**X-CALI**



**Conceptual Design Report**

*“Design of Robots Collaboratively Carrying A Long Object Through An Open-Top Maze”*

Submitted to: Ali Özgür Yılmaz

Date of Submission: 29/12/2017

**Company Shareholders**

* Oytun Akpulat 2093201 [oytun.akpulat@metu.edu.tr](mailto:oytun.akpulat@metu.edu.tr)
* Göksenin Hande Bayazıt 2093441 [hande.bayazit@metu.edu.tr](mailto:hande.bayazit@metu.edu.tr)
* Emre Doğan 2093656 [dogan.emre@metu.edu.tr](mailto:dogan.emre@metu.edu.tr)
* Taha Doğan 2093672 [taha.dogan@metu.edu.tr](mailto:taha.dogan@metu.edu.tr)
* Burak Sezgin 2094456 [burak.sezgin@metu.edu.tr](mailto:burak.sezgin@metu.edu.tr)

**TABLE OF CONTENTS**

[ABSTRACT 3](#_Toc502321467)

[INTRODUCTION 4](#_Toc502321468)

[PROBLEM STATEMENT 5](#_Toc502321469)

[SOLUTION APPROACH 7](#_Toc502321470)

[MOVEMENT 7](#_Toc502321471)

[SENSORS 8](#_Toc502321472)

[DECISION AND CONTROL 9](#_Toc502321473)

[Solution I: Image Processing for Collaboration 9](#_Toc502321474)

[Solution II: Processing the Information Retrieved from Plank for Collaboration 10](#_Toc502321475)

[PLANS 13](#_Toc502321476)

[MANAGEMENT 13](#_Toc502321477)

[Responsibilities Among the Group Members 13](#_Toc502321478)

[Detailed Version of Gantt Chart 14](#_Toc502321479)

[Cost Analysis 18](#_Toc502321480)

[Deliverables 19](#_Toc502321481)

[ENGINEERING 21](#_Toc502321482)

[Test Plans 21](#_Toc502321483)

[CONCLUSION 31](#_Toc502321484)

[REFERENCES 32](#_Toc502321485)

[APPENDICES 33](#_Toc502321486)

# ABSTRACT

This report provides the problem statement and solution procedures, together with management and engineering plans, of the very first project of X-Cali.

X-Cali is a company that was created by five undergraduate students. In our company, it is aimed to produce robots with high-quality and that include latest technology items.

In the transportation sector, import and export goods have a large percentage. Nevertheless, it is not possible to carry all goods in a similar way due to their shape and other features. For instance, to carry a long object, a forklift may not be a good resolution. We focused to carry a long object in a maze (an example to a difficult environment to carry a goods) that the object, normally, is not suitable to turn the corners.

The main issue that needs to be dealt with in this project is monitoring and determining the other robot’s next possible movement, since direct communication is off-limits and due to the constraints, the robots cannot make U-Turn at once.

To solve this trouble, as first solution, “Image Processing for Collaboration” is preferred. With image processing, the angle between plank and robot’s direction vector will be measured by making use of the color differences. And with the help of the angle, other robot’s position will be determined by mathematical and algebraic calculations.

“Hall Effect Encoder and 3-axis Gyroscope” is chosen to be the second solution to our main problem. A Hall Effect Encoder can find the rotation angle of plank that is mounted on the robot. Nevertheless, detecting the position of angle (whether it is in positive or negative direction) is not among the functions of the sensor. Therefore, 3-Axis Gyroscope is, also, utilized in our method to compensate for the lack of the Hall Effect Encoder.

The product package includes the robot itself (size is determined by customer), a plank, 2 spare tires, a back-up battery, a remote-controller, a user manual and 2-year warranty (extendible). Complete package (optional) also includes another robot and a maze. For further information and contact, please visit our website “<http://www.xcali.ml/>”.

# INTRODUCTION

This report includes problem statement and detailed information about the solution of the project, test results, management and engineering plans of the X-Cali company.

In the project which is chosen by the company among five projects, two robots will collaboratively will carry a long object, that is called ‘’plank’’, through an open top maze. The fact that direct communication is not allowed between the robots makes this project hard to implement. Two robots are supposed to be able to make U and L turns carefully and solve the maze. There are also some restrictions and specifications about the maze and the plank which is explained in detail in the problem statement part of this report. The main reason of the choice of the project is that it does not require complicated mechanical implementation and mainly composes of software and algorithm parts. While this report is being written, the software implementation is still in progress.

# PROBLEM STATEMENT

In this project, carrying a long object through an open-top maze with two robots, which are not allowed to communicate each other directly, is aimed. The field that robots are carrying a long object is a maze. The maze specifications are given below.

