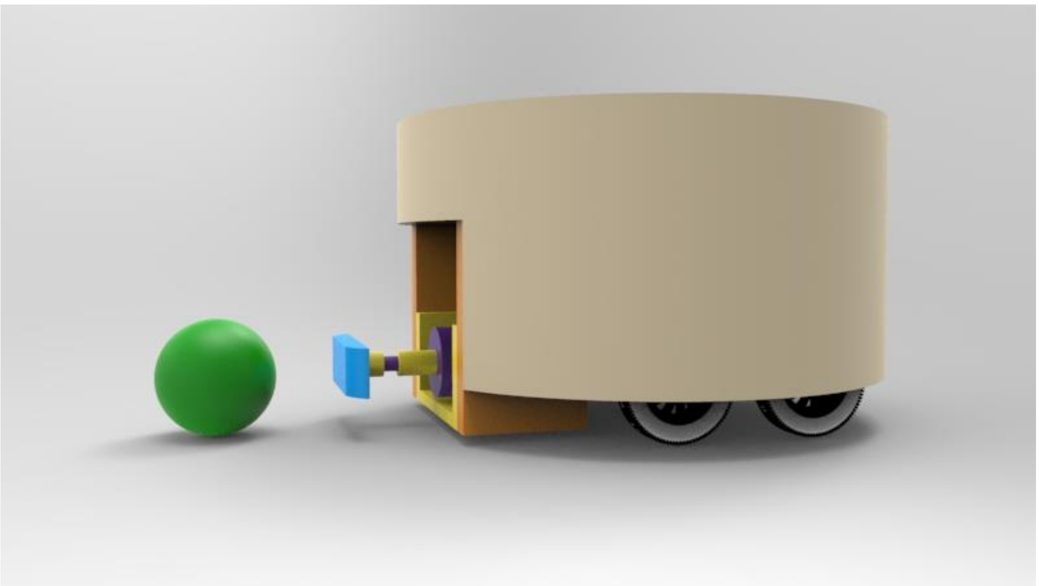

Figure 5.12: Second shooting subsystem design side view

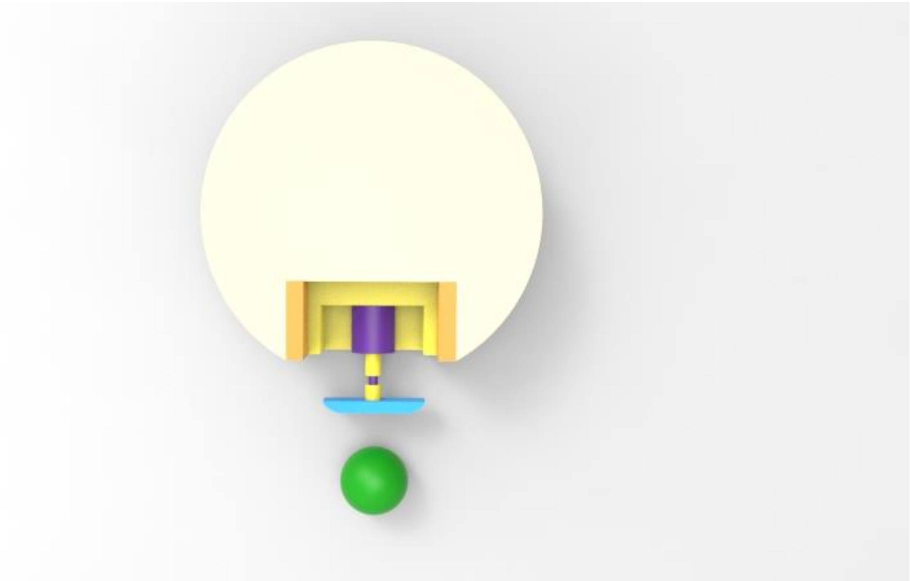

Figure 5.13: Second shooting subsystem design top view

For choosing the motor of shooting system is the second topic for shooting system. There can be 3 motor types to use in shooting part and main solution is usage of DC motors for that.

DC motors are the most widely used engines in robotics. Also, the DC motors have magnets instead of the coils in the stepper motors. The DC motors are available in a variety of versions with or without gearbox, which have different operating voltages and rpm values.

Speed control of DC motors can be done with PWM. DC motors are cheap, small and effective. Also, the wide variety of sizes, shapes and power is another reason for the frequent use of DC motors. For these reasons, they are ideal motors for using as a shooting part motor.

Also, the usage of Step Motor is our second solution. Stepper motors are electric motors which are surrounded by a magnet or a metal rotor and a coil that creates an inductance of voltage by creating an electromagnetic field effect. They are also suitable for sensitive applications such as robot arm applications and laboratory robots. However, areas of use are narrowed due to the heavy weigh, high current draws, low torque and difficulty in assembling in robotics. Hence, this type of motor will remain as our backup plan.

The usage of Servo Motor is our third solution for shooting system. Servo motors are very used in robot operations, especially since they are very easy to control and do not require any control circuitry. Although servo motors have advantageous features such as easy control and installation, compatibility with robot projects and having their own gearbox; they are not as popular as dc motors due to their costly, low torque and ineffective speed controls.

5.3.1 Level Risks Assessment

There are few risks of the shooting system.

- This subsystem heats a lot which can create a risk for other parts of the system.
- The command to be given from the controller can be corrupted.

5.3.2 Error Sources

Some internal and external sources can cause an error at this subsystem. The possible error sources are as follows:

- Environmental temperature conditions can be a source of error. To eliminate this effect heat sink or fan can be used.
- Overflow can be an error caused by increasing the current coming to the system. To eliminate this a voltage comparator can be used so that we should be sure that we are not feeding the systems a voltage higher than 12V.
- Wrong command can be the problem for shooting system. To eliminate this, we can use more accurate signals.

5.4 Motion Subsystem

5.4.1 Solutions and Relevant Algorithms

5.4.1.1 Plan A

Motion subsystem consists of motors for driving the robot on the play field, wheels, other assembly parts such as encoders and gearboxes attached to these motors and also driver IC which convert inputs from the Main Processor Subsystem into meaningful inputs for the hardware part of the motion subsystem. Motion subsystem also sends feedback to Main Processor Subsystem.

5.4.1.1.1 Motor Selection

Motor selection is highly crucial for building a stable and accurately driven robot. Before choosing the most suitable featured motor we have to decide which motor type will be most suitable for our purpose. Available motors in the market are DC Motors, Stepper Motors and Servo Motors.

5.4.1.1.1.1 Servo Motors

Using servo motors for driving the robots seems unreasonable since the working principle of servo motors are rotating the shaft to a desired angle with respect to its input duty cycle.

5.4.1.1.1.2 Stepper Motors

Since steppers move in precise repeatable steps, they excel in applications requiring precise positioning such as 3D printers. However, we don’t need precise positioning in motion subsystem. Normal DC motors don't have very much torque at low speeds. A stepper motor has maximum torque at low speeds, so they are a good choice for applications requiring low speed with high precision. This feature is also not applicable for motion subsystem.

5.4.1.1.1.3 DC Motors

DC motors in the market are the ones that we will most likely use. We can modify these with respect to our torque and speed regulations. These are also easy to control with a simple encoder integration. Other than reading how many turns have been made by the motors we may also consider using a motor driver IC to control how fast our motor turns as well. For DC motors driver IC simply turns the voltage on and off to vary the speed of the motor. PWM is used as driver input which can be easily produced using an Arduino.

Therefore, we concluded that for motion subsystem DC motor is the best option. Since we need higher torque but lower rotational speed, brushed DC motor is chosen as the best fit.

Before selecting the suitable motor, we need to consider some calculations. Such as total weight, torque and speed estimations of the robot. Total weight estimation for the robots excluding the motion subsystem motors are as in Table 5.2. Suitable motor for shooting subsystem is included in Shooting Components section.

Table 5.2:Weight Estimation Table of the Robot

Components	*Approximate Weight (g)*
Arduino MEGA	37
Powerbank	200
LiPo Battery	230
Chassis	150
Radio	10
Shooting Components	500
Cables	50
Other Circuitry	100
Total Weight	**1227**

For speed estimation we started the process with a reasonable assumption. For our purpose we assume that the nominal robot speed is 0.4m/s**.** It should be enough. The maximum speed will depend on other factors, which are yet to be identified. In order to calculate the rpm of the shaft needed to drive the robot with 0.4 m/s velocity we can use the following equation 1 below:

$$N_T = \frac{60\ x\ v_N}{\pi\ x\ D_W} \quad (1)$$

Where N_T is traction wheel rotation speed, v_N is nominal robot speed and D_W is wheel diameter. We take D_W as 0.08 m for starter. Therefore, calculated N_T is approximately 95 rpm on the wheel side. Since we are planning to use gearbox in order to increase the torque and decrease revolution per minute, rpm of the shaft will be 48 times the gear box ratio and the torque produced by the shaft should be 1/ (gearbox ratio) times the needed torque to move the robot with desired acceleration. We chose gearbox ratio as 60:1 at first. Then needed rpm for the desired nominal speed became 95 x 60 = 5700 rpm. Considering DC motor characteristics in Figure 5.14 which satisfies our speed and torque requirements. At ~5500 rpm produced torque is ~2.5 oz-in which is equal to 0.0176 Nm. One important thing is that this torque is on the shaft side. Since we took gear ratio as 60:1, torque on the wheel side becomes 60 x 0.0176 = 1.056 Nm. Considering the radius of the wheel, produced force at this speed is equal to 1.056 Nm / 0.04m = 26.4 N. However, since the efficiency is found to be 70 % at this level of speed, only 70 % of the torque and force will be transferred (~18 N) which is highly enough for driving our robot taking the overall weight as 2kg. Since without the DC motors weight was as calculated in Table 5.2. We took each DC motor as approximately 300 gr.

