MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Potato Integrated Technologies

Final 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 ... 3
2 Introduction ... 4
3 Overall System Description ... 5
3.1 General System Description ... 5
3.2 Individual System Description ... 5
3.2.1 Power Supply Subsystem ... 5
3.2.2 Communication and Telecontroller Subsystem ... 6
3.2.3 Shooting Subsystem ... 11
3.2.4 Motion Subsystem ... 13
3.2.5 Detection Subsystem ... 17
3.2.6 Main Processor Subsystem ... 18
3.3 System Level Block Diagram & Flowchart ... 21
4 Results and Analysis of Performance Tests ... 22
4.1 Power Supply Subsystem ... 22
4.2 Telecommunications Subsystem ... 23
4.3 Shooting Subsystem ... 24
4.4 Motion Subsystem ... 24
4.5 Detection Subsystem ... 25
4.6 Main Processor Subsystem ... 25
4.7 Overall Tests and Results ... 26
5 Deliverables ... 27
6 Budget ... 29
6.1 Cost Breakdown ... 30
6.2 Additional costs ... 31
7 Discussions ... 31
7.1 Safety ... 31
7.2 Applications ... 32
7.3 Environmental Effects ... 32
8 Conclusion ... 32
9 References ... 34
Appendices ... 35
Appendix I ... 35
Appendix II ... 39
Appendix III ... 40

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.

Mission of our company is to provide creative and reliable solutions that fulfill the needs of industry in the field of Industrial robot applications while or vision is to become the most compelling technology company by driving the Industry's transition to smart manufacturing.

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 programming skills. Mr. Beyenir's knowledge in electronics will help us in the integration 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, 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. Also, this robot should be controlled without actually monitoring the play-field with naked eye; the only means of monitoring the field is by means of a camera mounted onboard the robot. Wi-Fi connection is not allowed for transmission, that is the reason why we are using RF communication to transmit the data. In addition to that, the robot is not allowed to cross the center-line into opponent's half-field. Another requirement is that the ball must be transferred to opponent's half-field in no more than 20 seconds. Robots can hit, push or otherwise drive the ball but not grasp, scoop or otherwise carry it.

In order to fulfill these requirements, 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 the final form of this project is presented. Description of overall system and subsystems is presented in this report. In addition to that, drawings, block diagrams, flow charts, test results, cost analysis, time plan can be found in following sections of this report. Also, deliverables, user manual, information about budget and discussions can be found in the following sections.

3 Overall System Description

3.1 General System Description

In this project, we created a robot which is controlled by a remote user using a telecontroller. The robot is able to play hockey game within the standarts declared in last semester. Our robot contains different subsystems corresponding their functions, power supply, telecommunication, shooting, motion and detection.

3.2 Individual System Description

3.2.1 Power Supply Subsystem

This subsystem is for distribute power towards other subsystems at required voltage level and with specified current limitations. The power subsystem is composed of 2 LiPo batteries. As is known, our project consists of 2 separate parts which are controller side and robot side. Controller has “3s 900Mah LiPo” and the robot has “3s 2900Mah LiPo”. Both sides uses LiPo for supplying power to Arduino’s and other subsystems’ components. In Figure 3.1., the one Li-Po cell can be seen.

Figure 3.1 Example of 3 cells Li-Po Pack [1]

The motor draws a much higher current at maximum 2A. Connecting directly with Arduino will result in not working motor and destroying Arduino due to high currents. Hence, we used L298N Motor Drivers for supplying this current with Li-Po cells. The models can be seen in Figure 3.2.

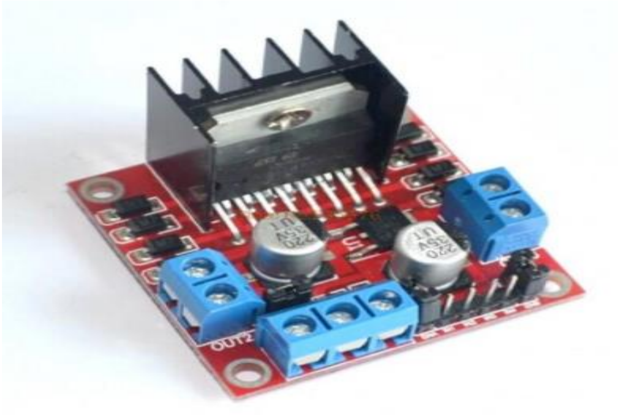

Figure 3.2 L298n dual motor driver [2]

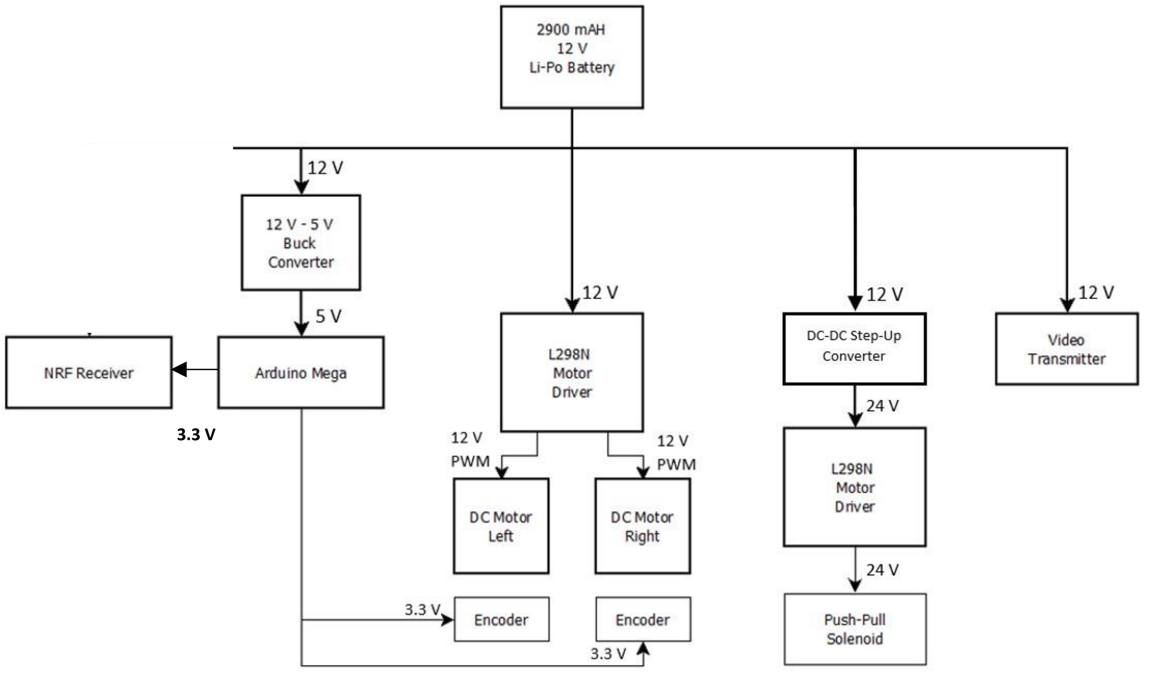

Figure 3.3 Block Diagram of Power Supply System

3.2.2 Communication and Telecontroller Subsystem

3.2.2.1 Communication Link for Sending Commands

As a company, we saw 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.

3.2.2.1.1 The Algorithm

In this communication link, we utilized an RF tranciever module, NRF24L01 and Arduino development boards for both the transmitter and receiver. In the controller side, the command from the user is taken with a joystick. Then the joystick position is read by an

Arduino Uno. Then using SPI serial interface, the generated command signal is sent the RF module and finally the RF signal is sent the antenna. At the receiver side, the robot, the same NRF24L01 module receives the RF signal and again using SPI serial interface, the received signal is sent to the Arduino Mega. Then, the received signal is interpreted via Arduino and the respective command is decided. Afterward, the command is executed. . For better understanding, a simplified functional block diagram of the communication link for sending commands is given in Figure 3.4.

We used a 8 bit register to describe different commands. Whenever an assigned bit is 1, the corresponing command is sent via NRF module as 8 bit register. Bit sequence can be seen in Table 3-1.

Table 3-1 Bit sequences for each individual command

Shoot	00010000
Forward	00001000
Backwards	00000100
Right	00000010
Left	00000001
Slow-Right	00100000
Slow-left	01000000
Slow-Forward	00001001
Slow-Backwards	00000101
Stop	00000000

