X-CALI

Critical Design Review Report

"Design of Robots Collaboratively Carrying a Long Object Through an Open-Top Maze"

Submitted to: Ali Özgür Yılmaz

Date of Submission: March 9, 2018

Company Shareholders

- Oytun Akpulat 2093201 oytun.akpulat@metu.edu.tr
- Göksenin Hande Bayazıt 2093441 hande.bayazit@metu.edu.tr
- Emre Doğan 2093656 dogan.emre@metu.edu.tr
- Taha Doğan 2093672 taha.dogan@metu.edu.tr
- Burak Sezgin 2094456 burak.sezgin@metu.edu.tr

Contents

Introduction

This report describes the hardware and software progress and implementations carried out so far.

As the company X-Cali, we have completed most of the hardware implementation of the project. Physical architecture of the robot has been designed and almost finalized. We have also remarkably proceeded at the software part of the project. For subsystems, the design procedure was completed and final decisions were made. Even though we do not have a full functioning robot now, we made tests on subsystems and the results of these tests are included in this report.

Detailed description of the overall system and subsystems are given in this report. Technical details of designed systems were explained. Moreover, block diagrams, flow charts and 3D technical drawings of the subsystems are included. At the "Modifications to Conceptual Design" part, all the development and modifications to the system were clearly clarified. The cost and risk analyses of the robot are made. Finally, we mentioned our plans in "Development Schedules".

Project Goals and Objectives

Problem Definition

In this project, we aim design and produce a robot that can carry a long object, plank, through a maze collaboratively with another robot. Maze specifications is as follows.

- There are 5 streets maximum on the one side of the maze
- Street are 250 mm wide
- Maze walls are with 100 mm height
- Maze walls' width is 10mm
- 10 mm from the top of the maze walls are painted with black, the rest is white.

After robots are connected each other with the plank, one of them will be chosen as master and the other slave. Master leads the way in the maze and slave robot follows master. The main problem in this project is "Handling the corners". That is, in the case of U-turn, robots are not able to handle it at once. The whole procedure in such a case is given in the Figure 1.

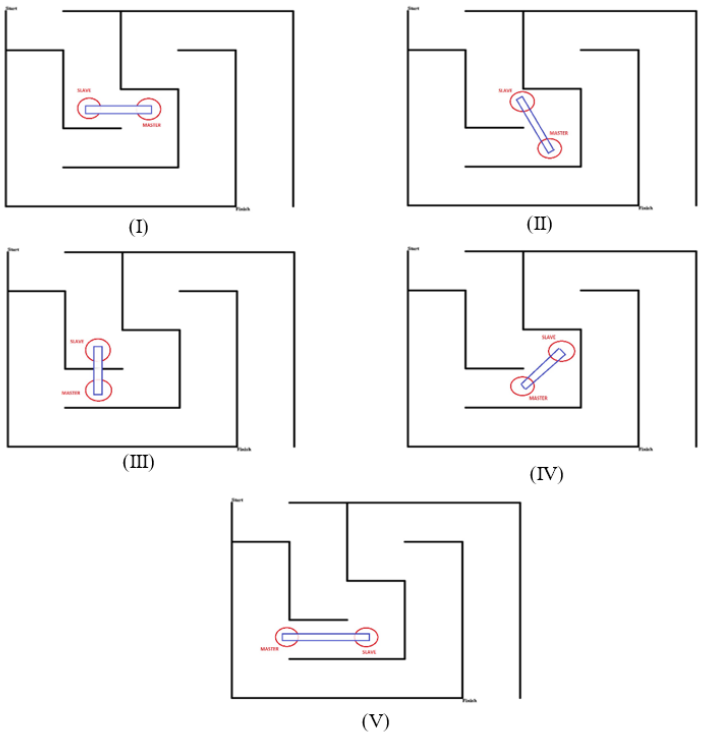


Figure 1: U-turn demonstration

Robots are not allowed to communicate each other directly. Thus, during the maze solving operation, our robot X-Cali performs *sensing*. Both active and passive sensing approaches are applied on X-Cali.

As X-Cali Group, our main aim is to design and produce a robot, is a solution for this maze problem, in in an efficient way. Efficiency is in terms of both software and hardware efficiency. Robustness is achieved firstly in the mechanical design period and it is improved with software. We have chosen the colors in a way that image processing is affected less by changing environment conditions such as light, brightness or shadows. Additionally, we have taken to account aesthetic beauty as we bear commercial concerns. The robot is designed symmetrical manner so that its posture increases its sales appeal. Because of the commercial concerns, after sale services are also provided for this design.

Objectives

Company Objectives

- Improving theoretical and practical experiences
- Applying theory on practice
- Continuing in grow and being a corporate
- Overcoming the project
- Increasing the team skills: productivity and efficiency and time management
- Placing first in the contest at the very end of semester

Project Objectives

- Minimalist and efficient design
- Considerably low power consumption
- Completing the maze without collision and damaging the other robot
- Keeping customer satisfaction high by providing high quality after sales services

Design Description

System Description

This project aims to build a robot that can collaboratively carry an object with another robot in an open-top maze. This duty's main requirements are designing a robot which is aware of its surrounding and which can freely, predictably and precisely move. To achieve this purpose, we have designed and built the robot in three main subsystems according to their functions. The robot includes a body part, which is the main chassis and passive components on it, a detection part that is composed of a camera, proximity sensors and algorithms that provides surrounding awareness, and finally movement part with motor drive system, tires and movement control algorithms. These subsystems are discussed and explained in detail further in the report. In addition, detailed block diagrams, flowcharts of these subsystems, technical drawings and general appearance of the robot are provided on Figures 2 to 6 below.