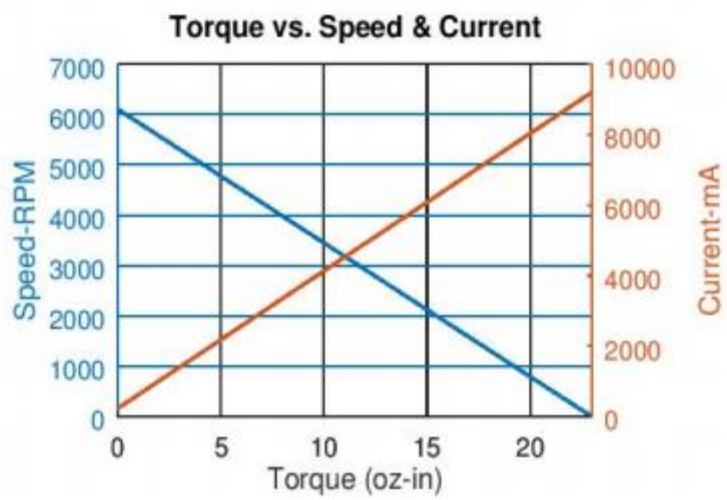

Figure 5.14: RS-555 12 V 6100 rpm Brushed DC Motor Characteristic [6]

5.4.1.1.2 Encoder Selection

After deciding on DC motor specs, we need to choose an encoder to give feedback to the Main Processor Subsystem. Therefore, we can directly send commands to the driver IC how many turns that the shaft should complete to reach to the desired location. After some research on encoder types there are two possible types of encoders in the field which are:

- Incremental Encoders
- Absolute Encoders

Both type of encoder components and a pulse train produced by Incremental Encoder is as in Figure 5.15.

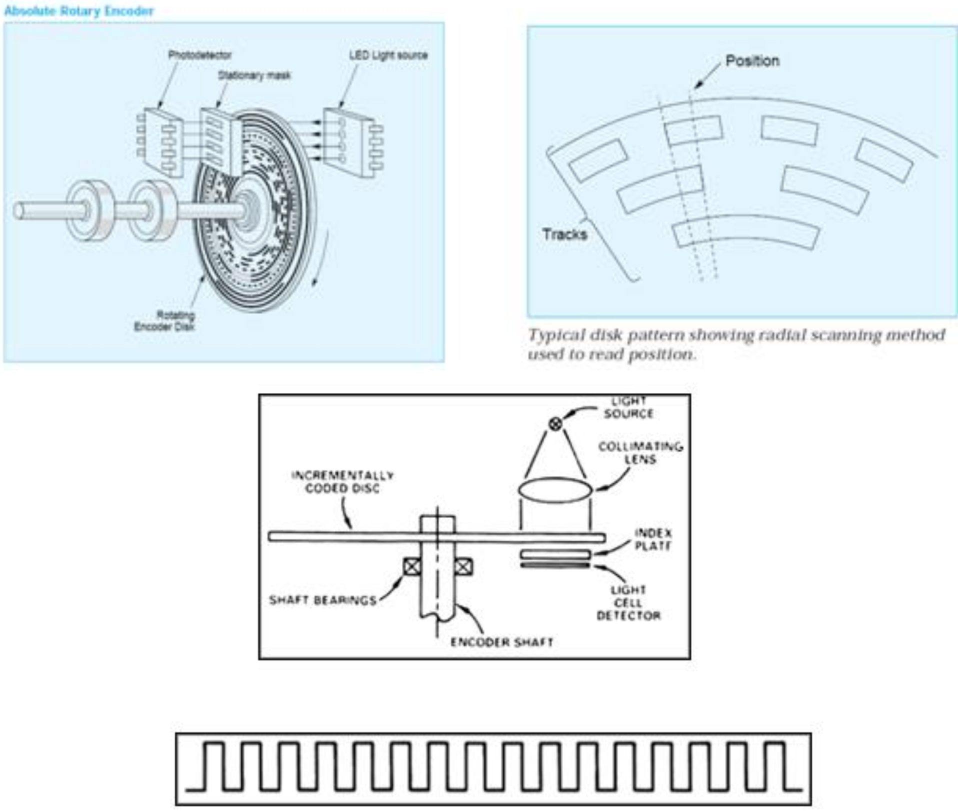

Figure 5.15: Incremental (Right) & Absolute Encoder (Left) Components [7]

First type is used for speed and position tracking. In this type turn count is measured easily but position accuracy is not available.

Second type of encoders are used for accurate position tracking since each position have specific ID so that we can simply know the exact position of the shaft attached to the encoder.

Incremental encoder seems to be more useful for our purpose. Since in motion subsystem we will command the motors to turn specific number of turns to reach the desired position or achieve the desired turn to the left or right. The encoder type we consider using, has Hall-effect sensor in it and counts up to 44 for one single turn but it should be multiplied with the gear ratio. Hall effect encoders use magnetic phased arrays that contain hall sensor elements arranged in a pattern to match a magnetic wheel. A signal is produced as the sensor passes over the magnetic field which is then interpolated to the desired resolution. The representation is as in Figure 5.16.

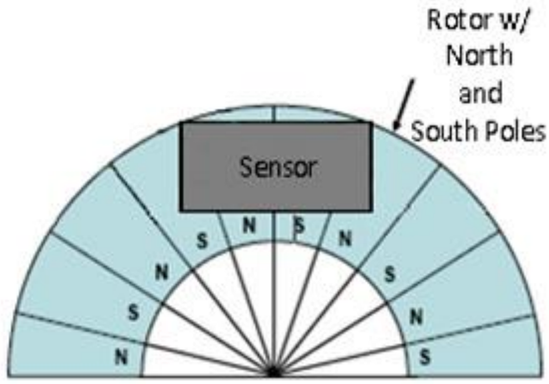

Figure 5.16: Magnetic Encoder Representation with Hall-effect Sensor [8]

For our specific type of encoder and gearbox ratio (60:1) every 2640 count means one full turn. Both CW and CCW is available. This module will be used in motor driving part. It will also prevent the turn mismatches between the right and the left wheel for example. We will use this encoder count as a feedback between tires to make our robot move more accurate and straight. Test code and related outputs are presented in the following parts.

5.4.1.1.3 Wheel Selection

There are different variety of wheels in the market such as standard, orientable, ball, and omnidirectional wheels. Our main solution for wheel selection is two standard wheels and also two ball wheels to stabilize the robot chassis. Wheels-chassis integration is represented as in Figure 5.17.

Figure 5.17: Wheel & Chassis Integration Representation [8]

By using the integration illustrated in Figure 5.17 above, we will drive the robot via Differential Drive method. This method can be visualized as in Figure 5.18.

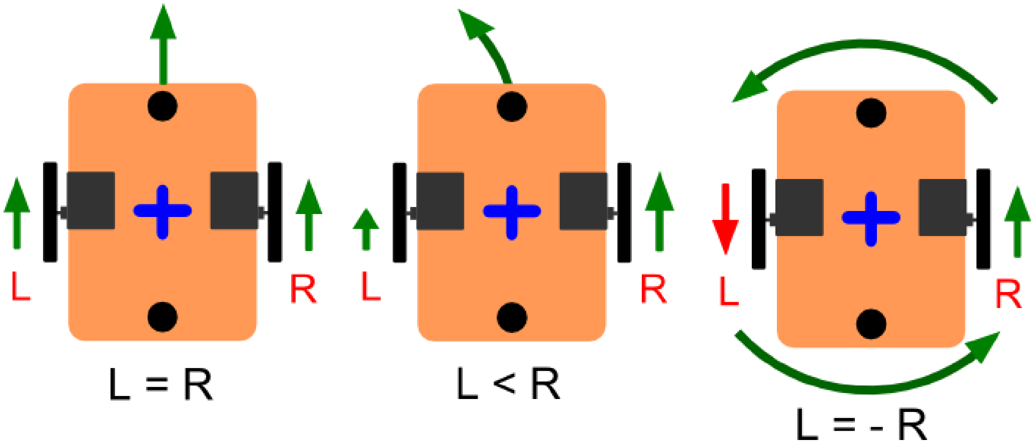

Figure 5.18: Differential Drive Kinematics [9]

5.4.1.1.4 Motor Driver Selection

Our main solution includes using L298N driver which can be easily integrated with Arduino. The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. We will drive two DC motors using this IC with respect to the pin connections as seen in Figure 5.19. Related test codes and results are presented in the following parts.

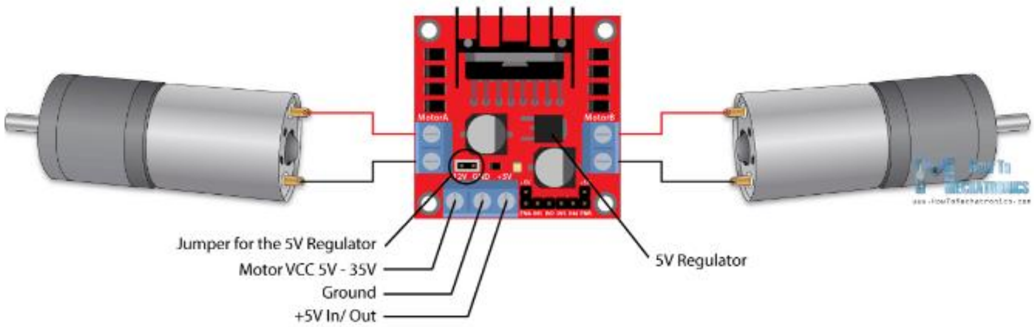

Figure 5.19: Pin Connections of L298N Driver IC [10]

5.4.1.2 Level Risk Assessments

There are several risks of our main solution approaches such as motor torque and speed specs may be different in practice therefore, we may need to reconsider our DC motor selection. Other than that, standard wheels may not be more efficient than omnidirectional wheels or we may not need two ball wheels. Encoders we chose may not be durable or since our main solution is for using magnetic encoders it may be affected from outside effects therefore it may give wrong readings to our Main Processor Subsystem. This will prevent our robot from moving accurately.

5.4.1.3 Plan B

Alternative solutions for Motion Subsystem are as below:

5.4.1.3.1 Motor Selection

DC motors with different specs such as rpm, torque, power and efficiency values can be used alternatively.

5.4.1.3.2 Encoder Selection

Encoders with different CPR (counts per revolution) values or different types can be used to isolate the encoders from environmental effects, with DC motors.

5.4.1.3.3 Wheel Selection

Omnidirectional or orientable wheels can also be used with DC motors.

5.4.1.3.4 Motor Driver Selection

L293DNE IC as in Figure 5.20, can be used to drive the DC motors. This IC is a quadruple high current half-H driver. These devices are designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors.