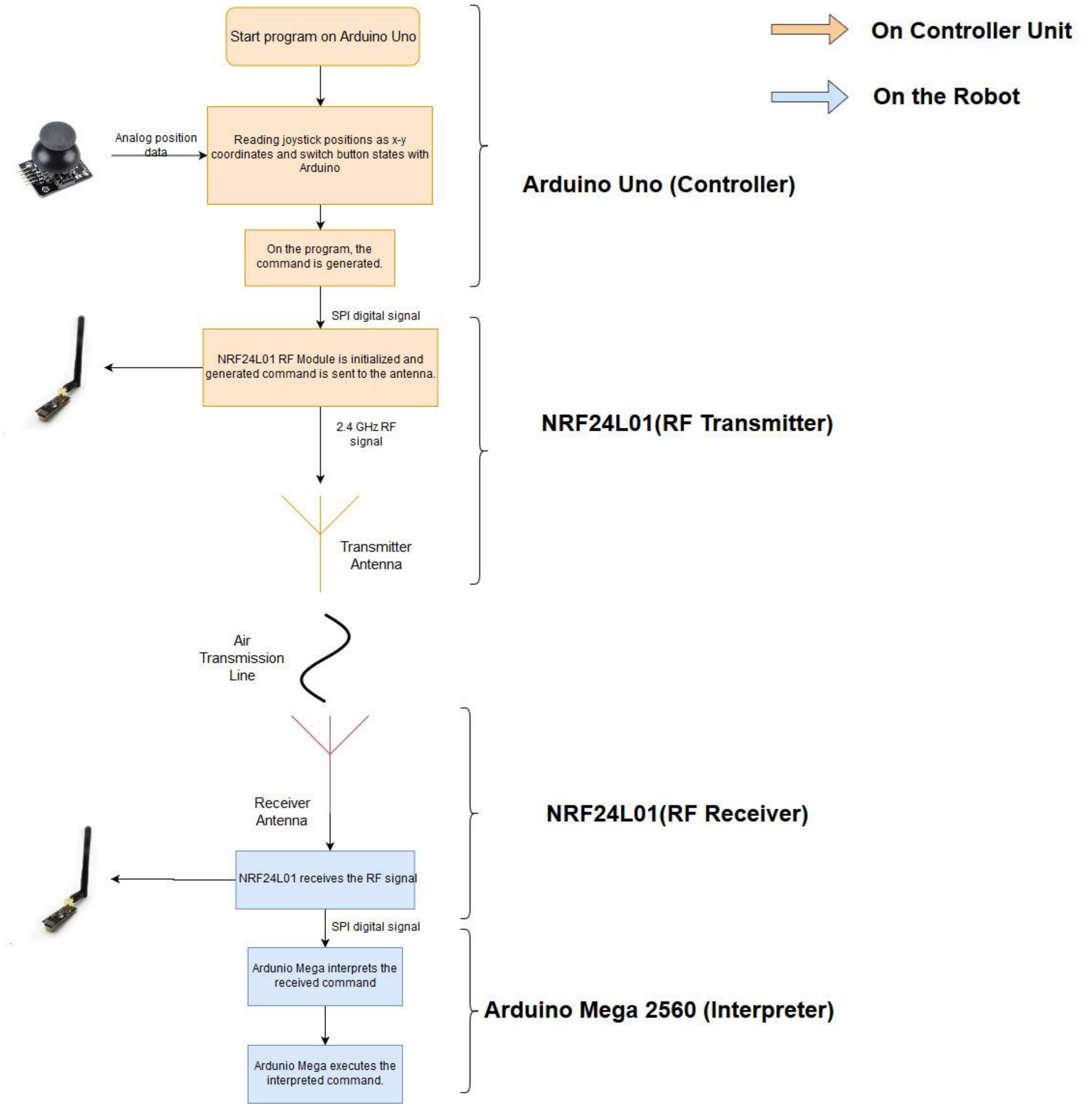

Figure 3.4 Block Diagram of the telecommunication subsystem for sending commands

3.2.2.1.2 Joystick (User Interface)

To receive commands from the user, we implemented a joystick which is composed of two potetiometers and a push button. Depending on the position of the joystick nipple, an analog signal is generated from the potentiometer and then this analog signal is received at the Arduino Uno. This part completes the user interface of the controller subsytyem. A photo of the joystick can be seen in Figure 3.5..

Figure 3.5 The joystick

3.2.2.1.3 Arduino uno (Controller)

On the Arduino, depending the analog signal values from the joystick, the bit sequence is generated (explained at the algorithm part). The program on the Arduino controls the RF tranciever module, NRF24L01, and writes the bit sequence values to the module. We used RF24.h and nrf24l01.h libraries to control the module.[3]

3.2.2.1.4 NRF24L01 (RF Tranciever)

To send the bit seqeunces that carries the command information, we used NRF24L01 module which uses 2.4 GHz RF signal to communicate (Figure 3.6). This module can be used as both transmitter and receiver. On the controller side, the module is used as only transmitter and on the robot side, the module used only as receiver. The selection between transmitter and receiver modes are declared in the program code.

The modules come with an dipole antenna. However, the antenna can be replaced.

Figure 3.6 NRF24L01 RF Module

After sending and receiving the message from the NRF24L01, the message is transmitted to the robot and ready to be interpreted at the Arduino Mega which is located at the robot.

3.2.2.1.5 Arduino Mega 2560 (Interpreter)

After the signal is received by the NRF24L01 module, the signal with command infformation is sent to the Arduino Mega, where the signal is interpreted and the command is decided. In the code, the received bit sequences are compared and matched to the commands. Basically, this is the exact reverse operation of the controller side.

After successfully interpreting the received command, Arduino Mega controls the motion and shooting subsystems by sending respective PWM control signals to the motor drivers.

3.2.2.2 Communication Link for Sending Video

To send the field images to the remote controller, we used a 5.8 GHz FPV drone kit. The system uses NTSC/pal video format which helps with the flow of the stream. With transmitter on the robot and receiver on the telecontroller, the link is complete. However, due to range issues we needed to implement different antenna which we designed and fabricated. The fabricated antenna for the receiver is given in Figure 3.7.. We made a helical antenna because it is highly directional and it inheretly radiates circularly polarized beam. Since it is highly directional, we implemented it on the receiver-controller isde. Because of this configuration, the operator must direct the beam to the robot at all times.

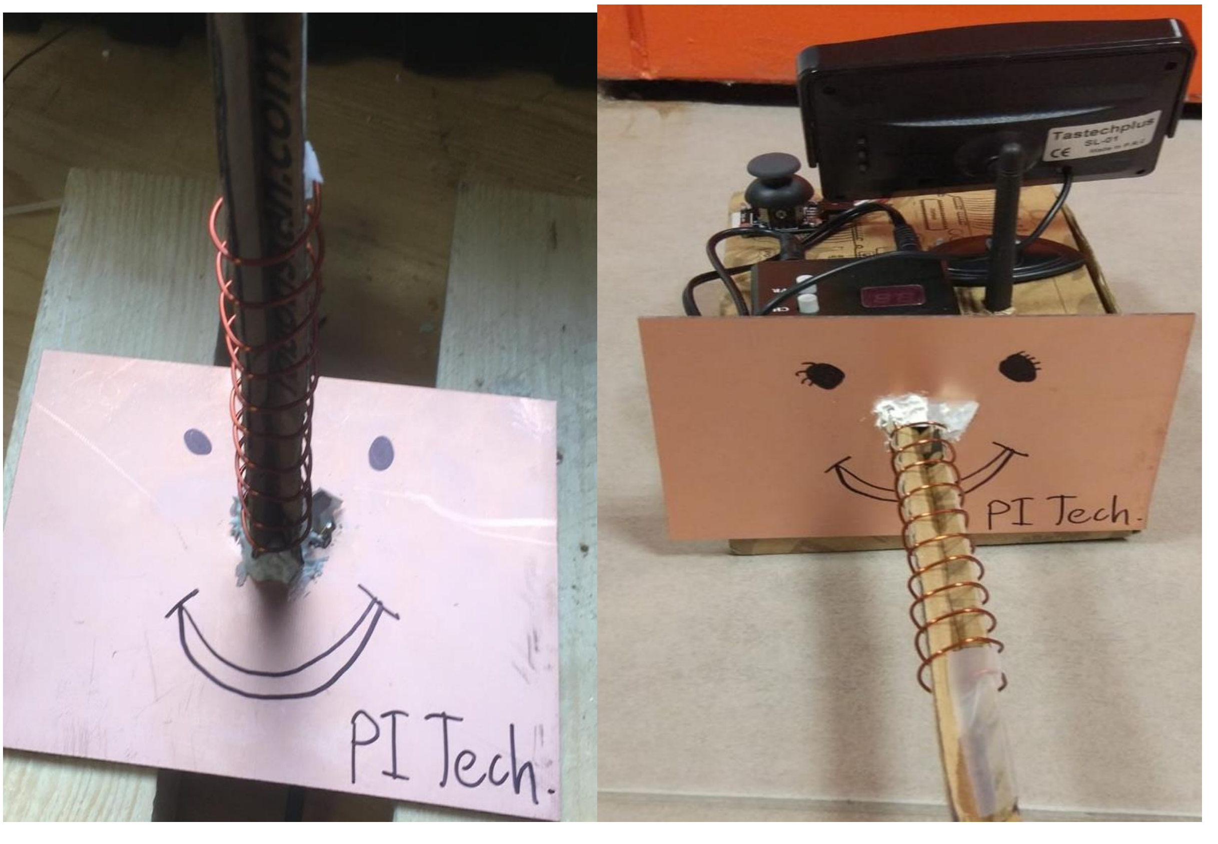

Figure 3.7 Fabricated helical antenna

3.2.3 Shooting Subsystem

The electrical energy stored in the batteries must be transformed in mechanical energy to move the ball. In this system which is solenoid actuated, self-inductance is used. A current is send through a coil which generates a magnetic field. This field can be increased by increasing the number of windings of the coil or by increasing the current through the coil. With this magnetic field, a ferromagnetic material can be attracted or repulsed. The shooting mechanical system design can be seen in Figure 3.8..