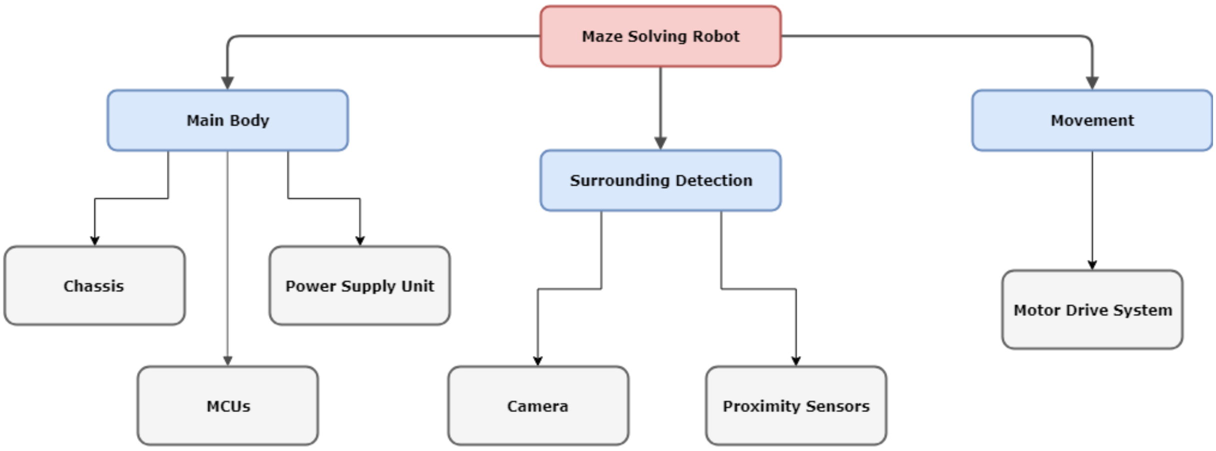


Figure 2: Block Diagram of the Overall System

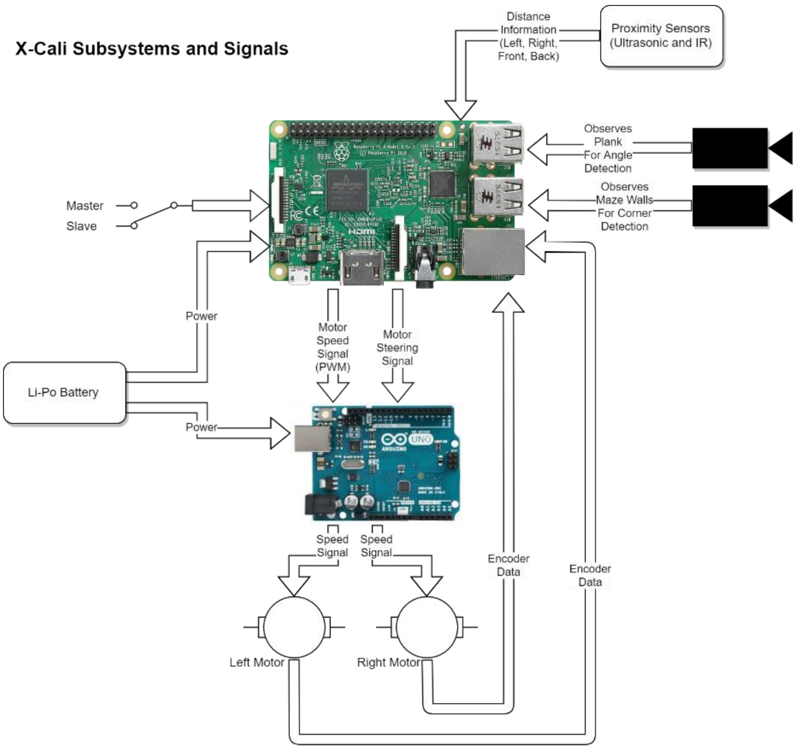


Figure 3: Flowchart of the System Functioning

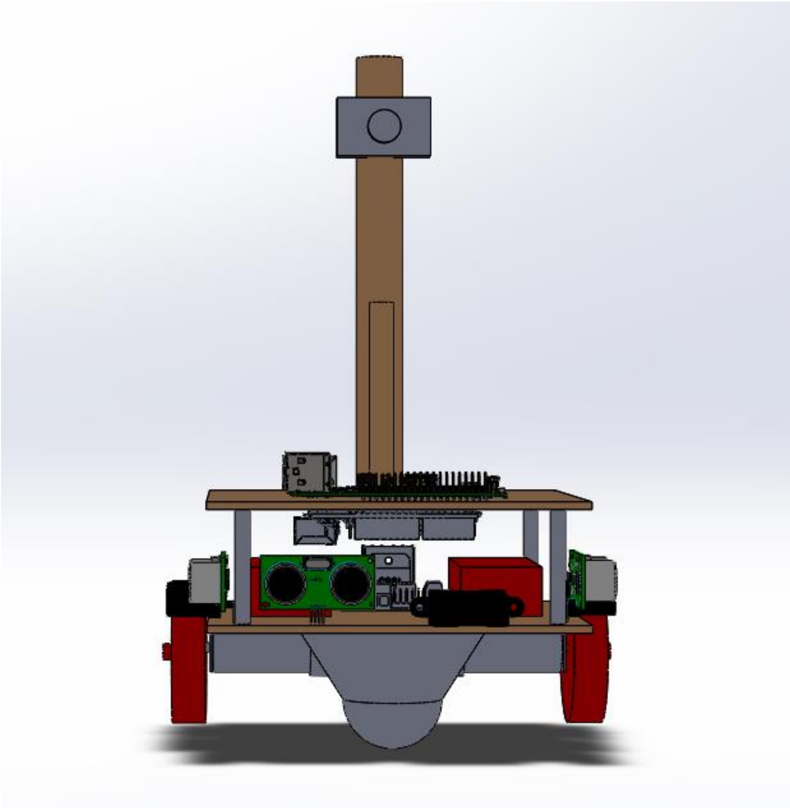


Figure 4: Technical Drawing of the Robot (Front View)

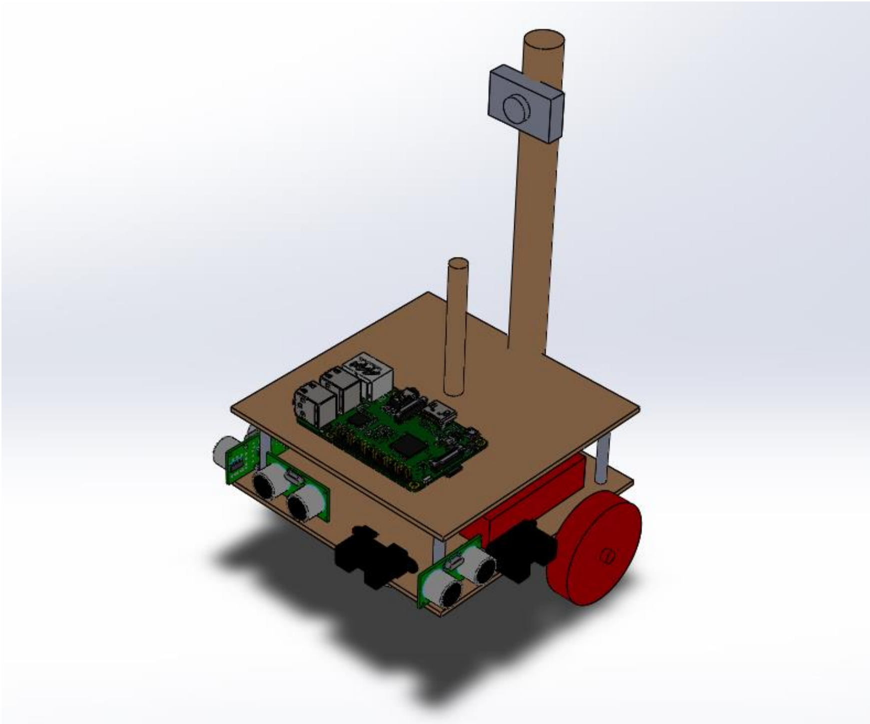


Figure 5: Technical Drawing of the Robot (Isometric View)

Figure 6: General Appearance of the Robot

Subsystem Analysis

Body

Chassis

The main platform of the robot is manufactured using hardboard considering its lower weight, durability, strength and convenience for mechanical assembly processes. As displayed in Figures 4 and 5 above, the robot is composed of two staged hardboard platforms that are 150 mm both in width and in depth. These stages are separated with metal supports that are 45 mm long. It was important for the footprint of the robot to be small because of restrictive standards and high maneuverability.

Two sticks are mounted on the chassis, for the plank and as a camera holder. These sticks are also made of wood, because of its lower weight and strength. Top of the holder stick of the plank is 170 mm above the ground, as determined in Standards Committee. On the holding point, a ball bearing is placed in order to reduce the friction of the plank while rotating. Camera holder is placed behind the plank holder and it is higher than it for a better perspective.

Heavier components are placed on the platform at the bottom, to decrease the level of center of gravity. Another consideration while placing the components is placing electronic parts that are in connection close to each other, to avoid complexity caused by cables and provide easy debugging.

Most of the passive components and motors are mounted on the platforms with screws and nuts to eliminate negative effects of mechanical vibrations on the system. Also, plastic clamps are used where necessary.

Power Supply Unit

The robot has two LiPo batteries that are 11.1 V and 1300 mAh. One of these batteries is to supply power to MCUs and the other supplies power to motor drive system. The main reason for using separate batteries is protection of sensitive electronic components from a damaged caused by high current. Also, at startup, motors draw high current, which causes a voltage drop on the other equipments that are connected to same battery. This issue causes a continuous “resetting” problem of MCUs. Batteries are placed on the bottom platform of the robot, as they are heavier components and it is necessary that they are close to the components that are going to be connected to them.

MCUs

The robot includes a Raspberry Pi 3 and an Arduino Uno. Arduino Uno is mainly used for motor driving and controlling purposes, whereas functions such as image processing, decision-making are performed using Pi.

Detection

In the robot, there are mainly three subsystems to have a knowledge of the surrounding of the robot. This subsystem supplies the most important data for the decision unit. All decisions will be made with respect to the signals taken from

- Camera
- IR Sensors
- Ultrasound Sensors.

These awareness sensors help the robot find its path in the maze and lead or follow the other robot. The structure of this module is given in Figure 7.

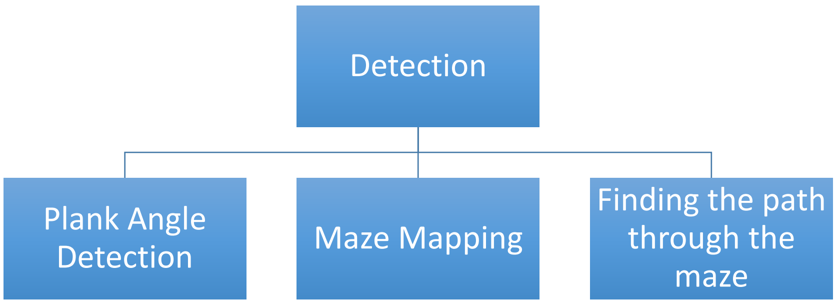


Figure 7. Block Diagram of the Surrounding Awareness Submodule

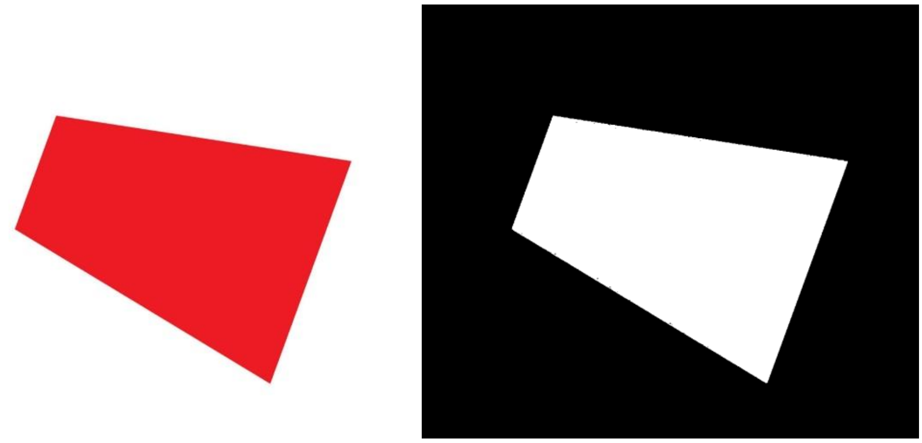
Angle Detection

In the plank angle detection, the image capture is taken from the camera and processed simultaneously. The main purpose is to achieve the angle of the plank to be able to identify the other robot's moves. With interpreting this angle in the critical points of the maze, the robot will be able to find its correct move in the decision unit.

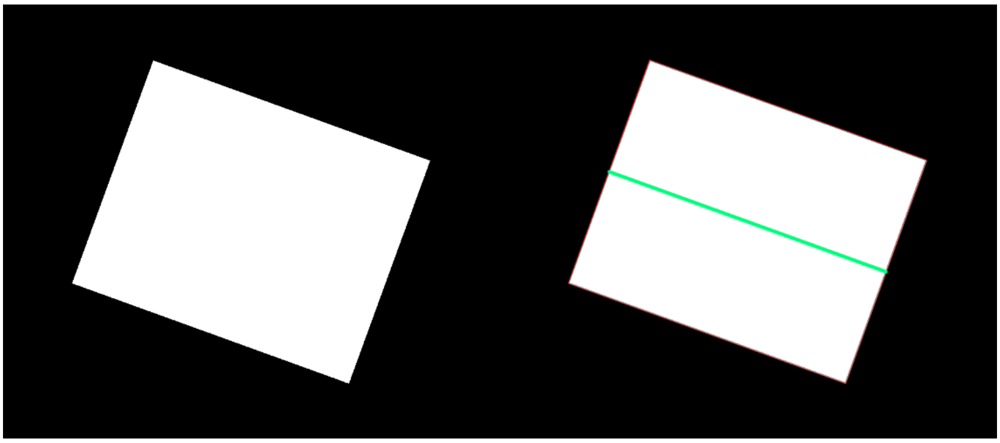
For this, we firstly developed an algorithm to be able to detect the correct angle. As the camera within the robot will see the plank in an uneven pattern, this image needs to be fixed so that we could find the angle in a more reliable way. The procedure is given below,

I. The colorful image is taken as an input
II. The input image is masked as processing in the grayscale is much more practical.
III. The uneven masked is image is applied some transforms to get a more linear shape.
IV. Angle finding algorithm is applied to the transformed image.