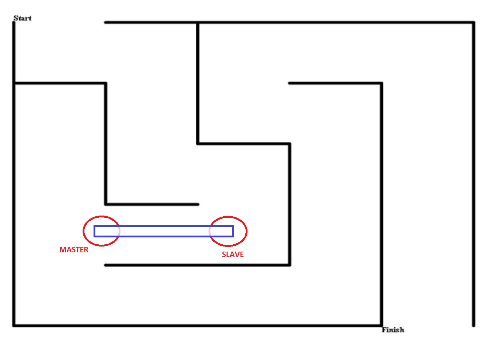
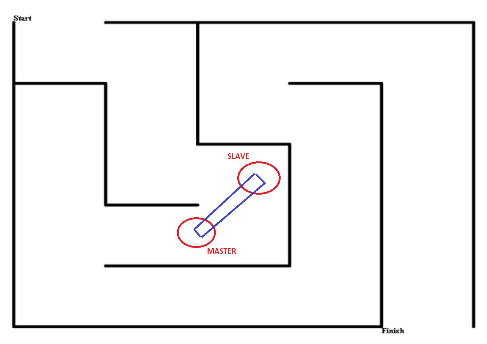
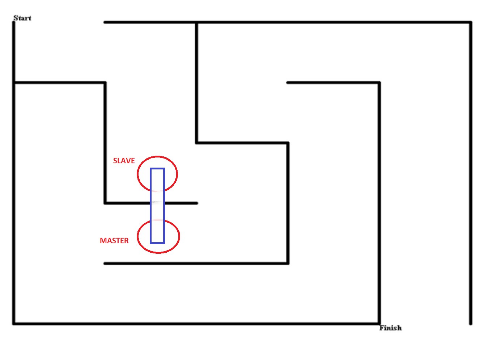
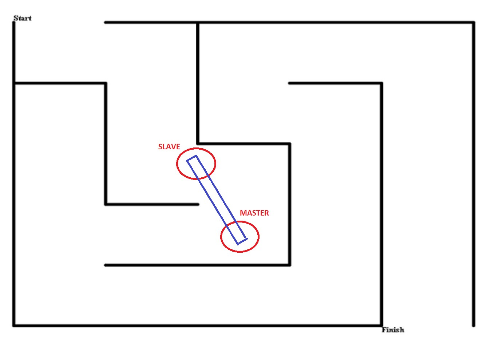
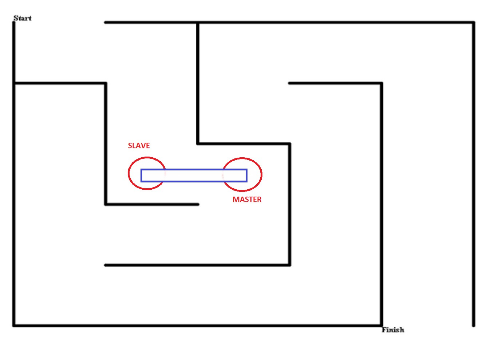
* Maze streets are 250 mm wide
* There are , one entrance and one exit in presence
* Maze pattern is arbitrary However, there is no dead-end.
* Height of the walls: 100 mm. From the bottom of the walls, between 90 mm and 100 mm and the top of the walls are painted black, the rest is white.
* Width of the walls is 10 mm.

When solving the maze “follow the right wall” rule is chosen during the Standard Committee meetings. Although the maze definition is not a real maze due to no dead-end case.

Two robots are expected to carry a long object called *plank*. No sensors or no devices that is mounted directly to the plank is not allowed. Thus, determining and processing the collected data are handled with the devices and sensors that are on the robot itself. The specifications for plank is given below.

* The length of the plank is 500 mm
* Distance between two holders is 400 mm.
* Width of the plank is 50 mm
* Color of the plank is red.
* Plank is elevated 170 mm from the top of the maze platform to the bottom of the plank

According to plank and wall specifications, these two robots cannot handle the *U-turn* at once. In order to turn *U-corners*, robots will perform indirect communication. Thus, the main problem of this project is that observing and determining the other robot’s next move so that two robots can collaboratively proceed in the maze. The illustration of *U-turn* is shown in Figure 1, below.



(I)

(II)

(III)

(IV)

(V)

Figure 1: Illustration of U-Turn

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.

One of the robots are dedicated as master and the other one is slave. Slave observes master so that they can handle such turns.

According to aforementioned physical constraint, the X-Cali maze solver robot proceeds on the path. The interpretation and handling of the path are explained under the Solutions section, in detail.