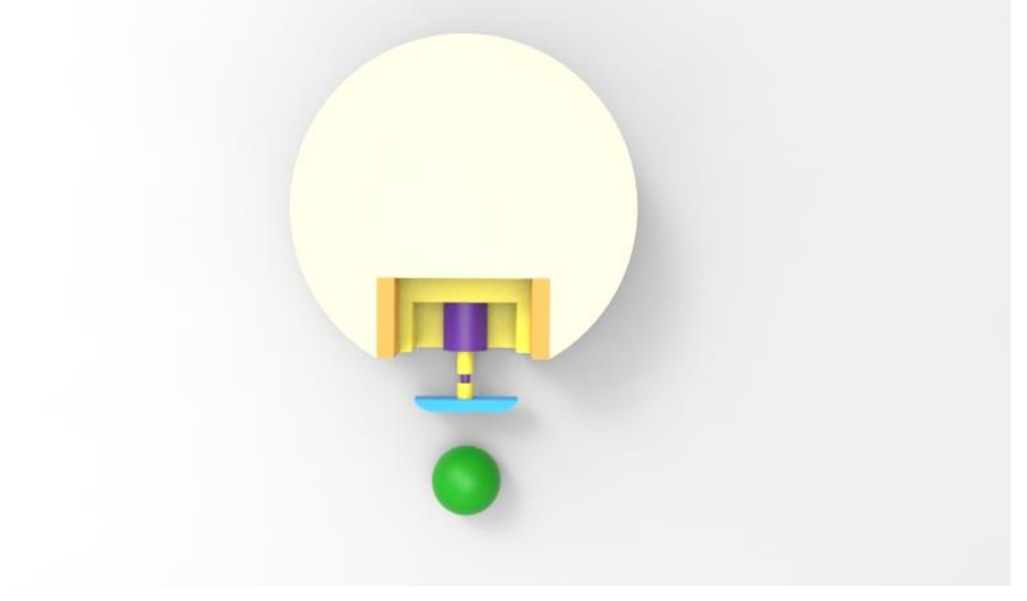

Figure 3.8 Top view of shooting mechanical system design

When a current is sent through a loop of a wire, a magnetic field is built. A solenoid is based on this principle, shown in Figure 3.9. It contains a lot of loops of wire, forming a coil, producing a magnetic field when an electrical current is sent through it. The plunger is used to provide a mechanical force which will be used to kick the ball. The force applied to the plunger by the coil is proportional to the change in current, radius and length of the coil. Also, we used striker plate which increases the surface are of the kicker, increasing the chance of it making contact with the ball[4].

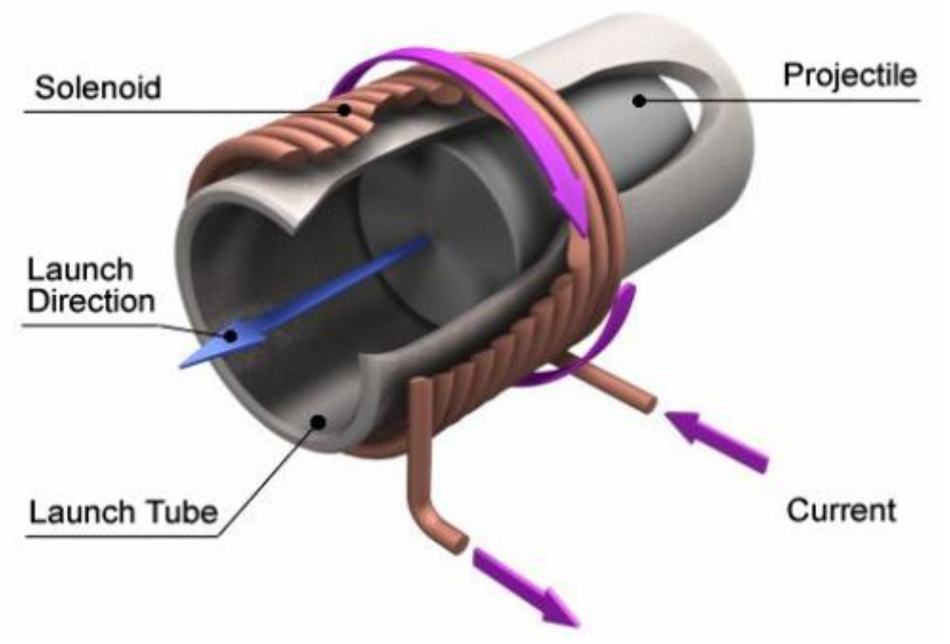

Figure 3.9 A schematic representation of a solenoid [5]

Also, the working principle for how to drive solenoids with Arduino can be seen in Figure 3.10. This schematic is given by supplier with some notes. These are:

1) You will most likely need a heat sink on the transistor.
2) This diagram is for DC solenoids rated up to about 24W: i.e. 12V@2A, 6V@4A etc.
3) The protection diode should preferable be a Schottky type.

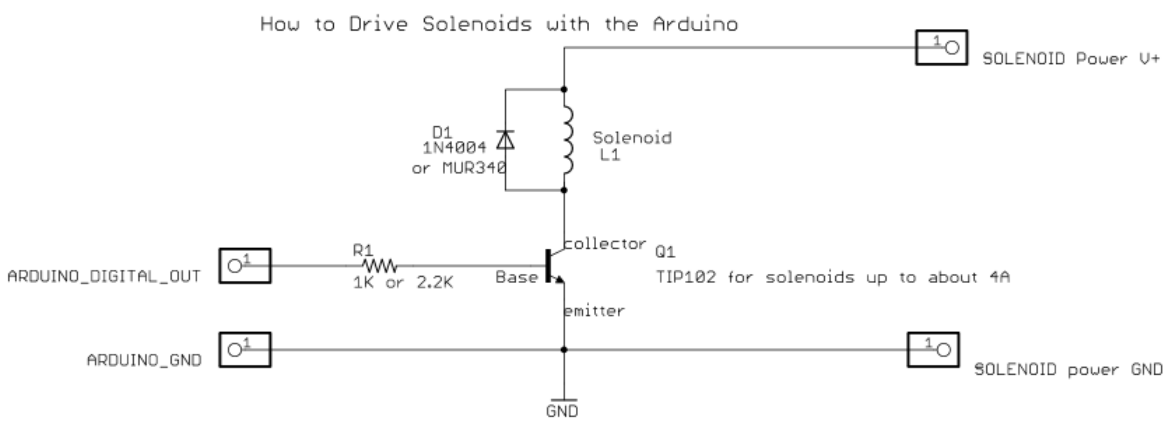

Figure 3.10 The schematic of solenoids with Arduino connections [6]

As can be seen in Block Diagram in Figure 3.11., Dc to Dc converter is used for upgrading DC voltage to hit the ball with more voltages. Also Flow Chart of shooting system can be seen in Figure 3.12.

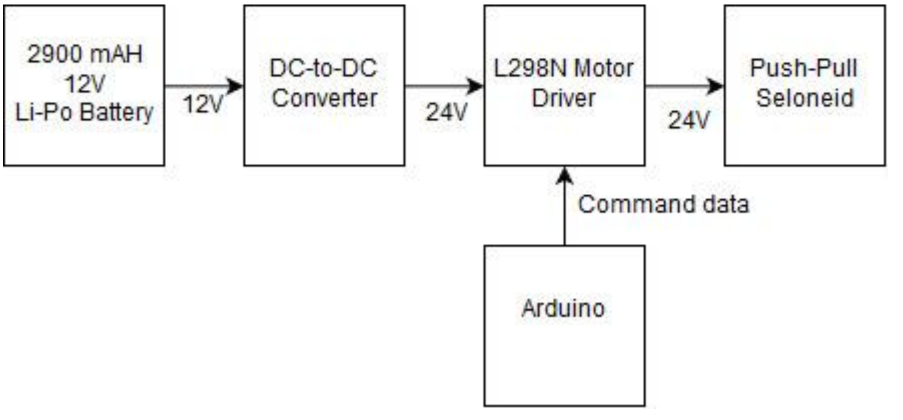

Figure 3.11 Block Diagram of Shooting System

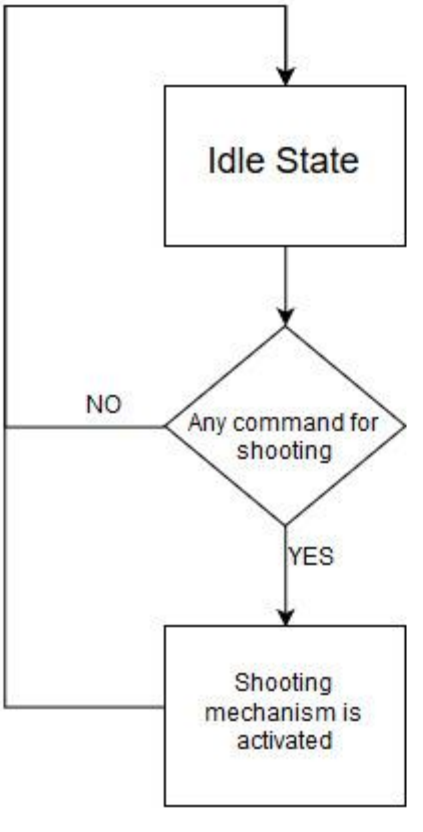

Figure 3.12 Flow Chart of Shooting System

3.2.4 Motion Subsystem

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 converts 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.

We chose to implement our motion subsystem via a DC motor as in Figure 3.13., with the following specifications in Table 3-3:

Table 3-2 DC Motor Specifications

Dimensions	37D x 102L mm
Weight	310 g
Nominal Operating Voltage	12V
Free run speed at 12V	120 rpm
Free run current	400 mA
Stall torque	25 kg.cm
Gearbox ratio	60:1

Figure 3.13 RW-ML-1333 DC Motor

In order to drive these two DC motors, we are using L298N which is our main solution as a motor driver as mentioned in the Conceptual Design Report. 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 two 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 3.14.