An example of this process can be observed in Figure 8.



(a) (b)



(c) (d)

Figure 8. The original input image (a), the masked version of the input image (b), the transformed version the input image, not uneven (c), and angle-detecting algorithm applied to the transformed image (d).

After the angle of the plank is detected, this information is sent to the decision unit so that the decision unit sends the necessary signals to the motor driver.

An illustration of this process can be seen in the command prompt in Figure 9.

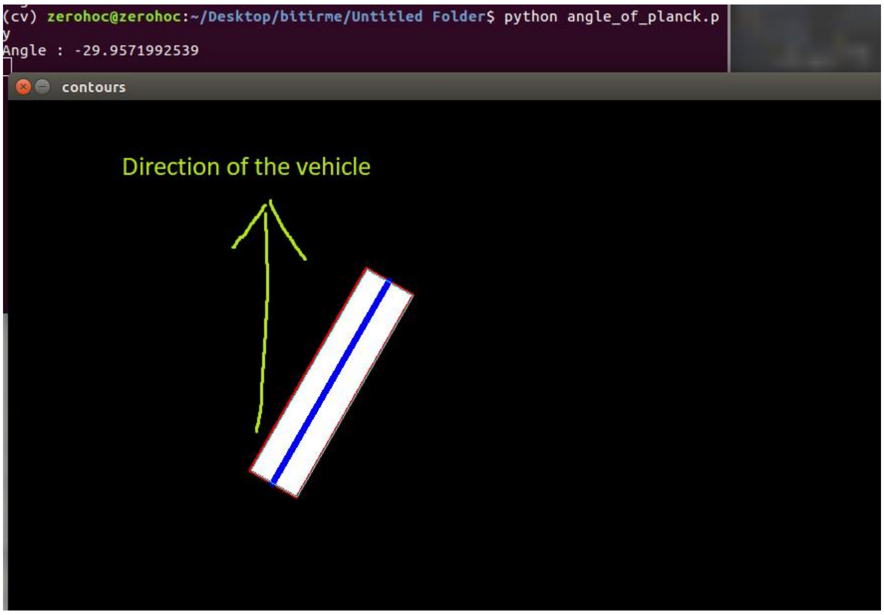
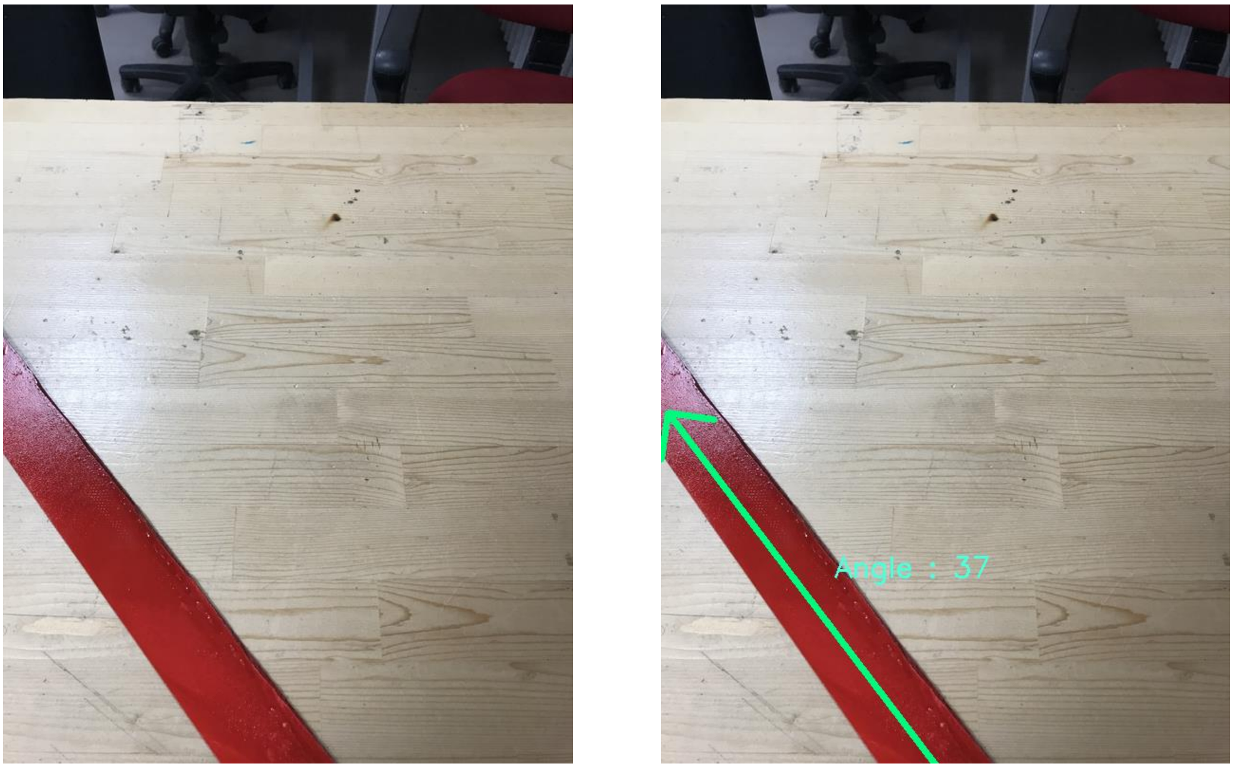


Figure 9: Angle Detection Output of the Plank.

All this algorithm process was debugged and developed on the computers. Then, we made a demonstration on the real robot, to see whether it could find the angle of the plank in a correct manner. Figure 10 given below is a basic demonstration of the algorithm on the Raspberry Pi 3, while all the robot components are integrated.



(a) (b)

Figure 10. Demonstration of the Angle Detection Algorithm on the Robot.

The camera captures the plank from the top view (a), and then the angle detection algorithm is applied to find the angle (b).

Maze Mapping

Besides the angle detection of the plank, we also use the camera to map the maze from the top view. Maze mapping is important because it eases to decide what the robot should do as its next move. For this mission, the camera used in the angle detection cannot be used, as the perspectives are quite different. Therefore, we will use another webcam camera to map the maze located higher than the angle-detecting camera.

Especially in the U and L turns, mapping the maze is important because the robot will remember its last moves and complete the turn with respect to this fact.

In our first try, we tried to map the maze visually such that while the robot is moving, the camera at the top will be observing the maze and from the edges of walls, it will create a map of the maze. A visual illustration of this mapping is given in Figure 11.

Figure 11. Maze Mapping Algorithm Visualization.

After mapping the path, this information is used to save a few movements which the robot completed. This becomes an important issue especially when there is a L or U turn. In the Figure 11, it is also available to observe these movement detections made.

This method is useful but it has some disadvantages. Mainly, this process is comutationally complex and it reduces the performance of other processes done in Raspberry Pi 3. So in stead of this, we thought about a new method of mapping which is less complex in computation.

In this method, a sufficient sized of matrix is created. As the robot goes through a path, this matrix is filled with “1”s to represent the path in the matrix. This method may seem strange but as it consumes much less CPU, it is preferrable rather than the visualization method mentioned above. An example of mapping array can be seen in Figure 12.

$$path\_matrix = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Figure 12. An Example of Mapping Matrix.

Path matrix is a 5x5 matrix initialized with all 0s. The robot begins its path from $path\_matrix[3,1]$ point. Then it follows the $path\_matrix[3,2]$, $path\_matrix[3,3]$, $path\_matrix[2,3]$, $path\_matrix[2,4]$, $path\_matrix[2,5]$. This visualization type is not strong but lets us to save the last few moves the robot made without occupying the processor.

Finding the Path through the Maze

Besides the plank’s angle detection and maze mapping, the robot should be aware of its surrounding while moving. For this subsystem, we have two options,

- Infrared (IR) Sensor
- Ultrasonic Sensor

IR and ultrasound sensors give information about the obstacles around the robot. This is quite important for the robot to be able to drive itself through the maze platform without hitting walls.

Both of them have their advantages explicitly.

Advantages of Ultrasonic Sensors over IR Sensors

- They are completely insensitive to environmental factors like,
  - Light
  - Dust
  - Vapor
  - Mist

Therefore, this sensor type makes the system more independent to the environmental factors, which is good.

Advantages of Infrared Sensors over Ultrasonic Sensors

- They perform better at defining edges of an area. As our robot needs to define the walls of maze and the other robot, using IR sensor might have a positive impact on the distance measurement performance.

For this performance difference, IR Sensor is more preferable rather than the ultrasonic sensor.

In our robot, we will keep both types of sensors. We still have doubts on the trade-offs between them. Most probably, one of them will not be needed. However, just in case we will be keeping both of them on the robot.

Movement

The robot has rear wheel drive system. Motion is achieved with two tires connected to the shaft of two DC motors at the back of the robot and a mad wheel at the front middle of the robot. Mad wheel provides mechanical stability to the system and increases the maneuverability.

In the second iteration of the robot, motors are replaced with DC motors with gearbox and encoders. Encoders are used to obtain instantaneous speed data from the motors, which is necessary for motor control.

Motor control is achieved with an Arduino and using PID control. Control algorithm is designed such that K_P, K_I and K_D parameters are determined using the data retrieved from proximity sensors and instantaneous speed data read by motor encoders.