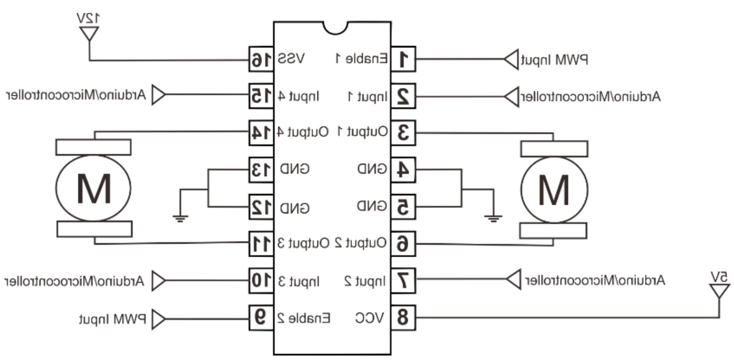

Figure 5.20: L293DNE Pinout [10]

5.4.2 Error Sources

Possible error sources for motion subsystem are as below:

- Since we are using two DC motors, we may face calibration issues. Even though these two motors are identical theoretically, in practice they will have minor differences. Therefore, we should consider this and update our algorithm respectively. A feedback should be available to prevent this calibration differences.
- Opening voltages of these DC motors can be slightly different. Therefore, when we apply a voltage for starter one motor can start rotating whereas the other does not. We should conduct relevant tests to compare two DC motors.
- We should adjust the center of gravity therefore weight should be equally transferred into the wheels and motors respectively. We may not place the center of gravity well enough which will cause one motor to turn more than the other. We may need to use a PID controller and tune it to obtain optimal results.

5.4.3 Test Results

In Motion Subsystem we implemented a tutorial for both encoder readings and also motor driving. Encoder reading test code is as in Figure 5.21 -5.22.

```
int encoderPin1 = 2;
int encoderPin2 = 3;

volatile int lastEncoded = 0;
volatile long encoderValue = 0;

long lastencoderValue = 0;

int lastMSB = 0;
int lastLSB = 0;

void setup() {
Serial.begin (9600);

pinMode(encoderPin1, INPUT);
pinMode(encoderPin2, INPUT);

digitalWrite(encoderPin1, HIGH); //turn pullup resistor on
digitalWrite(encoderPin2, HIGH); //turn pullup resistor on

//call updateEncoder() when any high/low changed seen
//on interrupt 0 (pin 2), or interrupt 1 (pin 3)
attachInterrupt(0, updateEncoder, CHANGE);
attachInterrupt(1, updateEncoder, CHANGE);
```

Figure 5.21: Encoder Reading Code Part 1

```
void loop(){
//Do stuff here

Serial.println(encoderValue);
delay(100); //just here to slow down the output, and show it will work even during a delay
}

void updateEncoder(){
int MSB = digitalRead(encoderPin1); //MSB = most significant bit
int LSB = digitalRead(encoderPin2); //LSB = least significant bit

int encoded = (MSB << 1) |LSB; //converting the 2 pin value to single number
int sum = (lastEncoded << 2) | encoded; //adding it to the previous encoded value
if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011)
encoderValue ++;
if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000)
encoderValue --;
lastEncoded = encoded; //store this value for next time
}
```

Figure 5.22: Encoder Reading Code Part 2

As explained before we must see 2640 after one full turn is completed both in clockwise and also in counterclockwise direction. The results were close enough to our expectation. There is an error margin at the results. Therefore, we need to take these error margins into consideration when driving the DC motors and writing our main algorithm.

Second tutorial was for driving the motors using L298N. Test Algorithm can be seen in Figure 5.23-5.24.

```
#define enA 9
#define in1 6
#define in2 7
#define button 4

int rotDirection = 0;
int pressed = false;

void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(button, INPUT);
  // Set initial rotation direction
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
}

void loop() {
  int potValue = analogRead(A0); // Read potentiometer value
  int pwmOutput = map(potValue, 0, 1023, 0 , 255); // Map the potentiometer value from 0 to 255
  analogWrite(enA, pwmOutput); // Send PWM signal to L298N Enable pin

  // Read button - Debounce
  if (digitalRead(button) == true) {
    pressed = !pressed;
  }
  while (digitalRead(button) == true);
  delay(20);
```

Figure 5.23: DC Motor Driving Tutorial Code Part 1

```
  // If button is pressed - change rotation direction
  if (pressed == true  & rotDirection == 0) {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    rotDirection = 1;
    delay(20);
  }
  // If button is pressed - change rotation direction
  if (pressed == false & rotDirection == 1) {
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    rotDirection = 0;
    delay(20);
  }
}
```

Figure 5.24 : DC Motor Driving Tutorial Code Part 2

We controlled the speed of the motor using a potentiometer and change the rotation direction using a push button. In the loop section we start by reading the potentiometer value and then map the value that we get from it which is from 0 to 1023, to a value from 0 to 255 for the PWM signal, or that's 0 to 100% duty cycle of the PWM signal. Then using the analogWrite() function we send the PWM signal to the Enable pin of the L298N board, which actually drives the motor.

Tutorial code mentioned above is implemented with the circuit schematic as in Figure 5.25.

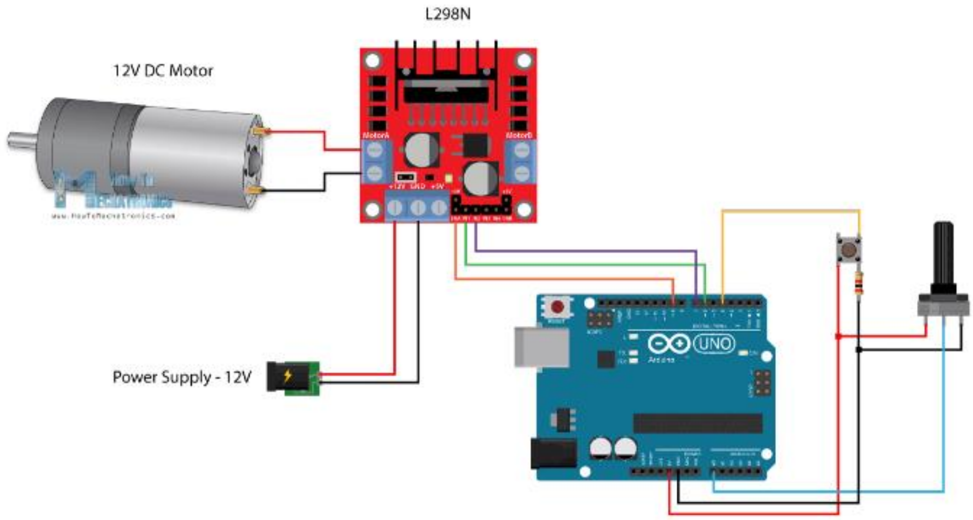

Figure 5.25: Circuit Schematic for Motor Driving Tutorial [10]

5.5 Detection Subsystem

The main aim of the detection subsystem is to create a field vision for the teleoperator. Since this subsystem is the only source of information for the teleoperator, it must gather all the important data around the robot. This system must gather lots of visual data with minimum delay.

5.5.1 Solution and Relevant Algorithms

5.5.1.1 Plan A

To design most suitable solution for detection unit, we decided to use a on board camera attached to the robot. Since the video data contains most of the valuable information the teleoperator needs, we decided to use an onboard camera for visual input. After obtaining the video input from the camera, the current frame of video is directly transferred to the telecontroller subsystem by communication subsystem without any processing.

Due to the competitive nature of this project, the delay between teleoperator and visual of playfield must be as short as possible. To decrease delay time caused by detection subsystem, we directly send the raw video data without any processing.

In this submodule we decided to use AKK KC03 800TVL NTSC Switchable Camera Module, shown in Figure 5.26, since it has a built-in transmitter, which we are going to use in communication subsystem. This camera module provides high quality video output for the telecontroller.

Figure 5.26 :AKK KC03 800TVL NTSC Switchable Camera Module 600mW FPV Transmitter [11]

5.5.1.2 Plan B

Despite being able to provide all the necessary information to the teleoperator, the directly sending raw data to teleoperator will cause some problems. In case of any problem, as a backup plan we have option to process the raw video data and send the processed data to the telecontroller. In this approach we need to develop an image processing algorithm which extracts all the necessary information from the raw image. The information retrieved from the frame is sent to telecontroller. In the telecontroller, the reduced image data is rebuilt to create a meaningful visual data

To obtain visuals of the playfield, the raspberry pi camera module, can be observed in in Figure 5.27, is the best available option, due to its compatibility with development board. The detailed information on development boards, which will be used in this case, is provided in main processor subsystem.

The working principle of this subsystem is, the camera takes current frame from the playfield and sends the data to processor. In the processor an image processor algorithm runs and obtains necessary information's from this frame. The necessary information is the location of ball, opponents goal, robot's location and midlines location. After obtaining this data, it transfers this data to telecontroller via communication subsystem.

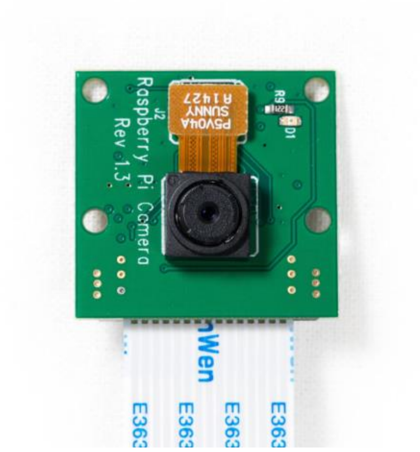

Figure 5.27: Raspberry Pi Camera module [12]

5.5.1.3 Plan C

Instead of being a backup plan, this plan is an enhancement plan for the plan A and B. Since the visual angle of camera is limited, it is impossible to obtain full field vision with a camera. Hence, to extent the vision of telecontroller a network of sensors will be added to the robot. The sensors are located in the blind spots of the camera and collect basic information from these regions. With the help of this sensor network, the vision of teleoperator will enhance.

When building sensor network two sensor option we consider two types of sensors, infrared proximity sensors and ultrasonic sensors. These sensors can be observed in Figure 5.28-5.29 respectively.

Figure 5.28: IR proximity sensor [13]

Figure 5.29: Ultrasonic sensor [14]

5.5.2 Level Risk Assessment

There are few risks of using the above-mentioned approaches. These risks are stated below

- In prebuilt system, all the elements are embedded into one module. In case of any problem or performance issue, it is very hard to implement any modifications.
- Due to high volume of data systems must work on very high frequency. This will cause overheating and if no precautions will be taken this seriously harm the robot.
- The precision of sensors is not stable, show variations in every run. In the development of the algorithm this must be considered.

5.5.3 Error Sources

Some internal and external sources will harm the operation of this subsystem. These sources are shown below

- The precision of detection systems heavily depends on environmental conditions such as heat, illumination, material of detected object etc.
- Voltage spikes in the power source will seriously harm voltage sensitive components. These sensors must be isolated from the rest of the robot.
- Cameras and sensors are sensitive components hence any hit on them will create irreversible damage on them. These components must be protected from any hit.

5.5.4 Test Results

After we connect the detection subsystem to the system, we run several test on the detection module to measure the quality and stability of sent data. To test the necessary qualities of the camera, we run the color visibility test under different environment.