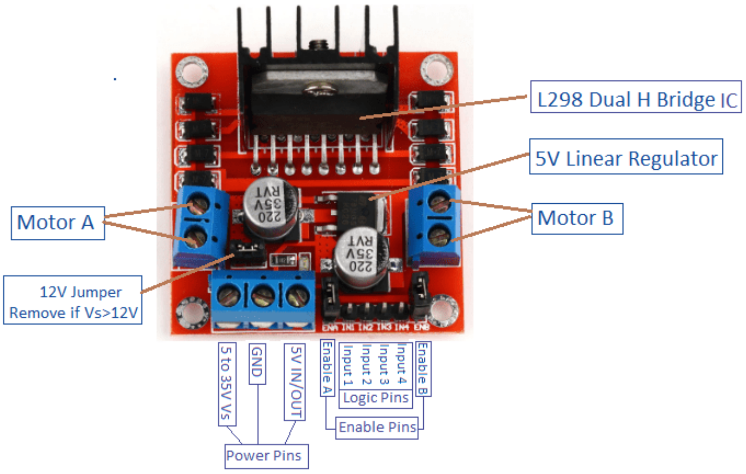

Figure 3.14 Pin Connections of L298N Driver IC

L298N motor driver uses PWM as input to adjust the speed of the motors by changing the duty cycle. 100 % duty-cycle means full throttle and 0 % duty-cycle means zero-velocity. We are using an Arduino Mega on the robot side which creates these PWM signals with respect to the commands sent from the controller side. We are not implementing speed control directly from the controller side. However, we are implementing smooth start and brake strategies in order to prevent high peak current at the start of the motion and also to protect the mechanics of the DC motors. We can also set the turning direction of the motors by simply setting two pins for each motor either HIGH-LOW or LOW-HIGH. These four pins can be seen in Figure 3.13. as the Logic Pins. This direction control is used for differential drive of the robot which was also mentioned in the Conceptual Design process.

After deciding on our DC motor, we needed to choose an encoder to give feedback to the Main Processor Subsystem. This feedback is vital since even though both of the DC motors are identical there may be a calibration difference between them. Our robot may deviate from its straight path. Therefore, we need to calibrate these two motors by implementing optimization with respect to the encoder counting differences.

Incremental encoders are useful for our purpose. The encoder type we are 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 3.15..

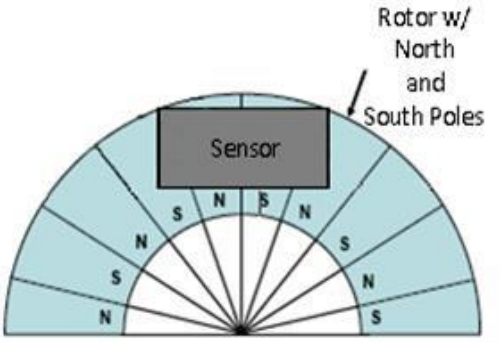

Figure 3.15Magnetic Encoder Representation with Hall-effect Sensor

For wheels we are using two standard wheels and also three ball wheels to stabilize the robot chassis. Wheels-chassis integration is covered in the technical drawings part.

Block diagram of the motion subsystem can be seen in Figure 3.16..

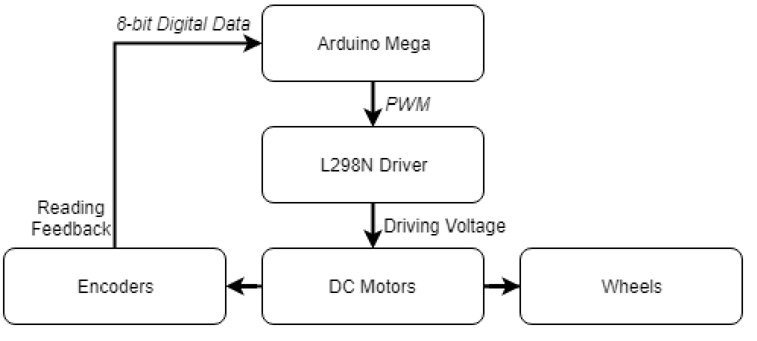

Figure 3.16 Block Diagram of the Motion Subsystem

Flowchart of the motion subsystem can be seen in Figure 3.17..

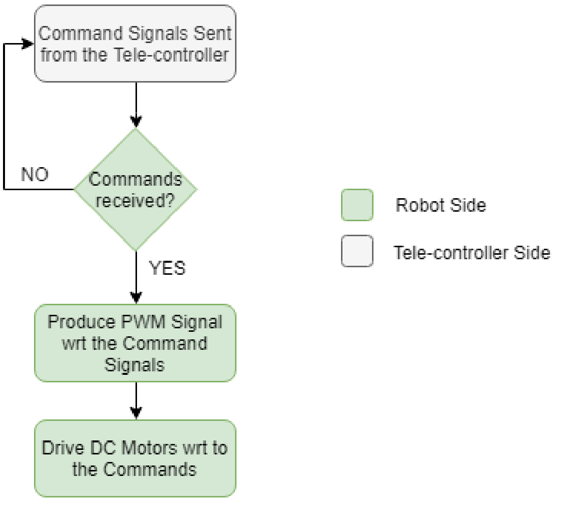

Figure 3.17 Flowchart of the Motion Subsystem

3.2.5 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.

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. The transferred data is streamed on an LCD screen for telecontroller.

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, this system directly sends the raw video data without any processing. The block diagram and flowchart of detection subsystem can be observed in Figure 3.18. and Figure 3.19. respectively.

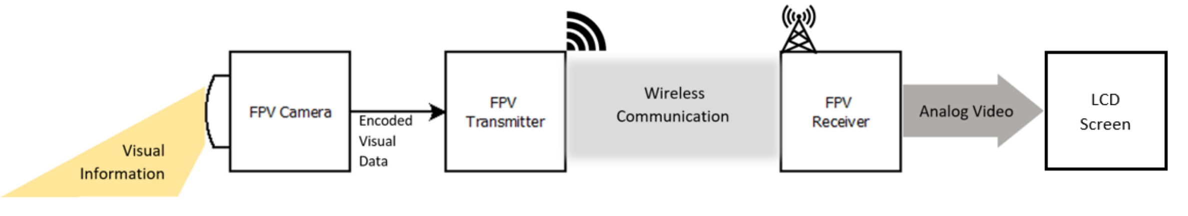

Figure 3.18 Block diagram of Detection Subsystem

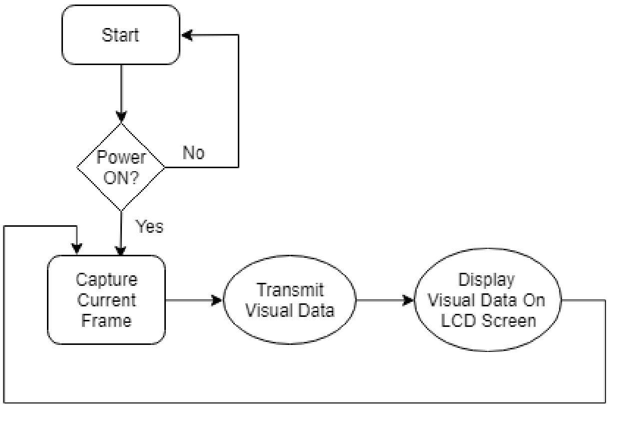

Figure 3.19 Flowchart of Detection Subsystem

As observed in figures provided, in the first step camera captures the real time visual information about the playfield. This visual information is encoded for the transmission and send directly to the receiver via RF channel. The transmitted information does not have any processing on it, due to the timing issues. The received real time video data is decoded and converted to the AV (Analog Video) format which is supported by our screen.

For the data transfer we use AKK KC03 800TVL NTSC Switchable Camera Module, shown in Figure 3.20., 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 3.20 AKK KC03 800TVL NTSC Switchable Camera Module 600mW FPV Transmitter

3.2.6 Main Processor Subsystem

Our use for main processor subsystem is to use Arduino Mega. Figure 3.21 below shows an Arduino Mega.

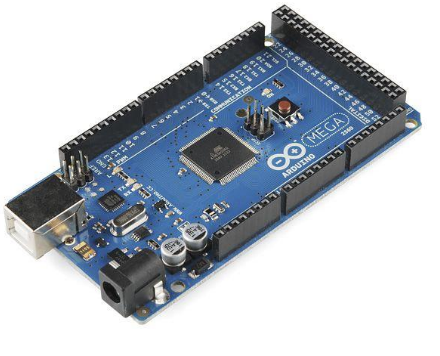

Figure 3.21 Arduino Mega

We used Arduino Mega for main processor subsystem it is chosen because it has variety of I/O pins which makes it possible for us to use only one processor for robot side. Also, Arduino has a variety of online sources that are easily reachable which makes it easier to program. Arduino MEGA will used to control motors, shooting system motor and command receiver module on robot side.

We used an Arduino UNO in the tele controller side as a processor. The reason why we use UNO instead of MEGA in tele controller part is that we do not need higher DC current or higher number of pins that MEGA offers in tele controller part. Figure 3.22. below shows an Arduino UNO.

Figure 3.22 Arduino UNO

Arduino UNO is used to control the joystick commands, command transmitting module and video receiving module on tele controller side.

Figures 3.23-24 below shows block diagrams for main processor subsystem. Signal interfaces are clearly labeled.