Component Analysis

One of the most important and critical step in progress of development of the project is component selection. We made necessary researches and with respect to the results of these researches, we chose the components of our robot. These selections are explained below.

Microcontrollers

Our criteria while choosing the microcontroller was speed, low price and compatibility. After some research, we concluded that Arduino Uno, Figure 13, is the best option for us since it is cheap, fast and compatible.

It has 28 pins including 16 digital pin, 6 analog pin, five-power pin and a reset pin.[1] These are enough for our needs.

Figure 13: Arduino Uno

Main Computer

We needed to choose minicomputer that has the highest performance/price ratio. We also had to consider our budget and choose the computer accordingly. Raspberry Pi 3 was chosen.

Raspberry Pi 3, Figure 14, has HDMI, USB 2.0 and Ethernet connections, Camera Serial Interface, Display Serial Interface and 40 general-purpose input/output pins.[2]

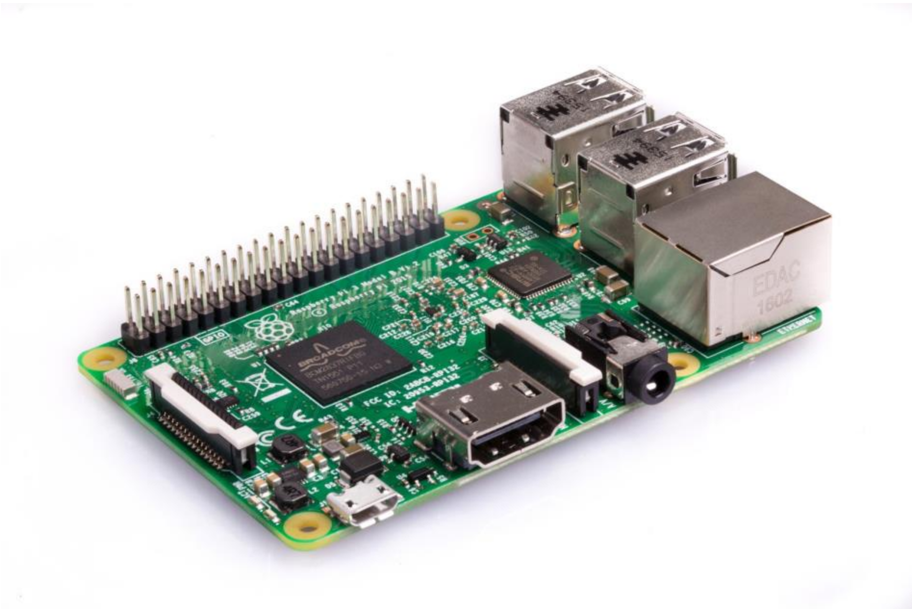


Figure 14: Raspberry Pi 3

Sensors

Ultrasonic Sensor

We used ultrasonic sensor in order to measure the distance. HC-SR04, Figure 15, was chosen for this purpose.

Figure 15: Ultrasonic Sensor HC-SR04

IR Sensor

IR sensor is also used for distance measurement.

Motors with Encoder

We selected a powerful enough motor with encoders. The data coming from the encoders will be processed in Arduino. Encoder data is important since it provides the robot with direction information.

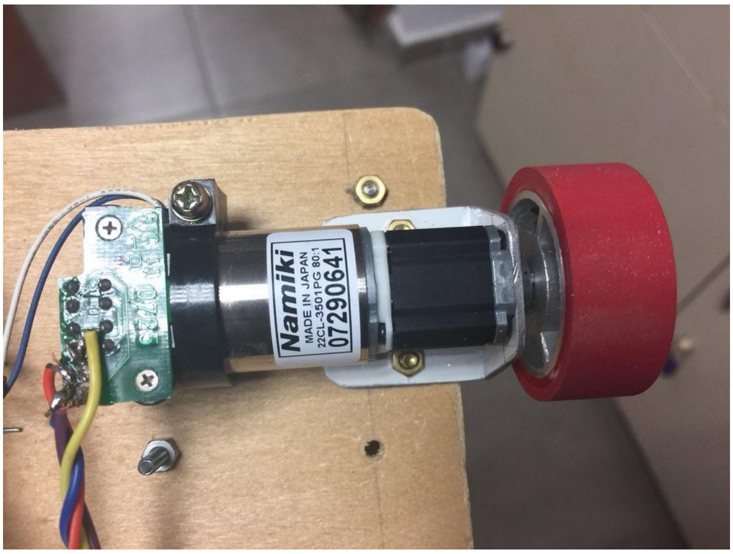


Figure 15: The motor

Motor Driver (H-Bridge)

Our priority was to select an H-Bridge that is cheap and compatible with the other parts of the robot. We chose L298N H-Bridge since it satisfies our requirements.

Two DC motor can be controlled with L298N, Figure 16, in both directions. It provides 2 A current in maximum case.[3]

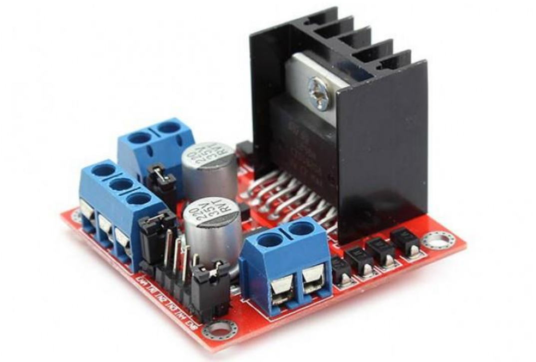


Figure 16: L298N H-Bridge

Battery

We needed to select a battery that is cheap, compatible with the other parts of robot, easy to use and easy to find information. We decided that the best choice is Li-Po battery.

Camera

Camera selection was one of the most critical parts of the project since our main solution relies on camera. After some research, we considered the performance/price ratios and decided to select a Bewell camera, Figure 17. It is a cheap camera; we can change it with A4 Tech PK-900H if our budget lets us.

Figure 17: Bewell Camera

Chassis and Mad Wheel

We selected a wooden chassis, which forms main body of the robot. It has several holes for motor connections, sensors, circuit cards and other components. Mad wheel is chosen as third wheel since it provides the robot with maneuverability.

Maze Platform

The maze the robots will proceed on it will be produced after the discussion with the other groups. The maze specifications are given in the Problem Statement part of the report. Sample Maze wall is given in Appendix A. We are working on two maze designs. In the first one, there will be holes on the bottom of the maze. Puzzle-like structure allows us to build maze with different paths. The other one is using double-sided tape. Using tape is a solution but temporary.

Requirement Analysis

All the requirements decided in the standard committee are evaluated in the Table 1.

Table 1: Requirement Analysis of the Project

# OF STANDARD	PROJECT STANDARDS	STATUS
1	Color of the ground and walls should be white.	Designed, but not implemented.
2	Height of walls should be100 mm.	Designed, but not implemented.
3	Upper side of walls (between 90mm height – 100mm height) should be black.	Designed, but not implemented.
4	Top of every wall pieces should have 1 cm width and should be black.	Designed, but not implemented.
5	Plank should have 3 mm thickness, 50 mm width and 500 mm length.	Designed and implemented.
6	Color of the plank should be red and material of	Designed and implemented.

	plank should be hardboard.	
7	Shape of the holding point should be circle. Diameter of this circle should be10mm. Length between centers of the holes is 400 mm.	Designed and implemented.
8	Robots of companies should fit into a square. This square has a 23cm side length.	Designed and implemented. Our robot fits into a square of 15 cm side length.
9	Height of the holding point should be 17cm.	Designed and adjusted.
11	The maze should be modular and reconfigurable. In other words, walls can be removed or added on the platform.	Designed, but not implemented.
12	Every team should design their own robot, maze and plank	Applied.
13	Every robot should be able to operate in each other's maze and each other's plank.	Not tried yet.
14	If the state changes, robots should behave accordingly.	Implemented in the software
15	Robots should move respect to right wall.	Satisfied. The algorithm is designed with respect to this fact.
16	The total cost of the project must be less than 200 USD.	Satisfied.

As it can be observed from Table 1, the standards related to the robot and the plank has all been considered and applied. The ones related to the maze were considered in our maze designing process but as we have not implemented it yet, it was not possible for us to observe them physically.

Risk Analysis

Mechanical Risks

- The chassis of the robot is wooden. Wooden materials may lose their stiffness as time passes. Thus, it can be fragile over time.
- If robot is treated harshly, the camera holder and plank holder may be broken from the chassis.

Safety Risks

- Li-Po Batteries in the robot may explode or start a fire or release poisonous gases during usage of the robot.
- Li-Po batteries may explode or start a fire or release poisonous gases during charging them.
- Overheating of microcontrollers may cause burns on the skin.

Customer Risks

- Two more companies are preparing to enter this business with the same aspects and goals. This may cause decrease in the customer or failure in the sales target.

Power Consumption Analysis

The robot's main power consuming parts are motors and Raspberry Pi 3. Their approximate powers in operation are:

- Raspberry Pi 3: 4.5 W
- Motors: 3 W

Total: 7.5 W

We have two separate batteries for Pi and motor drive system, with 11.1 V and 1300 mAh ratings. Using regulators, obtained energy from each battery (considering 90% efficiency) should be:

$$11.1\ \mathrm{V} * 1.3\ \mathrm{A.h} * 0.9 = 13\ \mathrm{W.h}$$

This is quite sufficient for the operation of the robot.

Cost Analysis

After a detailed research process, we checked the market in order to be able to figure out how to solve the problem with the minimized cost. By making a comparison list, we found our optimal component set and purchased them. This component list can be observed below in Table 2.