For the test, we built a test setup which consist of our camera module, transmitter, receiver and screen. We sent obtained camera data to the receiver and visually examine from the connected screen, which we are going to use in telecontroller. To determine visibility of colors we show a color scale half meter across to camera and visually examine this scale from the screen. To determine the effect of different ambient light, we run the test on capstone lab and Culture and convention center. To determine the effect of illumination level, in the test locations we run the test on shadow (low level), on well-lit area(mid-level) and under direct light (high-level). In the Figure 5.30 our test results are shown.

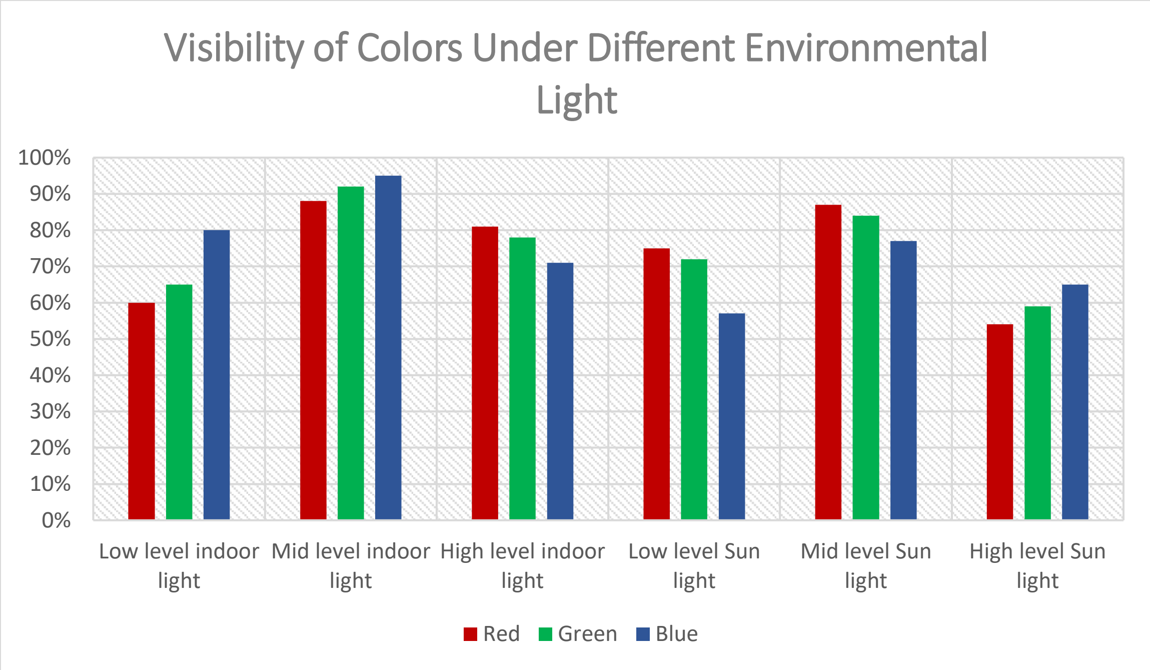

Figure 5.30:Graph of visibility of colors under different environment light for our camera

5.5.5 Comparative analysis

To have a better grasp on the proposed solutions, we have compared the properties of both systems. The comparison can be observed in Table 5.3.

Table 5.3 : Comparison of main and backup plan

	Main Plan *Direct video data transfer*	*Backup Plan* *Reduced video data transfer*
Data Size	900kByte per frame	10kByte per frame
Processing Time	None	0.1 sec
Data precision on Telecontroller	Original image	Rebuilt image
Price	10$	50$
Stability	High	Mediocre
Modifications	N/A	Available
Adaptability	N/A	Yes

As observed from the table our main plan is superior to backup plan in the most important fields. I can provide high quality data with no processing time. Moreover, it is inexpensive and more stable than its counterparts. On the other hand, the main limitation we faced in this plan is the unavailability of any modifications on the prebuilt module. In case of any problem, we are not able to modify the system. In this condition our backup plans come into consideration, since all aspects of this plan is open to modification.

5.6 Communication and Telecontroller Subsystem

As a company, we see the telecommunication system as the main design objective of this project. For this purpose, we developed unique communication link for sending commands to a remote location which will become the robot later.

5.6.1 Communication Link for Sending Commands

5.6.1.1 The Algorithm:

In this communication link, we utilized a Raspberry Pi 3 as a FM transmitter and used a FM modulated signal which contains command information from the tele-controller unit. Later, at the receiver side for the commands, at the robot, the FM signal is demodulated using a commercial FM radio and using an Arduino Mega 2560, demodulated signal which contains command information from the tele-controller unit is interpreted and respective commands are sent to motors and shooting mechanism. For better understanding, a simplified functional block diagram of the communication link for sending commands is given in Figure 5.31.

We embedded each command into different sine waves with different frequencies. This way, at the robot, sine waves are interpreted as commands and respective functions are executed. First, we have 5 basic commands to operate the robot:

1. Move forward
2. Move backward
3. Turn clockwise
4. Turn counter-clockwise
5. Shoot

For proper operation off full system, we need to overcome noise issue at the FM channel and get an accurate frequency detection and interpretation at the Arduino. The solutions for these issues will be mentioned later in details.

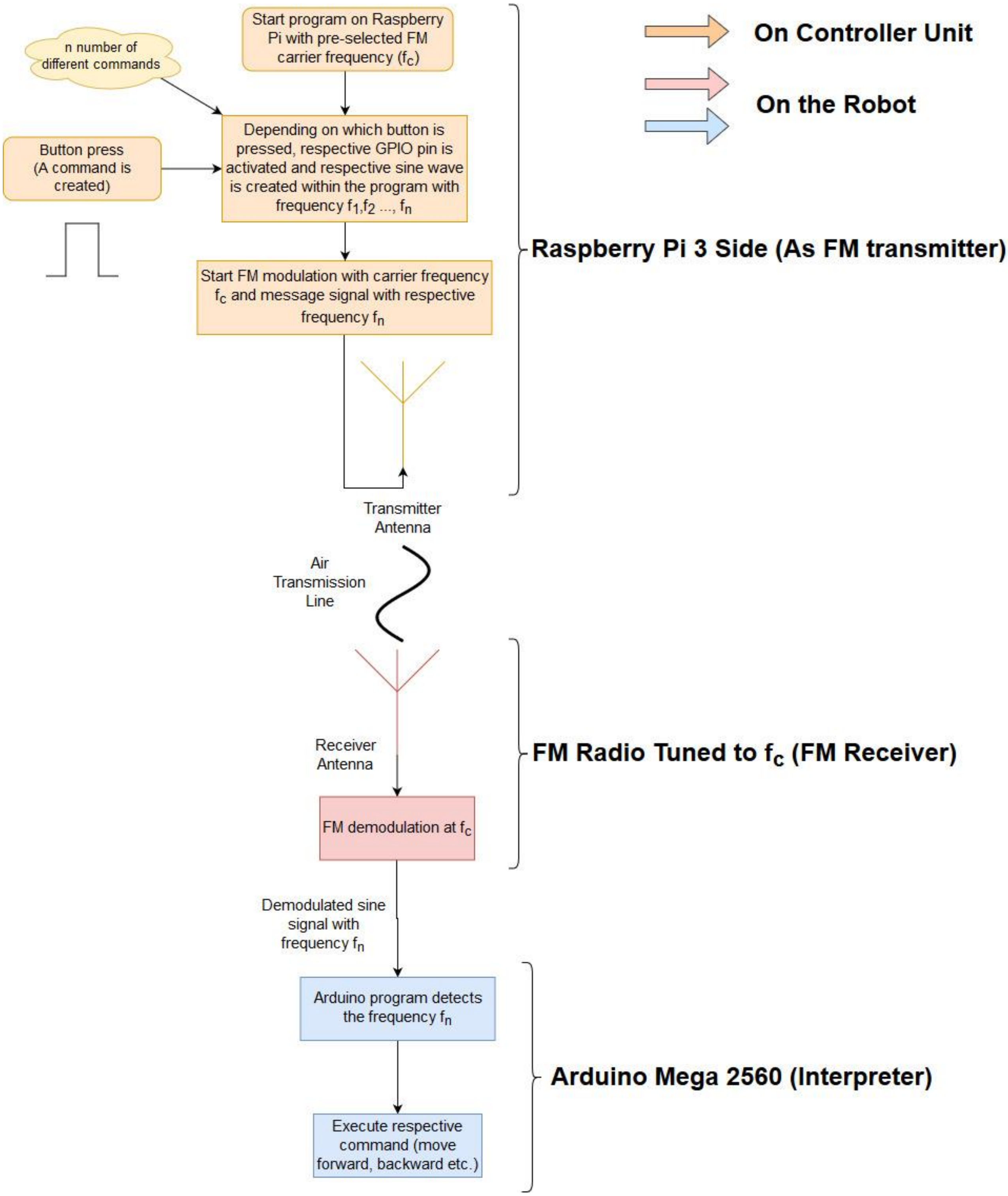

Figure 5.31: Block diagram of Communication link used for sending commands

5.6.1.2 FM Transmitter (Raspberry Pi 3):

We used Raspberry Pi 3 (Rpi3) to modulate and transmit a message signal which is a single tone sine wave and used its GPIO4 pin to transmit the FM modulated signal through an antenna. For this purpose, we find a GitHub repository for FM modulation on Rpi3. However, this program is only created to broadcast a .wav sound file. For our purpose, we needed to tweak the source code of the transmitter program. In our algorithm, we were to send different sine waves for different commands depending on which buttons pressed. First, we implemented a GPIO pin library to program in order to manipulate the program using GPIO pins (WiringPi Library). Later these GPIO pins are used for detection of which button is pressed. At this point, we decided to embed sine waves as arrays directly into the program instead of creating it in a running program. This way, the cost of computation time is reduced while the RAM usage is increased. Increase in RAM usage is not a problem because it is very small portion of RAM that is used for this purpose. After creating and embedding the sine waves, we connected the GPIO pin inputs with these sine waves. As a result, when a button is pressed, the assigned sine wave is modulated, and the modulated signal is sent to the antenna. A part of modified and tweaked transmitter code can be seen in Figures 5.32-5.34. For starter, we only implemented three different commands and therefore three different sine sequences.

```
float sin2k[44100]={0,0.28,0.54,0.75,0.91,0.99,0.99,0.91,0.76,0.55,0.29,0.01,-0.27,-0.53,-
float sin4k[44100]={0,0.54,0.91,0.99,0.76,0.29,-0.27,-0.75,-0.99,-0.91,-0.55,-0.01,0.53,0.
float sin6k[44100]={0,0.75,0.99,0.55,-0.27,-0.91,-0.91,-0.29,0.53,0.99,0.77,0.02,-0.74,-0.
  wiringPiSetup();
  pinMode(0, INPUT);
  int geri = digitalRead(0); //Reading initial values for commands
  int sag = digitalRead(2); //Reading initial values for commands
  int ileri = digitalRead(3);   //Reading initial values for commands
  bool first_entry_flag = true;        //this becomes TRUE after the first entry into
```