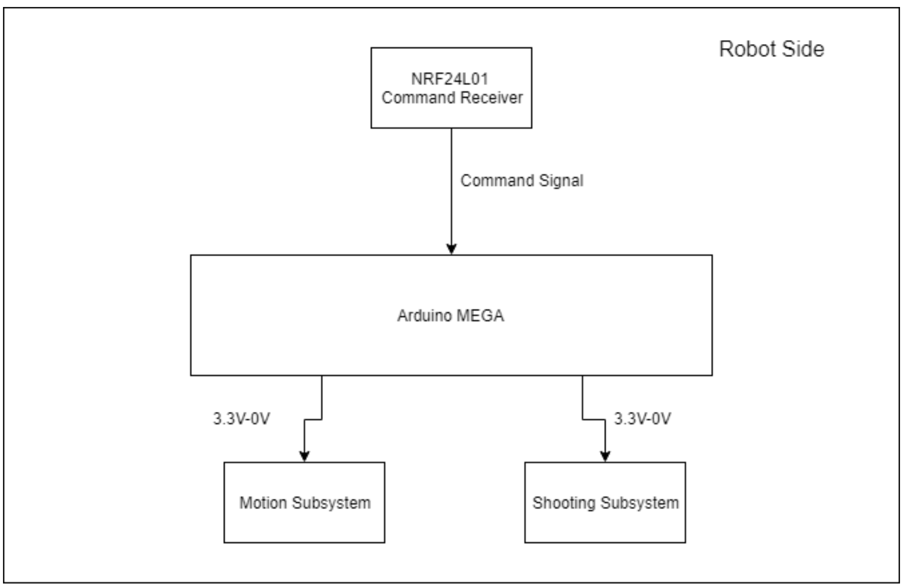

Figure 3.23 Block diagram of processor system at robot side

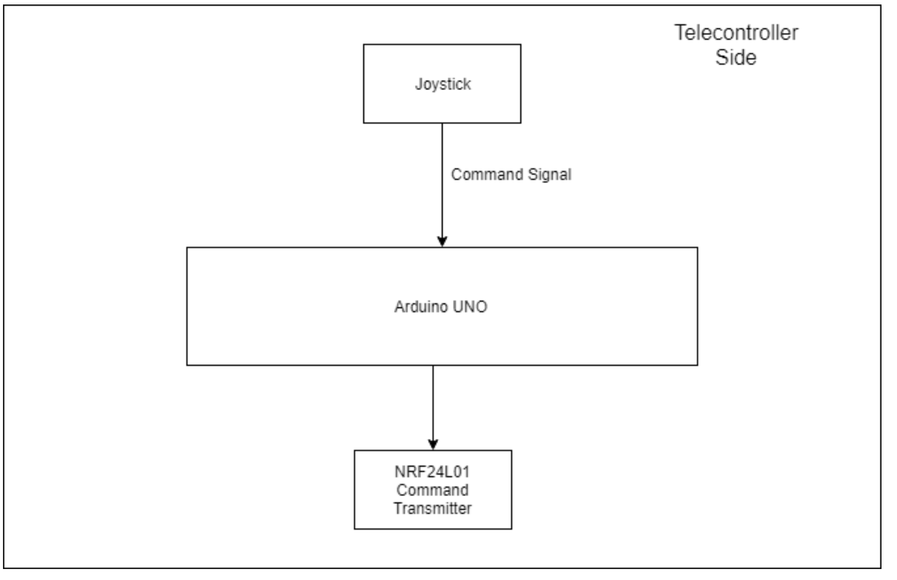

Figure 3.24 Block diagram of processor system on telecontroller side

Figure 3.25 below shows a flow chart for both of the processor system.

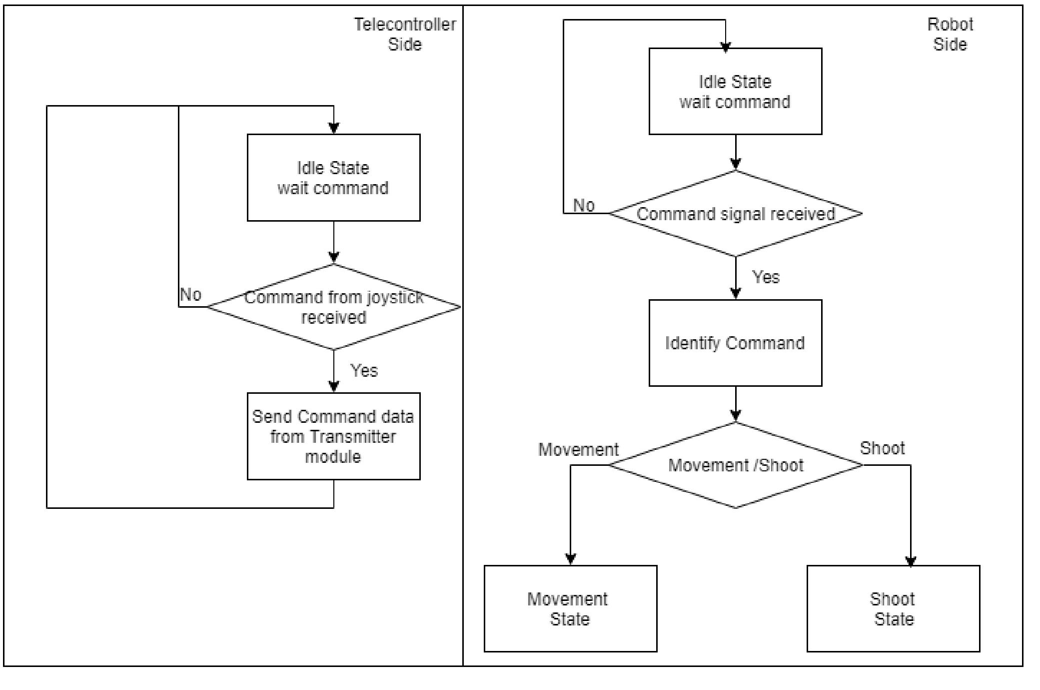

Figure 3.25 Flow chart for main processor subsystem

3.3 System Level Block Diagram & Flowchart

Block diagram of the overall system can be divided into two parts. First one is the Controller Side and the second one is Robot Side. Block diagrams of these two parts can be seen in Figure 3.26 and Figure 3.27.

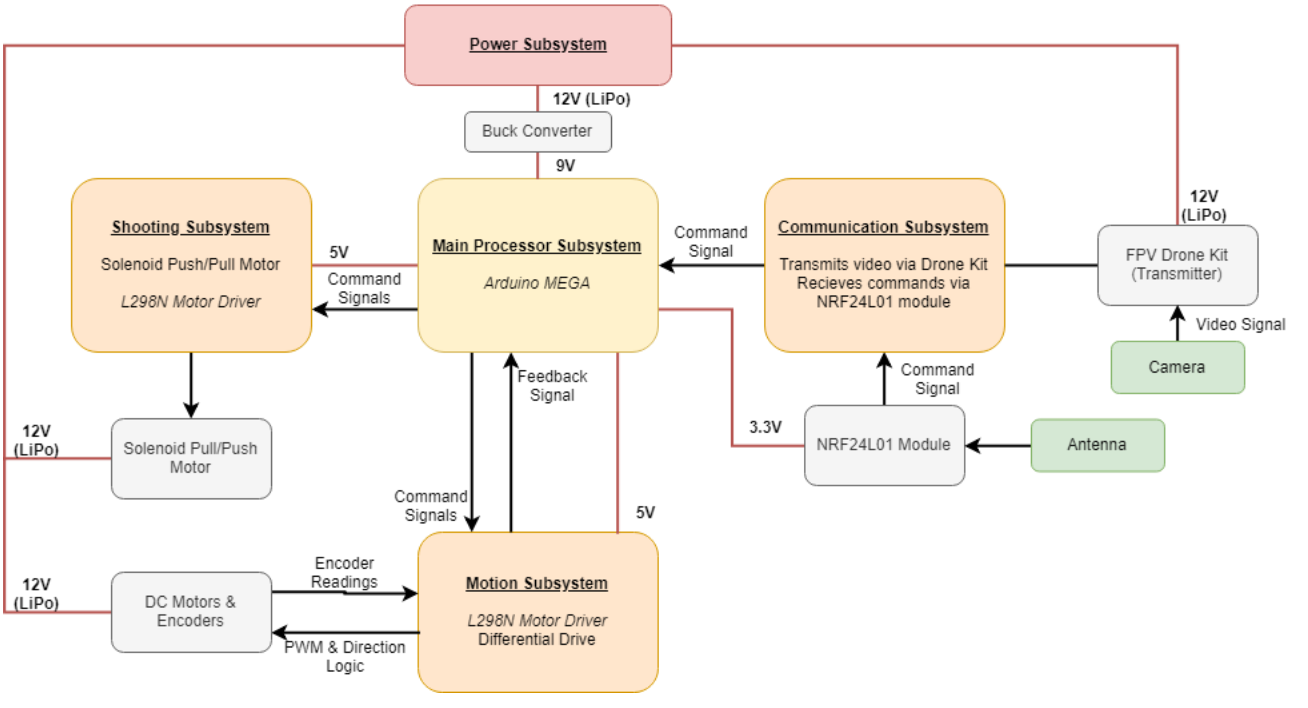

Figure 3.26 Block Diagram of the Overall System (Robot Side)