Table 2: Cost Analysis of the Project

Product Name	**Price / Product**	**Quantity**	**Total Price**
Raspberry Pi 3	₺181.65	1	₺181.65
HC-SR04 Arduino Ultrasonic Distance Sensor	₺5.50	4	₺22.00
E18 D80NK Infrared Sensor	₺22.75	2	₺45.50
Webcam	₺31.00	2	₺62.00
Motors	₺45.00	2	₺90.00
L298N Motor Driver	₺12.00	1	₺12.00
Mad wheel	₺5.00	1	₺5.00
Robot Chassis	₺10.00	1	₺10.00
Jumper Cables	₺10.00 (/set)	1	₺10.00
11.1V 1300mA LIPO Battery	₺69.50	1	₺69.50
LIPO Battery Charger	₺45.00	1	₺45.00
24V-5V 3A DC to DC USB Power Module	₺6.65	1	₺6.65
Arduino UNO	₺40.00	1	₺40.00
Screw set	₺30.00	1	₺30.00
Plank and ball bearings	₺20.00	1	₺20.00
		TOTAL PRICE	₺649.30

Please note that, we were allowed to spend up to $200,-. Total price is, according to today's currency, $171.05. That is, we are far below the top limit.

Development Schedule

As X-Cali, we were quite successful in our management plan despite the small breakdowns in our previous plan. Since we are halfway through and have little alterations in our plan, we saw fit to update our management plan and so our Gantt Chart. To start with, Communication Subsystem Design contents are updated to show the work to be done, more elaborately. Furthermore, Software Subsystem Design is divided into two branches as Raspberry Pi and Arduino, due to the addition of the Arduino. In addition, their contents are updated to fit the branches, as well. Integration of Subsystems part was affected from this change and, Raspberry Pi and Arduino implementation processes is added. The rest of the management plans is kept as it used to be.

Aforementioned plans of X-Cali can be seen in Figures 18 – 23.

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors
0			**X-Cali Project Management Gantt Chart**	**41 days**	**Thu 8.03.18**	**Thu 3.05.18**	
1			**Subsystem Designs**	**22 days**	**Thu 8.03.18**	**Fri 6.04.18**	
2			**Communication Subsystem Design**	**5 days**	**Thu 8.03.18**	**Wed 14.03.18**	
3			Algorithm for Master or Slave (Mode difference)	5 days	Thu 8.03.18	Wed 14.03.18	
4			Algorithm for U-Turn	5 days	Thu 8.03.18	Wed 14.03.18	
5			**Software Subsystem Design**	**7 days**	**Thu 15.03.18**	**Fri 23.03.18**	**2**
6			**Raspberry Pi**	**7 days**	**Thu 15.03.18**	**Fri 23.03.18**	
7			Maze Turn Type Detection Code Improvements	7 days	Thu 15.03.18	Fri 23.03.18	
8			**Arduino**	**7 days**	**Thu 15.03.18**	**Fri 23.03.18**	
9			Motor Steering Code	7 days	Thu 15.03.18	Fri 23.03.18	
10			Motor Speed Code	7 days	Thu 15.03.18	Fri 23.03.18	
11			**Subsystem Tests**	**22 days**	Thu 8.03.18	Fri 6.04.18	

Project: X-Cali Project Managem
Date: Thu 8.03.18

Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

Page 1

Figure 18. Gantt Chart (Management Plan) Page 1.

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors
12			Test Communication Subsystem	22 days	Thu 8.03.18	Fri 6.04.18	
13			Test Mechanical Subsystem	22 days	Thu 8.03.18	Fri 6.04.18	
14			Test Software Subsystem	22 days	Thu 8.03.18	Fri 6.04.18	
15			**Overall System**	**14 days**	**Mon 9.04.18**	**Thu 26.04.18**	**1**
16			**Integration of Subsystems**	**5 days**	**Mon 9.04.18**	**Fri 13.04.18**	
17			**Implement Software to Mechanical System**	**5 days**	**Mon 9.04.18**	**Fri 13.04.18**	
18			Raspberry Pi Implementation	5 days	Mon 9.04.18	Fri 13.04.18	
19			Arduino Implementation	5 days	Mon 9.04.18	Fri 13.04.18	
20			**Implement Communication to Mechanical and Software System**	**5 days**	**Mon 9.04.18**	**Fri 13.04.18**	
21			Raspberry Pi Implementation	5 days	Mon 9.04.18	Fri 13.04.18	
22			**Overall System Tests**	**14 days**	**Mon 9.04.18**	**Thu 26.04.18**	
23			Test the Overall System	14 days	Mon 9.04.18	Thu 26.04.18	

Project: X-Cali Project Managem
Date: Thu 8.03.18

Task	Inactive Summary	External Tasks
Split	Manual Task	External Milestone
Milestone	Duration-only	Deadline
Summary	Manual Summary Rollup	Progress
Project Summary	Manual Summary	Manual Progress
Inactive Task	Start-only	
Inactive Milestone	Finish-only	

Page 2

Figure 19. Gantt Chart (Management Plan) Page 2.

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors
24			**Preparing Presentation for Demo**	**5 days**	**Fri 27.04.18**	**Thu 3.05.18**	**15**
25			Prepare Power Point Presentation	5 days	Fri 27.04.18	Thu 3.05.18	
26			Practice for Presentation	5 days	Fri 27.04.18	Thu 3.05.18	

Project: X-Cali Project Managem
Date: Thu 8.03.18

Task	Inactive Summary	External Tasks
Split	Manual Task	External Milestone
Milestone	Duration-only	Deadline
Summary	Manual Summary Rollup	Progress
Project Summary	Manual Summary	Manual Progress
Inactive Task	Start-only	
Inactive Milestone	Finish-only	

Page 3

Figure 20. Gantt Chart (Management Plan) Page 3.

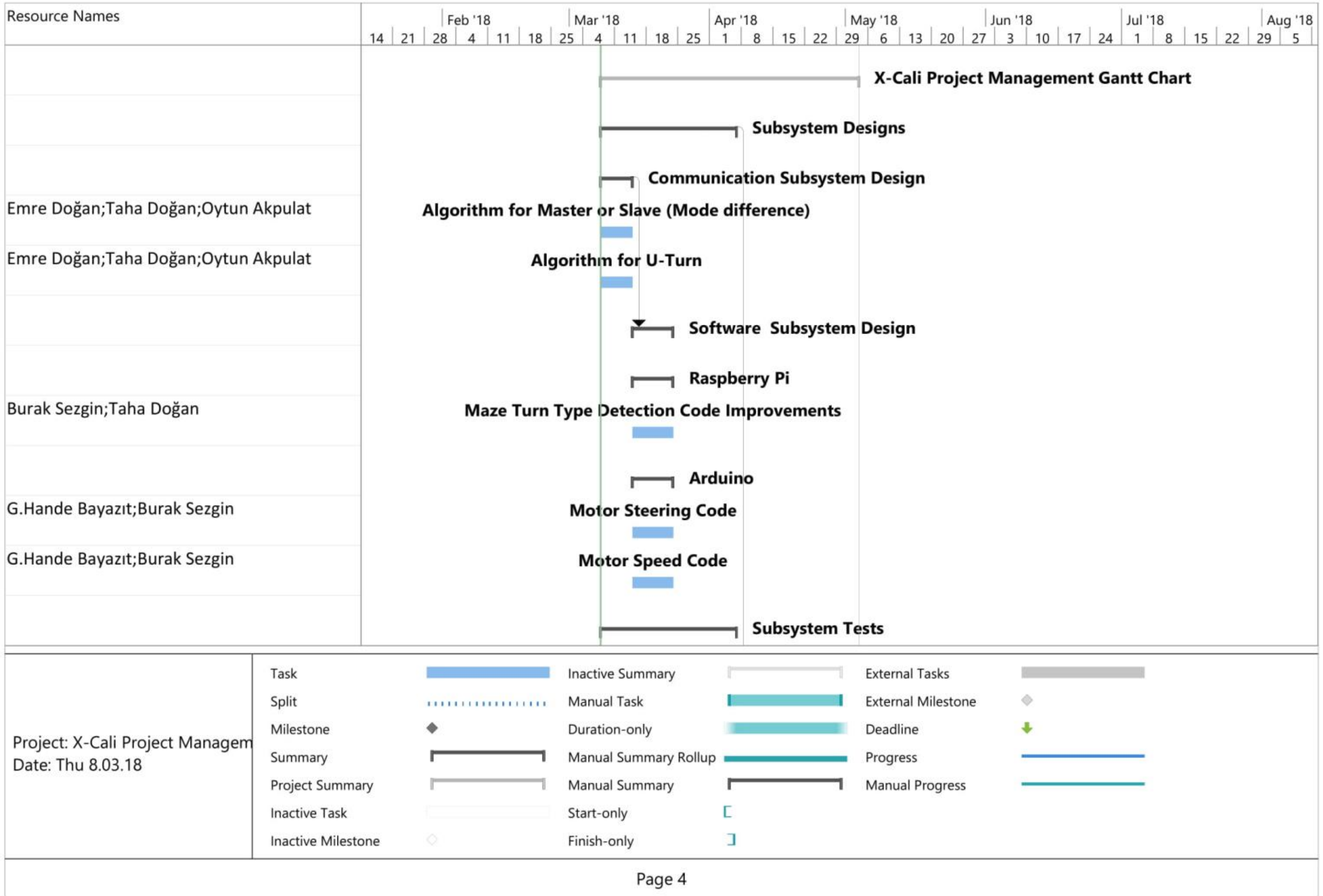


Figure 21. Gantt Chart (Management Plan) Page 4.