Figure 5.32 Embedded sine sequences

```
while (true) {
    geri = digitalRead(0);
    sag = digitalRead(2);
    ileri = digitalRead(3) ;
    if (geri == 1 && ileri == 0&&sag==0) {
    data = &(sin2k)[0];
    first_entry_flag = true;
    }
    if (geri == 0 && ileri == 1&&sag==0) {
    data = &(sin4k)[0];
    first_entry_flag = true;
    }
    if (sag==1&&geri == 0 && ileri == 0) {
    data = &(sin6k)[0];
    first_entry_flag = true;
    }
    if (geri == 0 && ileri == 0&&sag==0) {
    data = &(*frames)[0];
    first_entry_flag = true;
    }
```

Figure 5.33 Checking for GPIO inputs and assigning respective sine sequences in main transmission loop

```
 value = 1*data[offset];
PREEMP
 value = value + (value - prevValue) * preemp;
 value = (value < -1.0) ? -1.0 : ((value > 1.0) ? 1.0 : value);

 ACCESS(peripherals, CLK0DIV_BASE) = (0x5A << 24) | ((clockDivisor) - (int)(round(value * 16.0)));
```

Figure 5.34 Modulation and broadcast stage of the transmitter in main loop

In Figure 5.32, we declare three different sine sequences. In these arrays, each element represents voltage values for that sine sequence. In the program, depending on the input file, the sampling rate is determined. For the project, we used 44100 kS/s. So, each 1/44100 second a different voltage value is used for modulation. In Figure 5.34, the modulation index is determined such that the bandwidth of the FM signal is 200 kHz which is the bandwidth used for commercial radio receivers.

To choose a useful FM carrier frequency, we searched for empty FM channels between 87.5 MHz and 108 MHz. We found out that at 87.5 MHz, there was no channels, and, in our department, the relative noise levels are low. Choosing relatively low frequency is useful because observing the Friis' transmission equation(2) the free space path loss of the electromagnetic waves is less in lower frequencies.

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi r}\right)^2 \qquad (2)$$

After the modulation, the FM signal is to be transmitted via an antenna. We tried different topologies for antennas. First, a simple λ/4 monopole antenna. However, after some field tests, we saw that a monopole antenna could not provide adequate directivity for proper operation. After that we decided to use a commercial TV dipole antenna. After field tests, the results were satisfying, and therefore we stick with the dipole antenna. The physical length was quite long so we may utilize an RF amplifier reduce the antenna gain so the physical length later on the project timeline or we may use higher gain antennas such as dish antenna, Yagi-Uda antenna etc..

Finally, radiating the FM signal through an antenna to the air, the transmitter part is completed.

In the following months, we may completely eliminate Raspberry pi 3 for FM modulation purposes and use dedicated FM modulator modules to reduce total cost of the project.

5.6.1.3 FM Receiver (Beats by dr.dre Commercial FM radio):

After sending commands from the Rpi3 to the air, at the receiver side, the robot, we need to interpret these signals as commands. For this purpose, we decided to use a commercial radio. Because, they have built-in demodulation circuit and automatic gain control circuit (AGC). Also, they have better tuning capability then we can make in such a short period of time as the project timeline. To use a radio in our robot, we cut one of the earphones of the radio and soldered jumper wires to ends. By doing so, we have an electrical signal that have the command information which is a single tone sine wave. This signal will be fed to Arduino Mega 2560 for interpretation at later stage.

5.6.1.4 Interpreter (Arduino Mega 2560):

To use the received and demodulated signal, we need to interpret it as commands. For this purpose, we used Arduino Mega 2560. Since our commands are assigned to sine waves with different frequencies, we need to determine the frequency of the received signal. We implemented a piece of code which is given in Figure 5.35 to count the frequency of the input signal. As can be seen in Figure 5.35, we put a threshold value for detection of the signal periods. Also, to minimize the effects of channel noise on detection, we implemented a moving average filter which waits for three samples then averages them. After that, the average period of received signal is calculated. After measuring the period, simply the frequency is obtained. Later, this detection is used for motion subsystem to execute the commands.

Frequency_Counter §

```
Htime=pulseIn(41,HIGH);       //read high time
Ltime=pulseIn(41,LOW);         //read low time

Ttime1 = Htime+Ltime;

Htime=pulseIn(41,HIGH);       //read high time
Ltime=pulseIn(41,LOW);         //read low time

Ttime2 = Htime+Ltime;

Htime=pulseIn(41,HIGH);       //read high time
Ltime=pulseIn(41,LOW);         //read low time

Ttime3 = Htime+Ltime;
Ttime = (Ttime1 + Ttime2 + Ttime3)/3; // averaging 3 different samples
frequency=1000000/Ttime;    //getting frequency with Ttime is in Micro seconds

if( frequency < 6500 && frequency > 5500) //kırmızı
{ digitalWrite(23,HIGH);
}
else{
  digitalWrite(23,LOW);
}
if( frequency < 4500 && frequency > 3500) //sarı
{ digitalWrite(25,HIGH);
}
else{
  digitalWrite(25,LOW);
}

if( frequency < 2500 && frequency > 1500) //yeşil
{ digitalWrite(27,HIGH);
}
else{
  digitalWrite(27,LOW);
}
```

Figure 5.35 Arduino code for detection of the frequency of received signal

To choose proper sine frequencies to be assigned to a certain command, we needed to take into account that how well and with how much error rate can the Arduino can detect the frequency of the received signal. So, we performed an experiment to determine the percentage error rates at different frequencies form 50 Hz to 20kHz. The results of the experiment can be seen in Figure 5.36. Observing the Figure 5.36, we can say that at higher frequencies (>6kHz), the error rate is relatively high. And at low frequencies below 6 kHz, especially lower than 2 kHz, the error rates are between 1% and 2%. Although we used 2kHz, 4kHz and 6 kHz sines for the general test and submodule demo, for the final product, we will use different frequencies for the commands regarding the measured error rates.

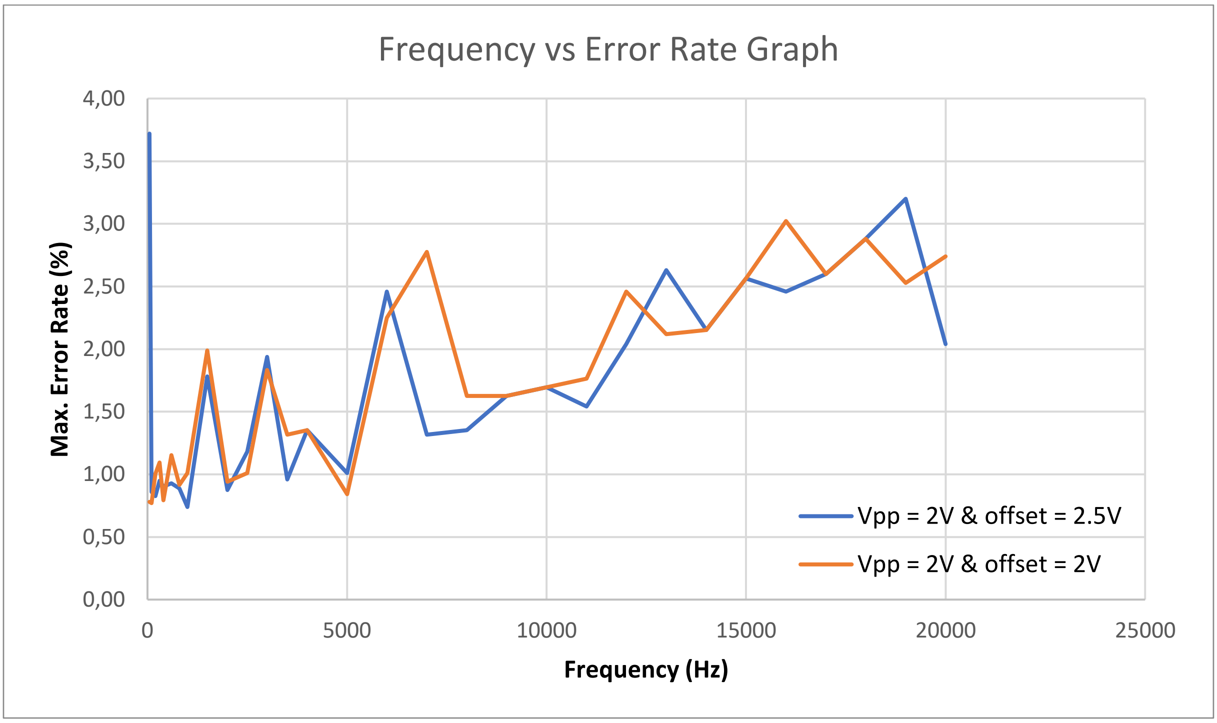

Figure 5.36 Maximum error rates of detected frequencies of different received signals

5.6.1.5 Error Sources for Command Sending System

- Antenna quality can reduce SNR in the transmitter side which is very crucial. At the transmitter we need SNR as high as possible because in the channel, a lot of noise is introduced due to environment. To prevent any issues, we will use commercially available antennas.
- Weather can affect SNR drastically. The path loss is heavily dependent on transmission channel. Rainy weather introduces electromagnetic interference due to high charges present in the clouds. To minimize the effects of weather, we will increase the transmitted power to keep SNR high enough.
- Misdetection of the commands on the receiver side can affect the overall operation very crucially. To minimize this situation, we will implement some digital filters while keeping command frequencies in minimal error rate bands.

5.6.1.6 Level Risk Assessment of the Main Solution

- Overheating on Rpi3 may result in irreversible damage on Rpi3. TO prevent this, we will utilize a heatsink and a cooler fan.
- Since physical dimension of our antenna is quite large, it may result in health injuries on people. To prevent unwanted accidents, we will use warning label which can be seen in Figure 5.37.