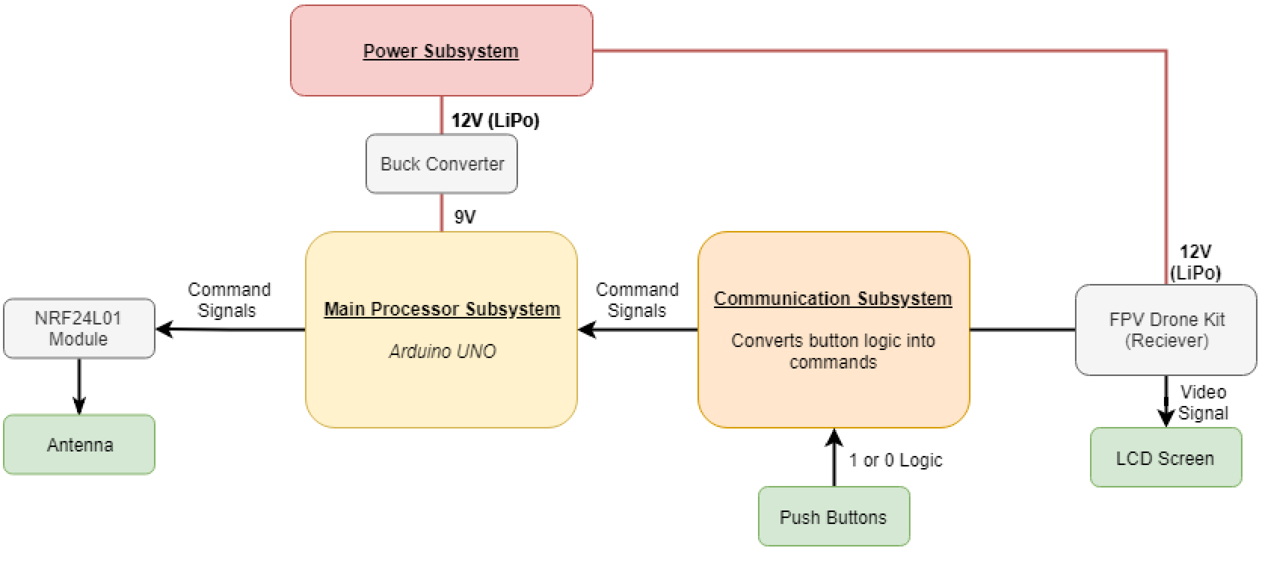

Figure 3.27 Block Diagram of the Overall System (Controller Side)

Flowchart of the overall system can be seen in Figure 3.28..

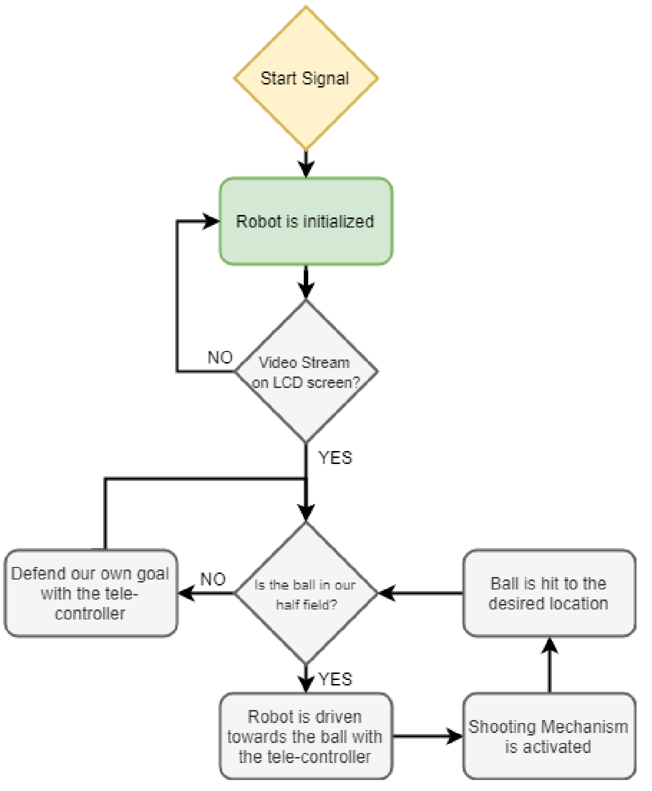

Figure 3.28 Flowchart of the Overall System

4 Results and Analysis of Performance Tests

4.1 Power Supply Subsystem

To comply with the power requirements power tests are conducted. First test was to measure current values under full load conditions. We measured the robot side draws 3.4 Amperes under full load conditions (going forward with full speed while transmitting video). Through related calculations this corresponds approximately 23 minutes of operations with our 1350 mAh Li-Po battery. Even though this result complies with the requirement that has been mentioned in the earlier reports we want operation longer that 23 minutes. On the other hand, this calculation made under the assumption of the system always runs under full load conditions which is not the case under real life conditions. Since our robot is not going to be move all the time with full speed our expected battery life is more than 81 minutes. Under no load conditions (when our robot is not moving but video transmission is available) our robot draws less than 1.5 Ampere. Under no load condition this result gives us 54 minutes of operation. So, if we assume that our robot works under full load conditions 50% of the time we will have approximately 38 minutes of operation on the robot side. To conclude if we assume that one match will take approximately between 10 to 15 minutes, our robot can play 2 to 3 games without the need of recharging.

This test was also made with demonstration. We played a game with an opponent for 10 minutes of time after we fully charged our battery. After 10 minutes, we checked our battery with our Li-Po battery charger and observed that we only used 30% of our battery. This result verified our prior calculations.

The same tests are also conducted for teleoperator side. Teleoperator side has 900 mAh battery. Teleoperator draws approximately 1.5 Amperes. Teleoperator's power rating does not change significantly so this value used for our calculations. With this current rating our 900 mAh Li-Po battery gives us 36 minutes of operation. When we conduct the test with demonstration after 10 minutes of operation we observed that also 30% of teleoperator's battery has been used.

To analyze these results, we can say that we offer our customer approximately 35 minutes of operation without the need of recharging. This value can be increased with the use of higher capacity batteries. However, since batteries are important costs of the robot it was a tradeoff. Customer can buy higher capacity batteries if they need to. Since, we assumed it will be sufficient to play 2 to 3 games for an average customer we used these batteries. Also, since the capacity is smaller they charge faster. So, at the end these battery values were most suitable for our case.

4.2 Telecommunications Subsystem

For our final design, we followed following test procedure for command sending subsystem: performing command sending quality with certain ranges. We measure data loss per 100 commands. Simply comparing the values we obtained bit error rate. Obtained results are given in Table 7-1.

Table 4-1 Bit error rate versus distance between modules

Distance (m)	7	14	21	28	35	42
Data Loss (bits)	0%	0%	0%	9%	15%	22%

In the final design, the command sending systems works without any problem. However, video sending system needs careful alignment from the controller side because of the directive helical antenna. Any deviation from the radiation pattern can result in loss of video stream.

4.3 Shooting Subsystem

We created a shooting mechanism that can hit the ball up to 150cm, which is the full length of the field. Our shooting mechanism is a system with a solenoid trigger. Therefore, we have experimented at 10V, 12V, 16V, 18V, 20V and 24V to obtain the desired power to hit the ball. Since we reached the desired result at 20V, we didn't need to use 24V of supply. Although it would give better performance, the power it consumes may damage the solenoid at this voltage. Therefore, we used 20V of supply voltage. Desired result is 150 cm, the distance between two farthest points of the playfield.

Table 4-2: Throw distance versus supplied voltages of solenoid

Supplied voltage (V)	10	12	16	18	20	24
Distance (m)	0.00	0.20	0.40	0.80	1.5	2

4.4 Motion Subsystem

We optimized our design considering the 20 sec restriction and also the amount of time we can play with a fully charged LiPo battery. Initially we tried increasing the speed incrementally. After some trials within the playfield we saw that it takes a lot of time to drive the robot to a desired location with this adjustment. Therefore, we decided to increase the number of commands rather than increasing the speed incrementally. Since we have a joystick on the controller side, we increased our direction commands to eight. There is one middle speed and also full speed value for each direction. This adjustment enabled us to perform small operations. As we mentioned earlier, the velocity of our robot is approximately 45 cm/sec at its full speed which is highly enough to comply with 20 sec restriction.

We also checked our final design after implementing the above-mentioned adjustment to see the stability of the robot during a real game. Since we have middle speed values, stability during small operations is highly increased. In addition to stability we also observed the heat generated by the motors after one full operation. Since we are not forcing DC motors too much, they don’t even get hot therefore, the mechanic strain on the DC motors can be neglected.

4.5 Detection Subsystem

The subsystem test of detection subsystem is designed to detect any possible malfunction on the real time video stream.

Screen test: The LCD screen must stream all incoming analog video information. To check functionality of the LCD screen, we connect external AV video input to the screen. The screen must show all frames of the input video without any pixel error.

Video stream test: This test is designed for controlling the functionality of camera. The test will be performed with all elements of subsystem are activated in a controlled environment (no other active video module around), with transmitter and receiver modules located 15 cm apart. During the test quality and delay of the video stream is measured independent from environmental effects.

Communication channel test: The main aim of this test is controlling the communication channel we are using. Our communication channel can communicate on different channels, which gave us great flexibility if any other signals are present. For this test we only activate receiver and LCD screen. Without any broadcast from our transmitter, we only listen other channels and test the availability of the channels. The occupied channels need to be avoided since they decrease the quality of our signal and visual data. The channels with minimum noise must be chosen at the end of this test.

Range test: After all the previously tests are done, the last test must be performed. In the range test, the camera and transmitter are slowly driven apart from receiver and screen, until the video stream on screen become fuzzy. If the video protects its quality more than 30 meters indoor, this test is successful.

When the all above tests are conducted, it is seen that detection subsystem works without any problem and passes all of the tests.

4.6 Main Processor Subsystem

Main processor subsystem only consists Arduino's so for this subsystems test procedure the only thing that should be tested is the operations of Arduino's.

Power Test: Power test will be conducted under different voltage supplies. At the end we won't supply Arduino's from computer or powerbanks so Buck converter output should be connected to Arduino to test the operations. If it is turning on this test will be successful. We tried it with power supply and it turned on, so this test resulted successfully.