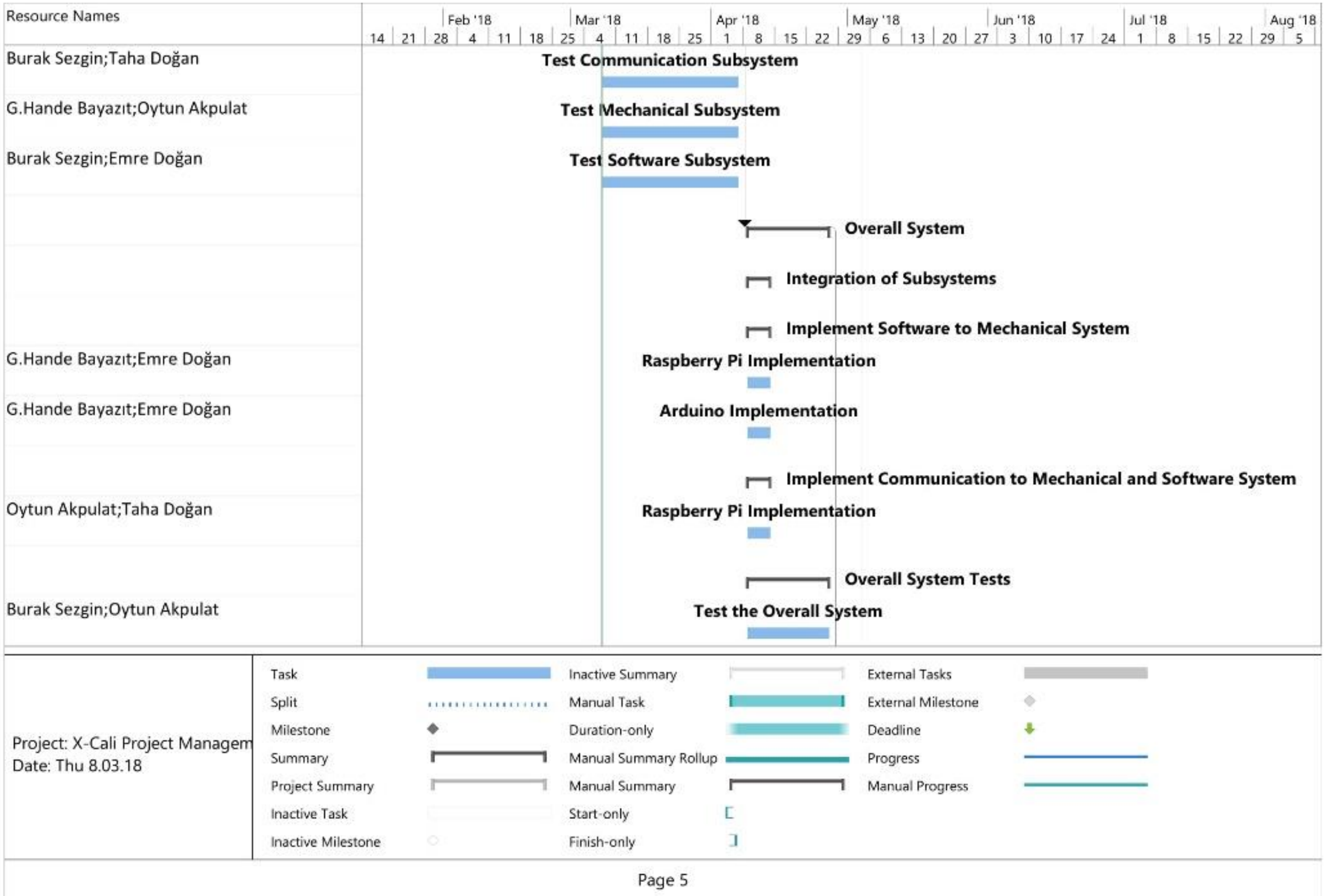


Figure 22. Gantt Chart (Management Plan) Page 5.

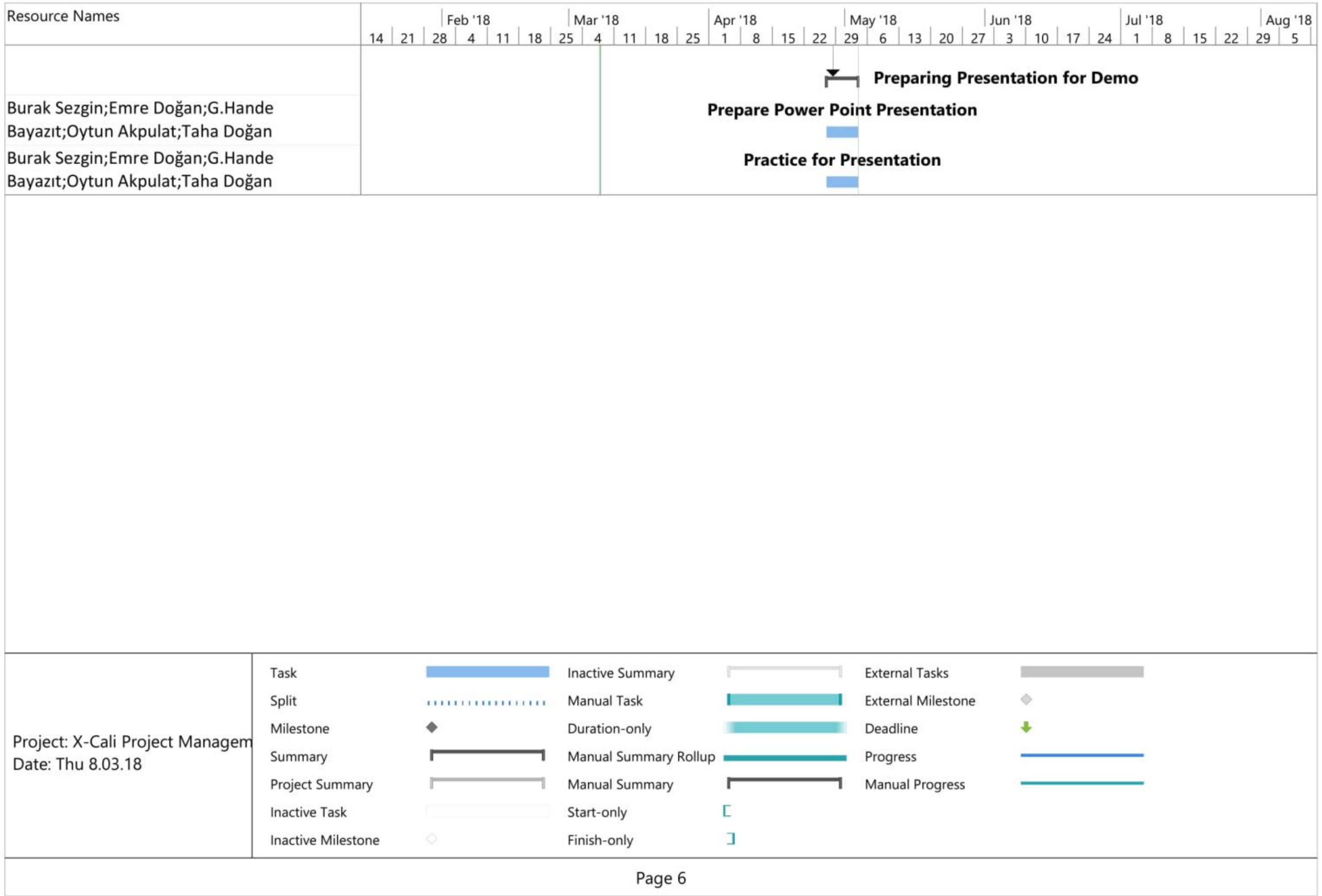


Figure 23. Gantt Chart (Management Plan) Page 6.

Modification to Conceptual Design

In the Conceptual Design Report, different solution approaches and details of both software and hardware parts of the project were explained. After the report was submitted, we made some tests and evaluated advantages and disadvantages of the choices we made in hardware components. According to the results of this evaluation, we decided to modify some parts of the hardware of the project. These modifications are as follows:

- Arduino will be used for motor control since we already have a motor control code that works properly in Arduino.
- We decided to add an extra camera. That is, robot will have two cameras instead of a single camera. The reason for this is that both the movement of the plank and top of the maze could not be observed with a single camera.
- The motors we used in the demo in first semester was not powerful enough. Hence, we have bought new motors, Figure 24, that are powerful enough and have encoders. Encoders was needed for obtaining direction information. The data, which is coming from encoder, will be processed in Arduino.

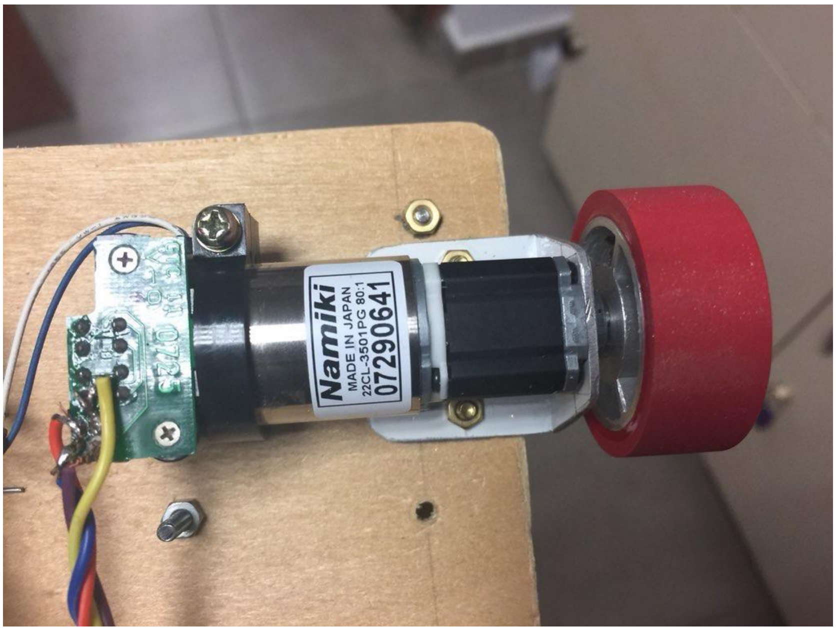


Figure 24: New motor with encoder

Test Procedures and Test Results

Before the integration of the angle measurement code and camera on the vehicle, they are tested. A4 Tech PK-900H model of webcam is used for this test to measure the angle of the plank. The test specifications are provided below.