Figure 5.37: Example warning label

5.6.1.7 Test Result of the Main Solution

To determine which frequency band is more suitable for communication we run a set of tests. For this test we build a test setup, a transmitter and receiver station located fixed distances. We sent 30 sample data from every step for every frequency band from the transmitter and recorded at the incoming signals at receiver side. The test result is shown in Figure 5.38

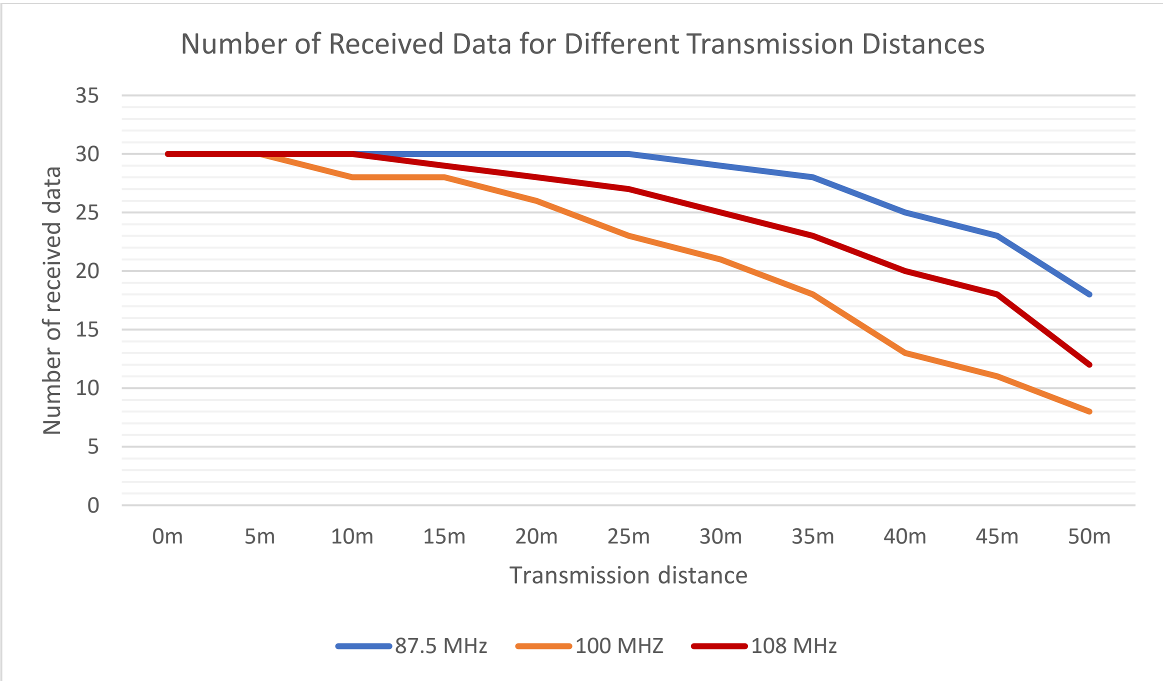

Figure 5.38: Graph of received data for various distances with different carrier frequencies

Observing the results, we decided to use 87.5 MHz carrier frequency because of its lower rate of data loss at farther distances.

5.6.1.8 Plan B:

Another approach from our main solution for sending commands to the robot can be implemented using commercially available RF transmitter-receiver modules for Arduino such as xBee and NRF24L01 given in Figure 5.39. In addition to these modules, we may implement a GSM module which uses GSM network to communicate.

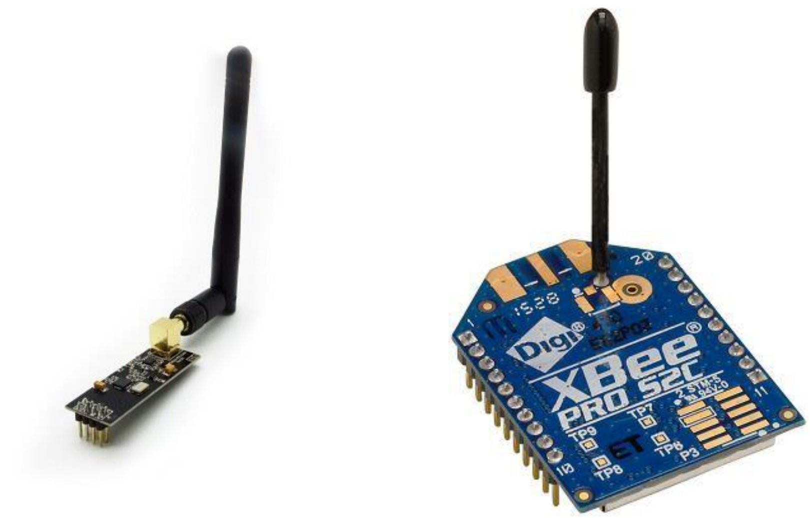

Figure 5.39: Communication modules NRF24L01 on the left xBee on the right[15]

5.6.2 Communication Link for Sending Video

To send instant images of the field to the telecontroller, we used an FPV 5.8 GHz drone kit AKK KC03 as transmitter and RC583 as receiver and last but not least, an LCD screen.

5.6.2.1 Video Transmitter

For video transmitter, we choose the module with relatively higher RF output power which is 600 mW. Because, since the frequency is very high, 5.8 GHz, the attenuation inside the walls will be very high and the RF signals will be prone to much higher order of magnitude of attenuation. The transmitter accepts 7 V to 20 V supply voltage. In this range, we can use 12V battery to supply enough power to both transmitter and motors. The image of the transmitter and its electrical characteristics can be seen in Figure 5.40.

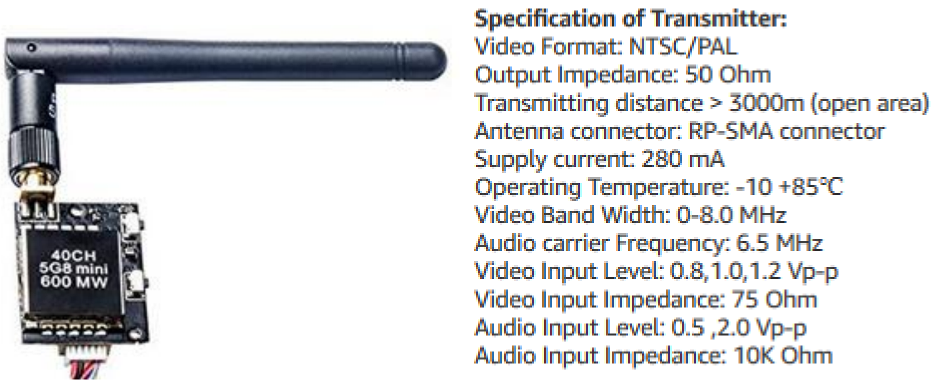

Figure 5.40: Video transmitter and electrical specifications[11]

5.6.2.2 Video Receiver

On the controller side, in order to see the field, we used a 5.8 GHz receiver kit. This kit shares the same protocols as our video transmitter which are NTSC/PAL. This receiver module also accepts 12 V supply, so we can use a common battery for the supply with the LCD screen which also accepts 12 V supply. The receiver module can be seen in Figure 5.41.

Figure 5.41: Video Receiver Module [16]

5.6.2.3 LCD Monitor

In order to monitor the video transmission from the robot, we implemented a 3.5-inch LCD monitor on our controller unit. The received and decoded video information from the receiver module is directly fed to the monitor and the clear view of the field is obtained.

5.6.2.4 Error Sources for Command Sending System

Main cause of error will be the interference issue. Since this kit will be used by most of the competitors, the limited number of transmission channels will make the clear video transmission a bit hard. To avoid such issues, we will buy an amplifier. As the transmission power increases, the interference immunity of our system would be increased.

5.6.2.5 Level Risk Assessment of the Main Solution

- For the transmitter, since it has power components in it such as a power amplifier, it will heat the module a lot, so overheating of the transmitter is a serious issue. To prevent any irreversible damages on transmitter, we implement a heatsink with a fan on top it.
- Considering the LCD monitor, it as fragile screen on it. So we will use the monitor carefully and embed the screen on a secured enclosure to prevent any physical damages.

5.6.2.6 Test Result of the Main Solution

During tests, we saw that the image on the monitor has kept its stability until 30 meters indoor distance. After this point, the image starts to flicker. We assume that multipath reflection may have caused these flickers. Other than that, the transmitter got really hot, so we installed a heatsink with a fan. After installation, the transmitter stayed reasonably cool.

5.6.2.7 Plan B:

Another approach for video or image transmission would be using GSM modules. As we make video calls from our phones, with these modules, we can make video calls to our robot. Other than that, we can implement a image processing system and use the command sending system to send processed image information to the controller.

5.6.3 Comparison

The ultimate benefit of our main solution is that the system is completely configurable. We can use any carrier FM frequency and we can use any sine message signal. We can call our method as modified frequency shift keying. The approach can be further developed for very high rates of data transmission. On the other hand, backup plans do not give any freedom on modulation or the protocol. The commercially available modules are easy to implement comparing our method. However, we are skillful engineers who can handle the challenge.

For the video transmission, the main advantage of our solution is that the FPV drone kit is really simple to implement and use. However, we may need to add an amplifier for the transmitter. The main drawback for the backup plans is is that they are relatively more complex.

6 Total Power Consumption

In the development of battery powered projects, development of the robot part in our case, power analysis is one of the crucial steps. Since the batteries are the only power source robot has, to ensure proper operation of robot for a predetermined time the power analysis must be done carefully.

In the total power consumption calculation of the system, we considered the worst-case scenario of each design element separately. Despite, this analysis method overestimates the power consumption, it creates a safety margin for us. In the case all the systems are active and at their full power, our system is going to be still fully functional. The power analysis of robot and telecontroller can be observed respectively in Table 6.1-6.2.