Simultaneously Parallel Operation Test: Since we want the work different operations in parallel, a test should be conducted while shooting, motion and receiver modules are connected. All of them worked successfully during the test, so result is successful.

4.7 Overall Tests and Results

After all the submodule tests are conducted and desired results are obtained, we conducted some series of tests to determine overall performance of the system as a whole. These tests are suited to measure the compatibility, synchronization and physical aspects of the module.

To observe all effects and errors we conducted these tests with complete demo setup. Teleoperator is located with a distance slightly more than 30-meter indoor with the robot. Robot is located at demo ground, METU EEE Block E top floor. Moreover, all these tests are conducted 3 times with 3 different teleoperator to minimize the effect of the experience level of operator.

First test was path following test. This test is designed to measure maneuverability of the robot and user friendliness of controller mechanism. To perform this test, a test path is drawn to the ground using red tape. Path contains both straight lines and sharp angle turns. From the teleoperator it is expected to complete the path with minimum time without going off-track. For success of this test we expect to complete the test under 3:30 minutes without going off-track. After 3 test runs all operators succeed to follow the path with 2 minute 26 second average time. Also, all the times are under 3:30 minutes. The use of the robot was simple and response time and speed was satisfactory in this test.

Second test was defense test. This test is designed to measure responsivity of the robot and its ability to defend its goal. To perform this test, half field is built and from 75cm distance ball are shoot to the goal. Total 10 shoot with various speed is sent to the goal for each teleoperator. For the success of this test at least 5 of the balls need to be saved. In the tests 5.33 of the balls are saved by the robot and 2 of 3 teleoperator performed above success criteria. Despite being above the success criteria, these results showed us that for defense, experience level of teleoperator is crucial.

Third test was offense test. This test is designed to measure ball handling ability of the robot and precision of the shoots. To perform this test, complete field is built and at random time intervals a ball is randomly placed into the operator's half field. Total 10 balls are placed for each teleoperator. For the success of this test at least 7 of the balls should be sent to opponent's half field and at least 3 of them go into the opponent's goal. Also, if the ball stays at players half field more than 20 seconds, or player crosses the half field a penalty point is given to the player. In the tests, mean of the successfully sent shoots to opponent's half field is 8.66. Average 4 of the sent shoots were gone into opponents' goal. In this test only two player has a one penalty point. The over the top location of the camera gives teleoperator to determine approximate location of the robot with wide angle view. This configuration also makes finding and controlling the ball simpler. Because of that we can say that independent from the experience level of teleoperator, attack ability of this module is simple and effective.

5 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.

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 unit; the robot unit which will take a part in a competition where it tries to score to the opponents' goal protect its goal. Our final product will be send fully attached only thing that the customer should do is to connect batteries to the main system. Figure 5.1 below shows the main unit that is going to be send to the customer.

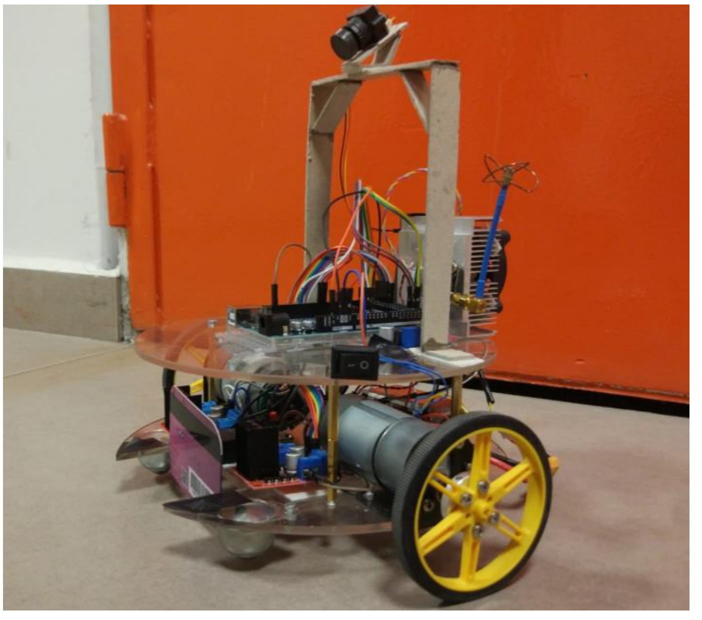

Figure 5.1: Main unit

Teleoperator: A controller unit which the operator going to use for teleoperate the robot unit is going to be delivered to the customer. This part will have a screen mounted on it in order to display the robot's environment. A joystick will be mounted on this module to control the robot's movements. This part will also be send fully attached except the battery. Figure 5.2 below shows the teleoperator.

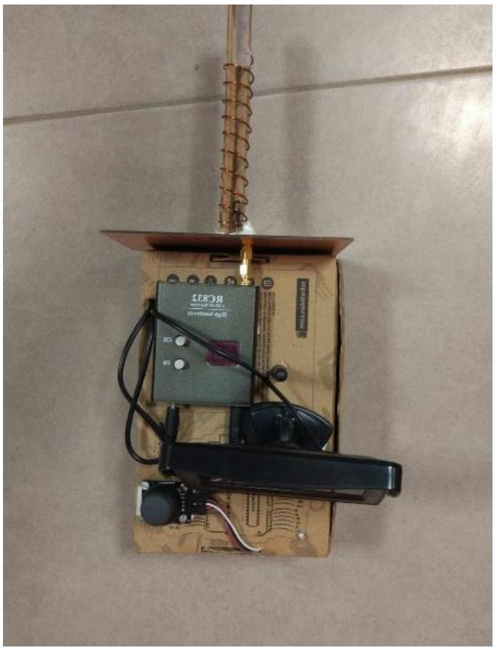

Figure 5.2: Teleoperator

Auxiliary Equipment: This package contains equipment necessary to fully operate the robot and build the competition environment. The pack contains; 2 Li-Po battery, 1 for teleoperator and the other for main unit, 6 Walls for game field that have tapes indicating the self-goal position, two balls (diameter 30-45 mm) (one is spare), a dummy robot (for demonstration purposes).

Documents: This package contains the all the necessary documents in hardcopy. This pack contains; one user manual (can also be found Appendix II), 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 purpose of computer interface is to make necessary adjustments on the robot for different environmental conditions. The user might change the speed of the robot using this software if it is necessary. Also, digital versions of the documents can be found on the software.

6 Budget

The financial expenses on the development process of this module can be examined in two parts, cost of module and additional costs. The cost of module shows the cost of building another module with current parts. On the other hand, additional costs shows the cost of the development process of this module from scratch. This expenditure is non-recurring and totally dependent of the learning process of the team.

6.1 Cost Breakdown

The total expenditure for building the robot is provided in Table 6.1. cost breakdown table.

Table 6.3: Cost breakdown table

Materials	*Quantity*	*Price per each*	*Total Price*
Motors			
DC Motors	2	$10	$20
Gearbox	2	$5	$10
Encoders	2	$5	$10
L298N Motor Driver	2	$1.5	$3
Development Boards			
Clone Arduino Uno	1	$4	$4
Clone Arduino Mega	1	$10	$10
Shooting			
Push-Pull Solenoid	1	$4	$4
Communication			
FPV Drone Kit	1	$35	$35
NRF24L01+PA+LNA SMA	2	$4	$8
Telecontroller			
Joystick	1	$0.50	$0.50
LCD screen	1	$30	$30
Power			
1000 mAH Li-Po Battery	1	$14	$14
1350 mAH Li-Po Battery	1	$18	$18
Buck Converter	3	$1	$3
Structural			
Plexiglass Chassis	1	$6	$6
Standard Wheels	2	$2	$4
Ball Wheel	3	$1	$3
Cables & Connectors	N/A	N/A	$6
3D Printed Parts	N/A	N/A	$2
Structural connections	N/A	N/A	$3
Demo Setup			
Dummy Robot	1	$2	$2
Balls	2	$0.25	$0.5
Play Field Walls	3	$1	$3
Total:			**$196**

6.2 Additional costs

Additional costs cover the expenses made on the development process of the module. Contains but not limited with cost of necessary tools, software, not used parts due to design changes, engineering cost, etc. These costs are non-recurring costs and after the first product these costs will not repeat.

Table 6.4: Additional costs

Materials	*Quantity*	*Price per each*	*Total Price*
Electronics			
Raspberry Pi 3(Unused)	1	$35	$35
Raspi Camera (Unused)	1	$30	$30
Radio (Unused)	1	$5	$5
Extra step-up converters	2	$1	$2
Resistors and Capacitors	N/A	N/A	$3
Tools			
Solder Iron and Utilities	N/A	$4	$4
Various Tapes	N/A	N/A	$3
Multimeter	1	$10	$10
Scissors and Cutters	N/A	N/A	$5
Tools	N/A	N/A	$20
Engineering			
Non-Recurring Engineering Cost	200 hr	$0.25	$50
Total Additional Costs:			**$167**
Total Expenses:			**$363**

7 Discussions

7.1 Safety

We know that many robot accidents do not occur under normal operating conditions but rather during programming, adjustment, testing, cleaning, inspection, and repair periods. During many of these operations, the operator, programmer or corrective maintenance worker may temporarily be within the robot work envelope while power is available to moveable elements of the robot system. Hence, we had to be careful to avoid receiving electric shock.

7.2 Applications

Our product has 3 important sub-modules. The method used in the shooting system can be used in applications where many kinds of triggers are involved. Image transfer and command transfer modules can also be used in any application in which wireless communication exists. Generally, the product can be sold in the market as a fun product for the entertainment industry.