Environment: Living room, luminous

Driver: RaspberryPi3

Camera: A4 Tech PK-900H

Temperature: 24°C

Measurement Tool: IC Measure (On-Screen Measurement Software)

Measurement Method (Angle): First, some arbitrary angles are adjusted for plank and it is photographed. Then, with the help of the angle measurement code, plank's angle on photo is measured. Furthermore, the software (IC Measure) measures real angle of the plank.

Measurement Method (FPS): First, the number of frames of the video is set to specific values such as 100, 200 … up to 1000 frames. Then, for these numbers of frames, FPS is calculated as follows: a stopwatch code is utilized which measures time difference between the first frame and the last one. Moreover, to find FPS value, frame number is divided into that time difference.

The following graphs are obtained using MATLAB.

Table 3: Measurement results of angle by camera

(In a range between -90 and 90 degrees)

Real Angle (Degrees)	Measured Angle by Camera (Degrees)
-70,5	-70
-56,5	-56
-34	-36
-15,5	-17
1	0
16,5	18
35	36
53,5	54
72,5	72
83,5	82

Table 4: Measurement results of FPS for all circumstances

(For when plank is detected and not detected)

	When plank is not detected	When plank is detected
Number of frames	Measured FPS	Measured FPS
100	22,42	21,59
200	21,62	21,34
300	21,94	21,95
400	22,42	21,41
500	22,29	21,68
600	21,93	21,58
700	22,09	21,61
800	22,14	21,34
900	22,14	21,86
1000	22,28	21,62