Table 6.1: Power analysis of Robot

Device	*Quantity*	*Maximum Values per Device*			*Total Device Power*
		Current	Voltage	Power	
DC Motor	2	300 mA	12 V	3.6 W	7.2 W
Drone Transmitter	1	400 mA	12 V	4.8 W	4.8 W
Shooting DC Motor	1	200 mA	12 V	2.4 W	2.4 W
IR Proximity Sensors	1	45 mA	5 V	225 mW	0.225 W
Arduino Mega	1	100 mA	5 V	500 mW	0.5 W
Total Power					**15.1 Watt**

Table 6.2: Power analysis of Telecontroller

Device	*Quantity*	*Maximum Values per Device*			*Total Device Power*
		Current	Voltage	Power	
Drone Receiver	1	10 mA	12 V	120 mW	0.12 W
Screen	1	400 mA	12 V	4.8 W	4.8 W
Antenna Driver	1	50 mA	12 V	600 mW	0.6 W
Raspberry Pi zero	1	200 mA	5 V	1 W	1 W
Total Power					**6.4 Watt**

7 Test Procedure

Test procedure mentioned below consists all the subsystems rather than the ones in demonstration. It can be seen as a User Manual for end user. Test procedure is as follows:

1) All switched should be in off position.
2) Li-po connections for data receiver and robot should be made.
3) Powerbanks should be connected.
4) All switches and powerbanks should be on for now on.
5) At this part we should observe some fan actions on the robot and receiver which are located for cooling purposes.
6) Moreover, video or image data transmission should be visible at this stage if not check the previous steps.
7) From now on robot should be controllable by telecontroller.
8) Start the test procedure when robot and telecontroller at the same place. (Approximately 0 m distance)
9) The buttons on telecontroller should create forward, backward, left and right movements. Also, not decided yet if any, shoot button controls the shooting subsystem.
10) Different combination of pushes will be tried during the process.
11) The steps 9 and 10 will be tried for different distances until the data transmission and command transmission both end.
12) At the end of the process it is necessary to wait at least 30 seconds without pushing any button in order to be sure that all motor activity ended.
13) The robot lifted up and switches should be off from now on.
14) Li-po batteries will be disconnected from the system for safety.

8 Plans Management

This section of the conceptual design report consists the plans, breakdowns, foreseeable difficulties and Gantt chart.

8.1 Planned Work

Detailed planned work can be seen in Table 8.1 below.

Table 8.1: Detailed breakdown of planned work

	Irem Coskun	Fatma Nur Arabaci	Aycan Beyenir	Berkay Goksu	Furkan Bahadır Elik
Telecontroller and communication subsystem integration					✓
Motion Subsystem integration	✓	✓			
Processor and signal receiving module implementation	✓		✓		
Detection algorithm verification and optimization			✓	✓	
Power system implementation to other subsystems				✓	✓
Testing of individual subsystem integrated modules	✓	✓	✓	✓	✓
Bringing subsystems together		✓	✓		
System Verification	✓			✓	
Optimization for motion system		✓			
Optimization for telecontroller					✓
Testing at different environment	✓	✓	✓	✓	✓
Final verification of systems	✓	✓	✓	✓	✓

8.2 Gantt Chart

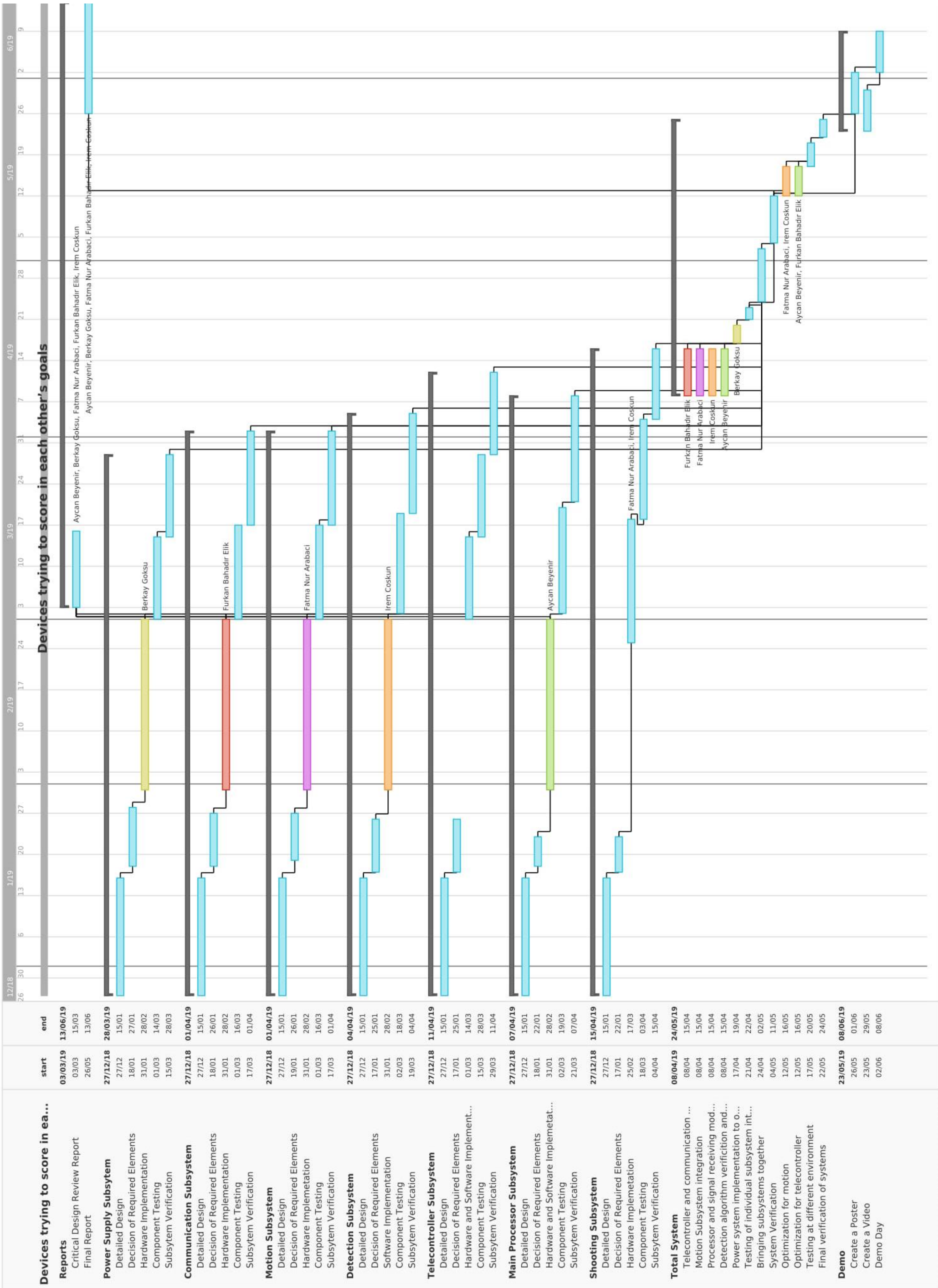

Figure 8.1: Gantt Chart

8.3 Foreseeable Problems and Solutions

In the implementation process of an engineering project, facing with problems and difficulties is something expected. The theoretical solutions not always work as expected in practical applications. As a newly formed company we are aware that, in order to develop a robust system, it is necessary for us to consider all the possible problems will harm the operation of system and come up with solutions to handle them. In this part we investigate foreseeable difficulties and problem sources we will face and propose their possible solutions. For the sake of clarity, we stated these possible problems and solutions in four subgroups.

8.3.1 Communication problems

- There might be a high level of noise in the communication band, which will decrease the quality of transferred data.
- The range of the communication dependents on the environment condition, and the maximum range might vary under different environment.
- The communication band might be used by another robot or local user, which causes both signals to mix-up and makes it impossible to communicate with robot.

Our communication system is designed to work on different channels. In case of any unexpected intrude to our communication band, we can maintain our communication with robot by changing the band. To withstand various environmental conditions the power of communication system is adjusted to cope with worst scenario.

8.3.2 Mechanical problems

- The movement of the robot might not be precision enough. This will make the control of the robot harder.
- The motors of the robot would not be fast enough and that will end up with time violation in attack and weakness in defense.
- If the force applied to the ball in the shooting is not well calibrated, it will decrease our control over the ball and decrease our chance to score.
- The weight of robot will deform the framing of the robot.
- If the sensitive parts and connections take hit from ball, they will be malfunction or lost connection.

To protect the robot from impacts, we add a protective shield covering the robot and all connections are going to be soldered. To increase precision of operation, feedback connections and algorithms will be integrated to system. The choice of motors and other mechanical parts will be done with additional safety margin.

8.3.3 Field vision problems

- Due to the limitation of visual angle of sensors, all elements on the field will not be visible at the same time to the operator.
- Location of sensors will create some blind spots around the robot.

To extend the field coverage some sensors will be attached to moving parts. Some of proposed visual systems can be used together to benefit from strengths of each.

8.3.4 Electrical problems

- Under continuous operation batteries, motors, development boards might overheat.
- Rapid changes in the current can cause voltage spikes, which will harm voltage sensitive elements such as ICs and sensors.

Heat sinks and fans are going to added to heat sources. Voltage protection circuits will be added in the supply of sensitive elements.

8.4 Future Test Plans and Measure of Success for Subsystems

To verify the overall operation of the robot, the subsystems must be verified first. To measure the performance of subsystems, we prepared future test plans and measure of success for subsystems.

8.4.1 Communication and Telecontroller Subsystem

- <u>Robot to Telecontroller Range Test</u>: The quality of sent data is measured for every 5 meters, starting from 10 meters. In indoor conditions if more than 60% of the visual data transferred to the telecontroller, then at this step the communication test is satisfied. The range more than 30 meter is satisfactory, and more than 40 meters is perfect score.
- <u>Robot to Telecontroller Delay Test</u>: At 30-meter indoor range the delay of the video input is measured. Between robot and telecontroller, delay less than 0.2 sec is satisfactory and less than 0.1 sec is perfect score.

- Robot to Telecontroller Data Quality Test: At 30-meter indoor range the quality of the video input is measured. Between robot and telecontroller, transfer of more than 60% of visual data is satisfactory and 80% is perfect score.
- Telecontroller to Robot Range Test: The quality of sent data is measured for every 5 meters, starting from 10 meters. In indoor conditions if more than 90% of the commands transferred to the robot, then at this step the communication test is satisfied. The range more than 30 meter is satisfactory, and more than 40 meters is perfect score.
- Telecontroller to Robot Delay Test: At 30-meter indoor range the delay of the commands is measured. Between robot and telecontroller, delay less than 0.2 sec is satisfactory and less than 0.1 sec is perfect score.
- Telecontroller to Robot Data Quality Test: At 30-meter indoor range the quality of the commands is measured. Between robot and telecontroller, transfer of more than 80% of commands is satisfactory and 90% is perfect score.

8.4.2 Motion Subsystem

All the motion subsystem tests are done with a fixed environment. For the test the telecontroller is located 2 meters across the robot since performance of communication module is not our concern in this test.