7.3 Environmental Effects

Li-Po batteries are the main cause of the environmental impacts of our product. Li-ion batteries are criticized because of the potential safety concerns that they pose. In some cases, batteries would release hot gases, which were enough to cause burns.

8 Conclusion

The main focus of this final report is to show that our final product meets the requirements that have been mentioned in previous reports. This report showed the results and performance requirements of our final product to emphasize the fact that the final product meets the customers expectations and needs. In this report we presented our system design 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. We clearly described our design with top-down system approach, showed results and analyses of performance tests.

We clearly justified our design decisions with related calculations and diagrams. Block diagrams and flow charts were given for visualization of the system. Moreover, individual subblock descriptions and their flow charts were presented in this report. General circuit diagram was also given.

Results and analyses of performance tests were presented in this report. These tests showed our final product meets the expectations of the final user. The contents of final product package that is going to be delivered to customer was covered in the Deliverables part.

The budget of the entire project was mentioned in two separate parts namely actual expenditures and total costs. Total costs consist of engineering and infrastructure costs in addition to the cost of final product.

This report also provides discussions on safety issues and precautions that our team take to avoid safety problems, possible widespread applications and potential environmental effects of our final product.

To conclude, this report contains all the parts of our design with clear justification. 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. So, we 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.

As PITech engineers, we tried to find the best solution within $200 budget that will fulfill the requirements and satisfy the customer. This report is to shows that our system under review can proceed into system fabrication, demonstration, and tests. We are confident and showed that our design meets the stated performance requirements within cost (budget), customer needs and other system constraints.

9 References

[1] Figure 3.1 Retrieved from http://www.helipal.com/storm-11-1v-2200mah-35c-pro-series-li-po-battery-xt60.html.

[2] Figure 3.2 Retrieved from tronixlabs.com.au.

[3] NRF24L01 Arduino Libraries. Retrieved from https://github.com/nRF24/RF24

[4] Mechatronic Design of a Soccer Robot for the Small-Size League of RoboCup. Retrieved from http://mech.vub.ac.be/multibody/final_works/thesis_joris_de_witte.pdf

[5] Figure 3.8 Retrieved from http://www.mate.tue.nl/mate/pdfs/6971.pdf.

[6] Figure 3.9 Retrieved from https://pdf.direnc.net/upload/solenoid-driver-adafruit.pdf.

Appendices

Appendix I

Arduino Code

Transmitter Side (Arduino Uno):

```
#include  <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"     //Modül ile ilgili kütüphaneleri ekliyoruz
RF24 verici(9,10);     //kütüphane tarafından kullanılacak pinleri tanımlıyoruz (CE,CSN)
const uint64_t kanal = 0xE8E8F0F0E1LL; //kanalı tanımlıyoruz
int joystick_x_pin = A0;
 int joystick_y_pin = A1;
 int joystick_switch_pin = A2;
 int joystick_x=0;
 int joystick_y=0;
 int joystick_switch = 0;
 int message[1] ;
void setup(void)
{
 Serial.begin(9600);
 verici.begin();       //nrf yi başlatıyoruz
 verici.openWritingPipe(kanal);  //kanal id si tanımlanıyor
}
void loop(void)
{
     //Reading commands from joystick
     joystick_x = analogRead(joystick_x_pin)-512;
     joystick_y = analogRead(joystick_y_pin)-512;
     joystick_switch = analogRead(joystick_switch_pin);
     if(joystick_y>400){
            message[0]=B00000010;   // "sağ" komutu
            }
        else if (joystick_y<-400){
            message[0]=B00000001;   // "sol" komutu
            }
       else if (joystick_x>400){
            message[0] = B00001000; // "ileri" komutu
            }
        else if (joystick_x<-400){
            message[0]=B00000100;   // "geri" komutu
            }
        else if  (joystick_switch <50) {
            message[0]=B00010000;    // "shoot" komutu
            }
        else if (joystick_y<450 && joystick_y>100){
            message[0]=32;       // "yavaş sağ" komutu
            }
        else if (joystick_y<-100 && joystick_y>-450){
            message[0]=64;      // "yavaş sol" komutu
            }
        else if (joystick_x>100 && joystick_x<450){
            message[0] = 9;    // "yavaş ileri" komutu
            }
        else if (joystick_x<-100 && joystick_x>-450){
            message[0]=5;            // "yavaş geri" komutu
            }
        else  {
             message[0]=B00000000;   //"dur" komutu
             }
 verici.write(message, 1);   //mesaj değişkeni yollanıyor
 Serial.print(message[0]);
 }
```

Receiver Side (Arduino Mega):

```
#include <SPI.h>
#include <Servo.h> //Servo kütüphanesini ekledik.
#include "nRF24L01.h"
#include "RF24.h"


#define enA 2
#define enB 3
#define in1 24
#define in2 26
#define in3 28
#define in4 30
#define in5 32
#define in6 34


int mesaj[1];
RF24 alici(9,53); //CE,CSN
const uint64_t kanal = 0xE8E8F0F0E1LL; //kanalı tanımlıyoruz
bool sag = false;
bool sol = false;
bool ileri = false;
bool geri = false;
bool shoot = false;
String karar ;
int i;



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

 alici.begin();
 alici.openReadingPipe(1,kanal);
 alici.startListening();            //gönderilen veri için dinlemeye başlıyor


  pinMode(enB, OUTPUT);
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(in5, OUTPUT);
  pinMode(in6, OUTPUT);


 }

void loop(void){
 if (alici.available())
 {
   bool done = false;

     done = alici.read(mesaj, 1);  //okunan veri mesaj değişkenine yazılıyor

       if (mesaj[0]==5){
         karar = "yavaş geri";
             analogWrite(enA, i);         //pwm değeri girme 1.motor için
             analogWrite(enB, i);         //pwm değeri girme 2.motor için
```

```
digitalWrite(in1, HIGH);      //yön belirleme
             digitalWrite(in2, LOW);
             digitalWrite(in3, HIGH);
             digitalWrite(in4, LOW);
             digitalWrite(in5, LOW);
             digitalWrite(in6, LOW);

        }

     if(mesaj[0]==9){

         karar =  "yavaş ileri";

             analogWrite(enA, i);
             analogWrite(enB, i);
             digitalWrite(in1, LOW);
             digitalWrite(in2, HIGH);
             digitalWrite(in3, LOW);
             digitalWrite(in4, HIGH);
             digitalWrite(in5, LOW);
             digitalWrite(in6, LOW);

        }

        if(mesaj[0]==64){

          karar = "yavaş sol";
           analogWrite(enA, i);
           analogWrite(enB, i);
           digitalWrite(in1, HIGH);
           digitalWrite(in2, LOW);
           digitalWrite(in3, LOW);
           digitalWrite(in4, HIGH);
           digitalWrite(in5, LOW);
           digitalWrite(in6, LOW);

        }

        if(mesaj[0]==32){

          karar = "yavaş sağ";
             analogWrite(enA, i);
             analogWrite(enB, i);
             digitalWrite(in1, LOW);
             digitalWrite(in2, HIGH);
             digitalWrite(in3, HIGH);
             digitalWrite(in4, LOW);
             digitalWrite(in5, LOW);
             digitalWrite(in6, LOW);
        }

        if(mesaj[0]==16){

          karar = "shoot";
             digitalWrite(enA, HIGH);
             digitalWrite(enB, HIGH);
           digitalWrite(in1, LOW);
           digitalWrite(in2, LOW);
           digitalWrite(in3, LOW);
           digitalWrite(in4, LOW);
           digitalWrite(in5, HIGH);
           digitalWrite(in6, LOW);
        }
        if(mesaj[0]==8){
```

```
        karar =  "ileri";
            digitalWrite(enA, HIGH);
            digitalWrite(enB, HIGH);
            digitalWrite(in1, LOW);
            digitalWrite(in2, HIGH);
            digitalWrite(in3, LOW);
            digitalWrite(in4, HIGH);
            digitalWrite(in5, LOW);
            digitalWrite(in6, LOW);
       }
       if (mesaj[0]==4){
        karar = "geri";
            digitalWrite(enA, HIGH);
            digitalWrite(enB, HIGH);
            digitalWrite(in1, HIGH);
            digitalWrite(in2, LOW);
            digitalWrite(in3, HIGH);
            digitalWrite(in4, LOW);
            digitalWrite(in5, LOW);
            digitalWrite(in6, LOW);
       }
       if(mesaj[0]==2){
         karar="sag";

           digitalWrite(enA, HIGH);
           digitalWrite(enB, HIGH);

            digitalWrite(in1, LOW);
            digitalWrite(in2, HIGH);
            digitalWrite(in3, HIGH);
            digitalWrite(in4, LOW);
            digitalWrite(in5, LOW);
            digitalWrite(in6, LOW);
       }
       if(mesaj[0]==1){
         karar="sol";
            digitalWrite(enA, HIGH);
            digitalWrite(enB, HIGH);
            digitalWrite(in1, HIGH);
            digitalWrite(in2, LOW);
            digitalWrite(in3, LOW);
            digitalWrite(in4, HIGH);
            digitalWrite(in5, LOW);
            digitalWrite(in6, LOW);
       }
       if(mesaj[0]==0){
         i=150;        //middle speed (pwm) value
         karar= "dur";
           digitalWrite(enA, HIGH);
           digitalWrite(enB, HIGH);

           digitalWrite(in1, LOW);
           digitalWrite(in2, LOW);
           digitalWrite(in3, LOW);
           digitalWrite(in4, LOW);
           digitalWrite(in5, LOW);
           digitalWrite(in6, LOW);
       }
    Serial.println(karar);

    delay(10);
}
```

Appendix II

Appendix III