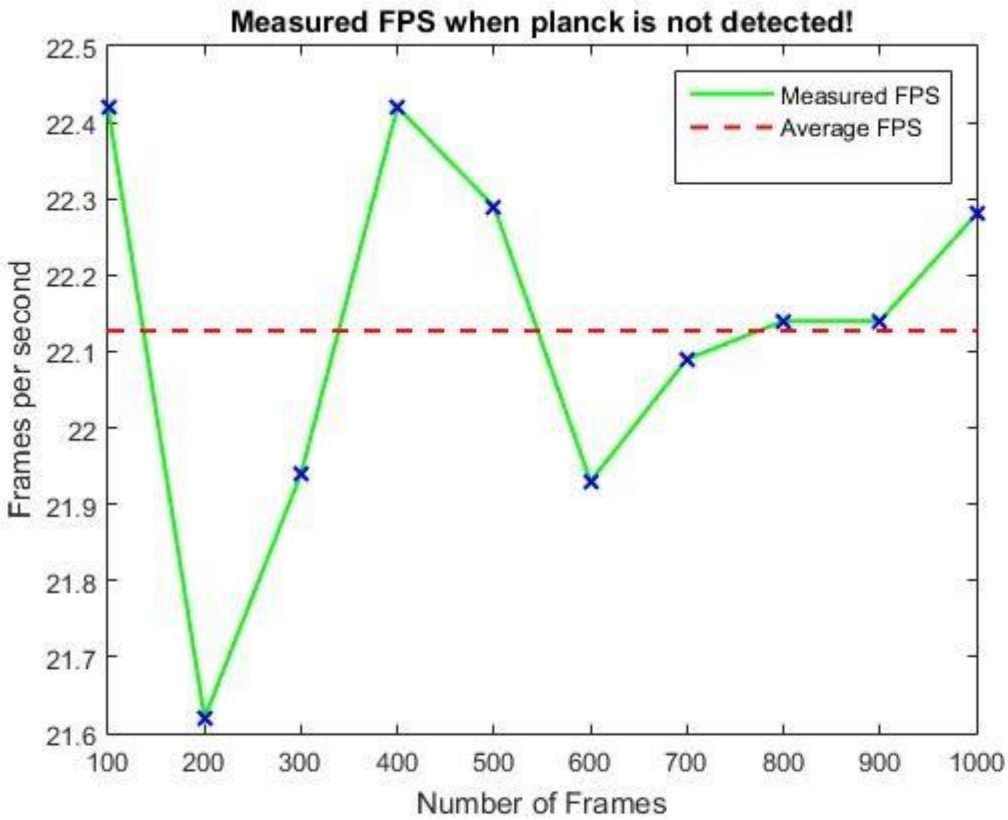


Figure 25: Measured FPS when plank is not detected

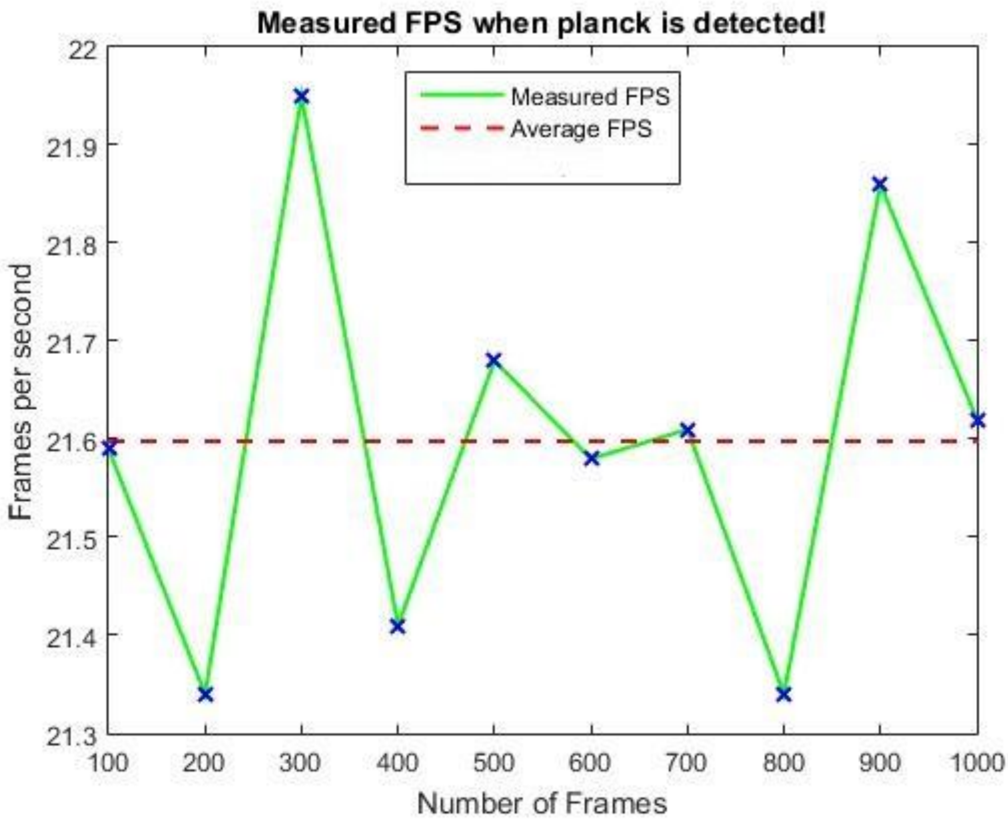


Figure 26: Measured FPS when plank is detected

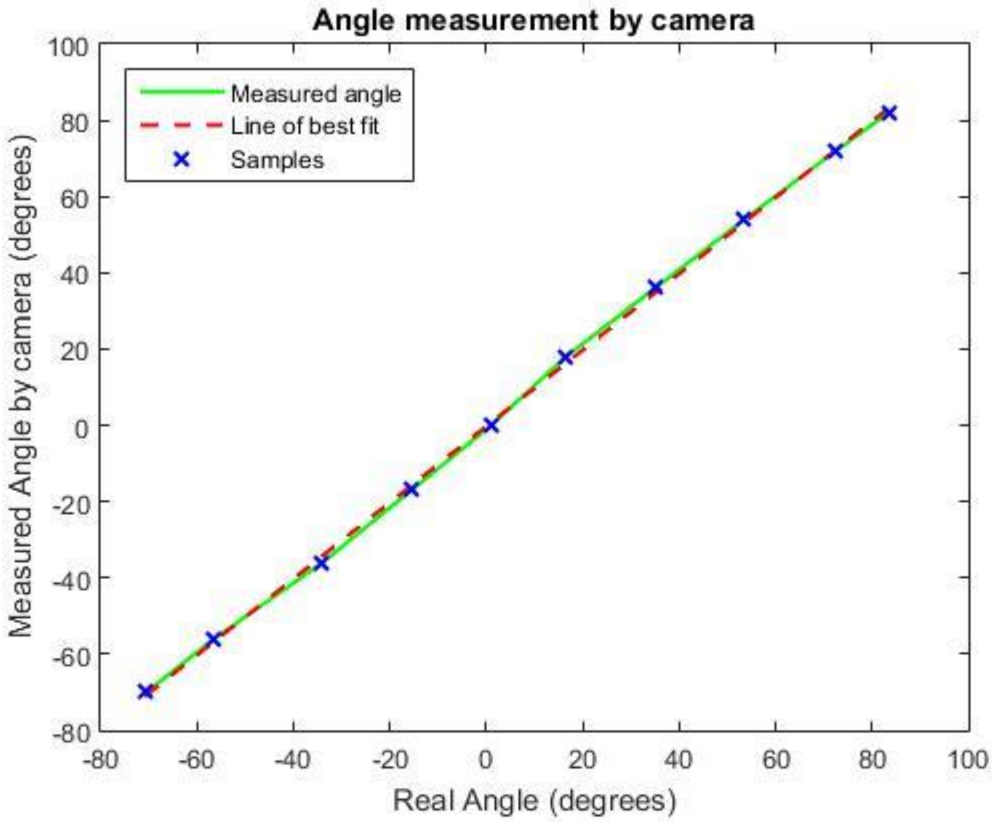


Figure 27: Angle measurement results of camera

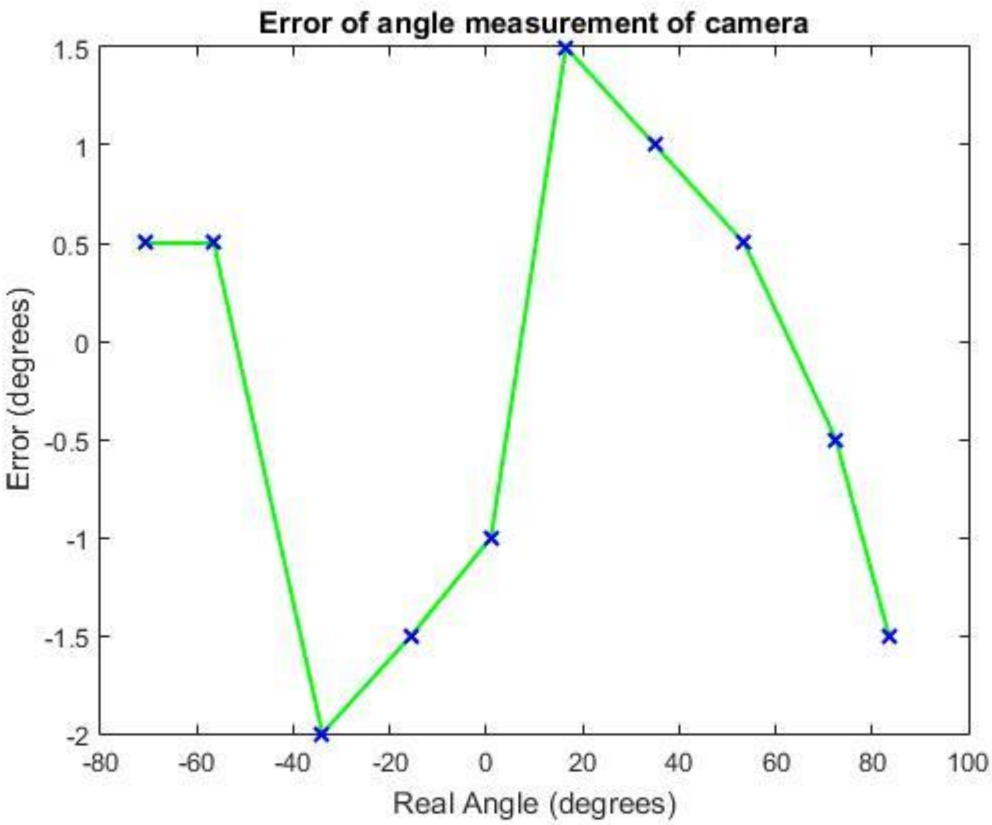


Figure 28: Error of angle measurement of camera

Deliverables

The first robot of X-Cali is designed for general usage. There is no specific client profile for our product. The robot can be used for different purposes such as gaming or educational purposes.

The expected deliverables of the work packages of our project can be seen in Table.

Table 5: Expected Deliverables of the Work Packages of the Project

The Work Package	Corresponding Deliverable	Status
Research	Tentative Report	Completed
Component Tests	Results and analysis of the component test	Completed for the ultrasonic sensors and RP3. Test plan is achieved.
Communications Subsystem Design	Results of the procedure of receiving& processing data	Not completed.
Mechanical Subsystem Design	Driving tests and analysis of the robot	Completed. Test plan is achieved.
Software Subsystem Design	Documentation of the algorithms and debugging results	Started but not completed.
Overall System Implementation & Tests	A robot completing the labyrinth by itself	Not completed yet.
Demonstration	A robot completing the labyrinth collaboratively with the other groups. The product within its package.	Not completed.

The package of our product will include the main body of the robot, a plank, user manual, two spare tires, a backup battery and a remote controller deciding the robot to become master or slave.

The size of the robot can be adjusted according to the customers' demands. The product will be prepared in 10 weekdays after the order. Users can find all the necessary information about the product in the user manual.

You can contact us via our web site http://www.xcali.ml.

Organization Plan

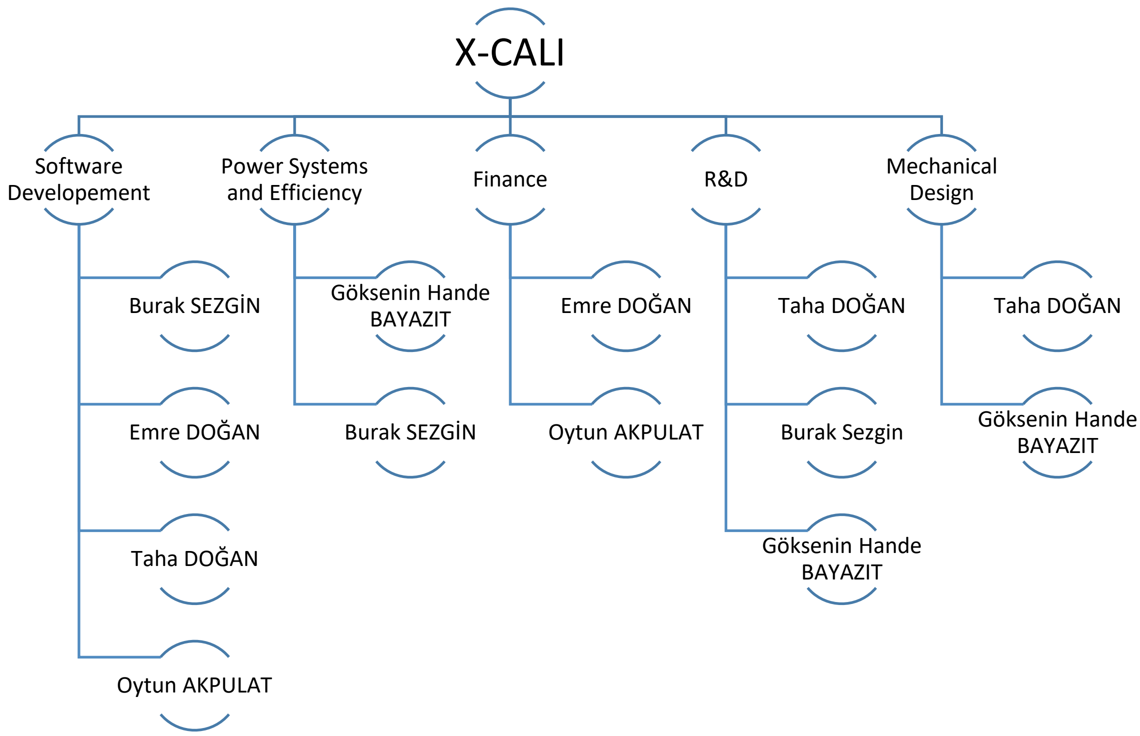


Figure 29: Organizational Structure of **X-Cali**

Conclusion

The engineers of the company X-Cali have prepared this report in order to inform customers about the progress of the project implementation. The present situation of the project was explained elaborately. The detailed solutions we have come up with were included.

All the subsystems, modifications to the conceptual design, the risk analysis and safety issues were included. Necessary tests were made for the subsystems. The test procedures were explained and the test results were interpreted in this report. Detailed explanations of the overall system and subsystems were given. Team members went shopping last week and bought necessary components, hence the cost table was updated. Gantt chart was also updated.

At this point, we can say that our robot is almost ready. Our design and implementation is almost finished, only slight improvements will be done. We are working to succeed and finish this project as soon as possible.

References

[1] https://www.farnell.com/datasheets/1682209.pdf

[2] https://www.raspberrypi.org/documentation/hardware/computemodule/RPI-CM-DATASHEET-V1_0.pdf

[3] http://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html

Appendices

Appendix A

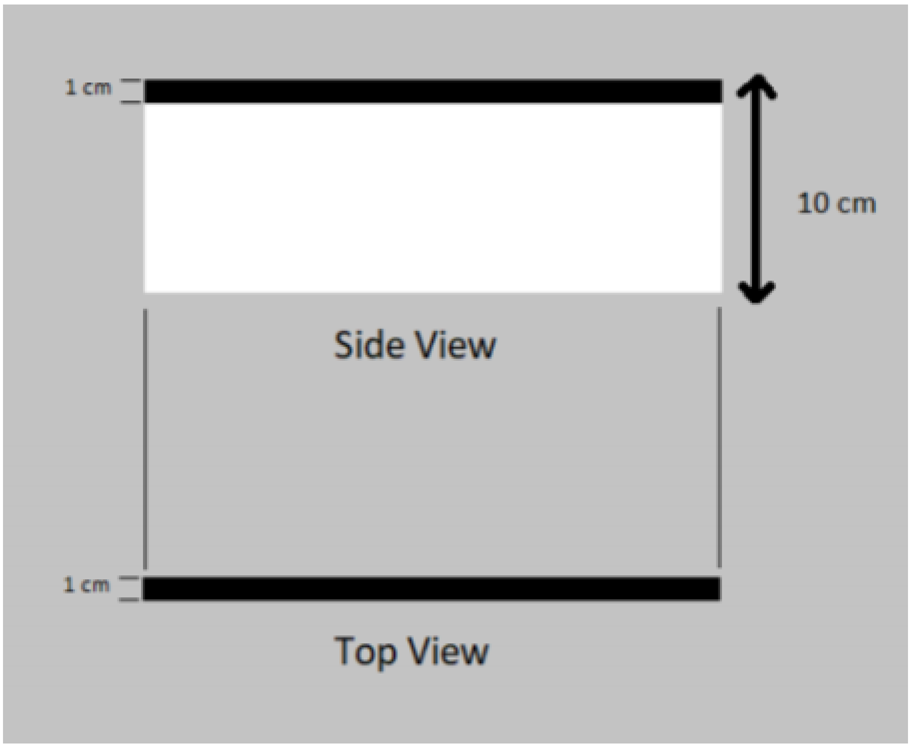


Figure 30: Maze wall drawing