- Straight movement precision test: Under forward and backward movement instructions the distances robot traveled are measured and these measurements are compared with expected values. The 20% or less error is satisfactory and less than %5 error is perfect score.
- Straight movement speed test: For the forward and backward direction movement, speed of the robot is measured, and these measurements are compared with expected values. The 30% or less error is satisfactory and less than %15 error is perfect score.
- Fixed center rotation test: The robots ability to fixed center rotation is measured. For a 180° turn the robot expected to preserve its center at same spot. Dislocation less than 5cm is satisfactory and 1 cm is perfect score.
- Rotation precision test: Under fixed center clockwise and counterclockwise rotation instructions the rotation angle of robot is measured, and these measurements are

compared with expected values. The 20% or less error is satisfactory and less than %5 error is perfect score.

- Rotation speed test: For the fixed center clockwise and counterclockwise rotation, angular speed of the robot is measured, and these measurements are compared with expected values. The 30% or less error is satisfactory and less than %15 error is perfect score.

8.4.3 Shooting subsystem

All the shooting subsystem tests are done with a fixed environment. For the test the telecontroller is located 2 meters across the robot since performance of communication module is not our concern in this test. The ball is located 1cm front of the shooting subsystem with 60° - 90° - 120° angle with central plane of robot.

- Shooting direction test: For 10 successive tests, number of shots going in opponent's goal is counted. More than 7 successful shot is satisfactory and more than 9 successful shot is perfect score.
- Shooting force precision test: For 10 successive tests, the distance ball travels is measured. The measurement and expected data is compared. Error less than 40% is satisfactory and less than 20% is perfect score.

8.4.4 Detection Subsystem

All the detection subsystem tests are done with a fixed environment. For the test the telecontroller is located 5 meters across the robot since performance of communication module is not our concern in this test.

- Data quality: For 10 second of data transmission, the ratio of successful visual data transfer is measured. If all the important elements are visible, the test is successful.
- Data frame rate: For 20 seconds of data transmission, the frame rate is measured. Frame rate higher than 10fps is satisfactory and 20 fps is perfect score.

8.4.5 Power Subsystem

Power subsystem test are run separately on robot and telecontroller. For both the measure of success is same.

- No operation battery life test: When all units are on and under no operation condition the duration of batteries are measured. If the battery stays longer than 3 hour it is satisfactory, if stays longer than 5 hour it is perfect score.
- Active operation battery life test: When all units are on and under control of teleoperator the duration of batteries are measured. If the battery stays longer than 1 hour it is satisfactory, if stays longer than 1.5 hour it is perfect score.

8.5 Cost Analysis

Component price table is as in Table 8.2.

Table 8.2: Cost Analysis

Materials	Number	Price per each	Total Price
Clone Arduino Uno	1	$5	$5
Raspberry Pi 3	1	$40	$40
DC Motors	2 +1	$10 / $5	$25
Gearbox	3	$5	$15
Cables & Connectors	nAn	$6	$6
Battery	2	$15	$30
Dummy Robot	1	$2	$2
Balls	2	$0.25	$0.5
Chassis	1	$10	$10
Encoders	2	$5	$10
Play Field Walls	3	$1	$3
3D Printed Parts	nAn	$8	$8
FPV Drone Kit	1	$35	$35
Wheels (Standard + Ball)	2 + 2	$2 / $1	$6
Sum :			*$195.5*

8.6 Deliverables

As Potato Integrated Technologies, providing creative and reliable solutions to our customers is a matter of paramount importance for us. To accomplish the highest customer satisfaction, we provide high quality robotic solutions specified for the problem. Therefore, the final product that is going to be served to the customer is guaranteed to satisfy our policy of quality.

After the final product satisfies all field test, our customers will receive a complete delivery of the main unit, auxiliary equipment, documentations and software. The robot unit, auxiliary equipment, documents and software that are going to be in the delivery packet are summarized below

Main Unit: This package contains the main delivery; the robot unit which will take a part in a competition where it tries to score to the opponents' goal and protect its goal and a controller unit which the operator going to use for teleoperate the robot unit.

Auxiliary Equipment: This package contains equipment necessary to fully operate the robot and build the competition environment. The pack contains; computer to robot communication module,1 Li-Po battery (one main), 1 Li-Ion battery (one main),3 Walls for game field, self-goal (at least twice the size of robot unit), two ball (diameter 30-45 mm) (one is spare), a dummy robot (for demonstration and sparring purposes).

Documents: This package contains the all the necessary documents in hardcopy. This pack contains; one instruction manual, complete list of part used in the main unit and their reference codes, a 2-year warranty and a 5-year spare part support.

Software: This package contains a digital source of necessary documents and computer interface. The computer interface has two purposes; first one is making necessary adjustments on the robot for different environmental condition and second one is for teleoperating the robot using the computer as a teleoperator. The digital documents consist of; 3-D models of 3-D printed parts on SOLIDWORK and digital versions of user manual and complete list of part used in the main unit, their reference codes and their datasheets.

9 Conclusion

The main focus of this conceptual design report is presenting the solution ideas for the previously stated problem, “Devices Trying to Score in Each Other’s Goals”, namely a robot that communicates with the controller and via remote operator, tries to score on opponent’s goal. In addition to the main solution ideas, other solution alternatives are also given for the sake of completeness in case of lack of achieving our high- performance goals. Beside of the main and backup solution proposals, analysis of subsystems, test plans, error analysis, risk assessments and comparison of solutions are also explained in detail. The challenge of communication with a remote unit and using it to operate a precise robot is mainly the interference issues and the robust design of the robot. Especially when most of the competitors are using somewhat similar approaches. Our state-of-the art technological innovations as Potato Integrated produces simple, creative and robust solutions for reliable communication and mechanical systems by means of five electrical engineers each competent in their subjects. A superior robot with excellent and robust communication and clever mechanical systems is our main objective to deliver. As pointed out formerly the task of the final product is to be controlled by a remote operator and try to score on opponents’ goal. Reliability and range of the communication system are crucial factors and therefore an innovative solution with FM modulated signals and FM radios is proposed as main solution for commands to be sent to the robot. Until this time being Potato Integrated have worked to achieve the necessary steps to build important parts of the subsystems as well as a heavily conducted literature research to compare the trade-offs of each possible approach to decide on the optimal one. Briefly, the FM method for commands and FPV drone kit method for the image showed successful results. Electrical parts with embedded codes that Potato Integrated have developed are nearly completed. Mechanical parts are to be designed with the same innovative and creative spirit. Range and noise immunity will be improved in the final assembled product. The integration process will follow up the completeness of all of the modules. Motion solutions have not been implemented yet thus leaving space for alternatives. During this period, several tests will be conducted as explained earlier. Most importantly, driving motors mentioned in motion subsystems hasn’t been focused on yet while being one of the crucial parts of the robot. Hence, this will be carefully investigated and implemented. The meticulous team of Potato Integrated also provides secondary solutions for all sub blocks. These back-up plans are essential for any process of a true engineer since the theoretical circumstances may not meet the practical ones. As a result, we may need some necessary modifications or a whole new strategy to accomplish a defined task. However, our main goal is using the most practical and simple solution that would not create redundant complications for this product is to be used in various real-life applications. These applications would reflect the social impacts and importance of the produced robot and controller and may create the basis of a higher scoped tool varying in size, objective and performance. Potato Integrated will deliver an exceptional final product together with the deliverables, within a short timeframe and economic budget. Our aim as an ambitious company to thrive on everyday problems with topnotch products with performance and durability, representing superiority within the market by virtue of all invaluable company partners.

10 Disclaimer

We, as Potato Integrated Technologies, hereby confirm on 26th December 2018 that our design process complies with the decision that the standard committee made on the project “Devices trying to score in each other’s goals”. We are not subjected to a penalty or a review for this purpose henceforth, nor do we accept any.

Fatma Nur Arabacı | İrem Coşkun | Furkan Bahadır Elik

Aycan Beyenir | Berkay Göksu

11 References

[1] Battery chemistry comparison, Retrieved from https://static4.arrow.com/-/media/arrow/images/miscellaneous/0/0717-battery-chemistry-comparison-chart.jpg?la=en&hash=5A3A6E8E9C68043D70609CF0C583A3C8522A397B

[2]How to select the right battery for your application, Retrieved from https://www.dfrsolutions.com/blog/how-to-select-the-right-battery-for-your-application-part-1-battery-metric-considerations

[3] Li-Po battery, Retrieved from http://www.helipal.com/storm-11-1v-2200mah-35c-pro-series-li-po-battery-xt60.html

[4] Wendt, Z. (2017, December 26). 5 Essential Factors for Choosing the Right Battery. Retrieved from https://www.arrow.com/en/research-and-events/articles/choosing-the-right-battery-for-your-internet-of-things-application

[5] Store.arduino.cc. (2018). *Arduino Mega 2560 Rev3*. [online] Available at: https://store.arduino.cc/usa/arduino-mega-2560-rev3 [Accessed 23 Dec. 2018].

[5] RS-555 12V 6100 RPM Brushed DC Motor, Retrieved from https://www.robotshop.com/ ca/en/rs-555-12v-6100-rpm-brushed-dc-motor.html

[6] Encoder Guide, Retrieved from https://www.anaheimautomation.com/manuals/forms/encoder-guide.php

[7] Robot Tracing Strong Magnetic Motor Car RT-4 , Retrieved from ttps://www.aliexpress.com/item/Smart-Car-Chassis-2wd-Robot-Tracing-Strong-Magnetic-Motor-Car-RT-4-Avoidance-Car-with-Code/32809659306.html

[8] Differential drive with continuous rotation servos and Arduino, Retrieved from https://42bots.com/tutorials/differential-steering-with-continuous-rotation-servos-and-arduino/

[9] Arduino DC Motor Control Tutorial – L298N | PWM | H-Bridge, Retrieved from https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

[10] L293D Dual H- Bridge Motor Driver IC, Retrieved from https://kitskart.com/product/l293d-motor-drive-dip-ic/

[11] AKK KC03. 800TVL FPV Camera with 600MW Transmitter (n.d.). Retrieved from https://www.akktek.com/akk-kc03.html

[12] Raspberry Pi Camera. (n.d.). Retrieved from https://www.dexterindustries.com/shop/raspberry-pi-camera

[13] Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YK0F. (n.d.). Retrieved from https://www.sparkfun.com/products/8958

[14] Ultrasonic Sensor HC-SR04. (n.d.). Retrieved from https://www.lazada.com.ph/products/ultrasonic-sensor-hc-sr04-i102973276-s103282735.html

[15] xBee Kablosuz Haberleşme Geliştirme Kartı. Retrieved from https://www.robotistan.com/xbee-24-ghz-1-mw-wire-antenna-xb24-dmwit-250

[16] AKK 5.8G RC832 Mini FPV Receiver Double-Screen Display for FPV Quadcopter Drone. Retrieved from https://www.amazon.com/gp/product/B01FXFZ0NS/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1