# SOLUTION APPROACH

The solution procedure that we provide can be investigated as several subsystems, with various options due to redundancy.

## MOVEMENT

Movement is the first milestone of the project. Although it seems to be easy to implement, its sustainability is hard to achieve. Nevertheless, our solution about the drivetrain of the robot includes no other alternative than the below-explained one.

In X-Cali, RaspberryPi3 and L293D is used. Motors are two DC Motors. DC motors are chosen, because it provides two options for speed control such as

* PWM
* Voltage Control

In order to drive motors, power supply is used during early stages of the project. After full mobilization of the robot is achieved, two separate Li-Po batteries are going to take place of the power supply. The main reason for using two batteries is to isolate the power flow of the motors and the RaspberryPi3, since current drawn by the motors may result in a large voltage drop on the motor-side load, which decreases the voltage drop on RaspberryPi3 and causes restart, malfunctioning problems.

Another alternative to using two separate batteries is using a single battery instead of two, with appropriate regulator circuits and filter inductances.

Motors are not connected directly to the RaspberryPi3. Sinking current for motors from RaspberryPi3 is dangerous for the microprocessor. Thus, by buffering the motors from the control units, we provided safety for the overall circuitry.

X-Cali is chosen as rear-wheel drive. The main reason of this can be understood by thinking forklifts. Rear-wheel driving provides more instability however, this is something we desire. It provides more maneuverability. When maze dimensions are considered, small turn radii have to be achieved. Additionally, rear-wheel driving, we use tank-like wheel movements. This phenomenon is illustrated in the Figure 2.

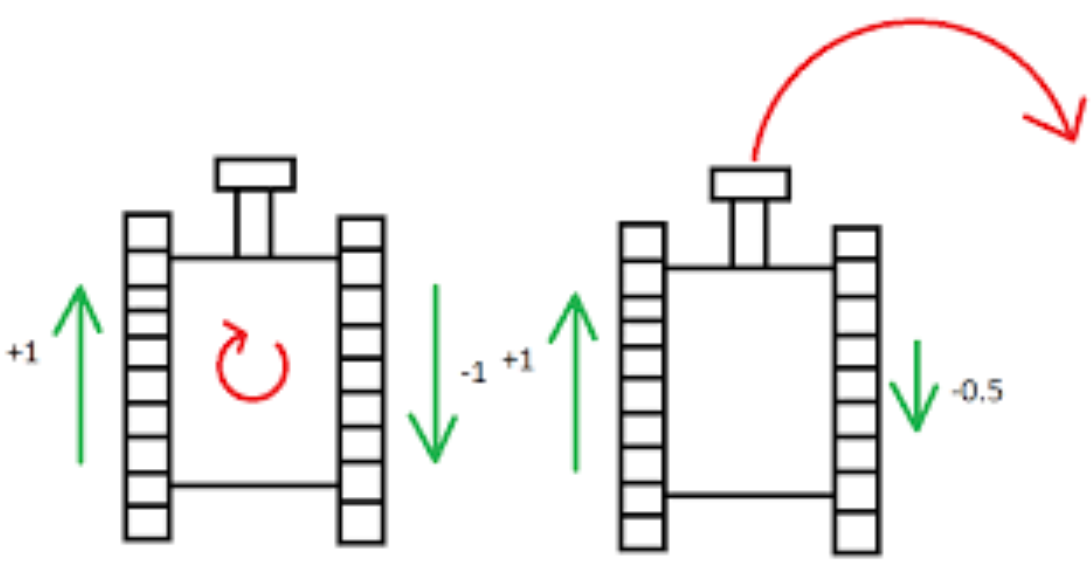


Figure 2: Illustration of Rear Wheel Driving System

A mad-wheel at the front is just used for keeping the balance of the robot.

## SENSORS

As sensors, after debating group members, we came up with three options due to redundancy considerations:

1. Ultrasonic Proximity Sensor
2. IR Proximity Sensor
3. Camera

The first two are active sensors. That is ultrasonic proximity sensor sends sound waves and receives the reflected waves. By calculating the difference between emitting time and receiving time, Python program that is embedded in RaspberryPi3 calculates the distance. IR Proximity works with the same principle but rather than sound waves, it uses light.

They have both advantages and drawbacks. Sonar sensors measure the distant obstacle with high accuracy, i.e. further than 500 mm. Whereas, IR sensor can measure the close objects, i.e. closer than 100 mm.

The camera is passive sensor, namely it does not change the environment. The first two proximity sensors are used to proceed in maze and to interpret the maze walls. The main sensor that determines the next movement is the camera. The camera is the crucial part of this solution. The algorithm is described under Decision and Control part.

## DECISION AND CONTROL

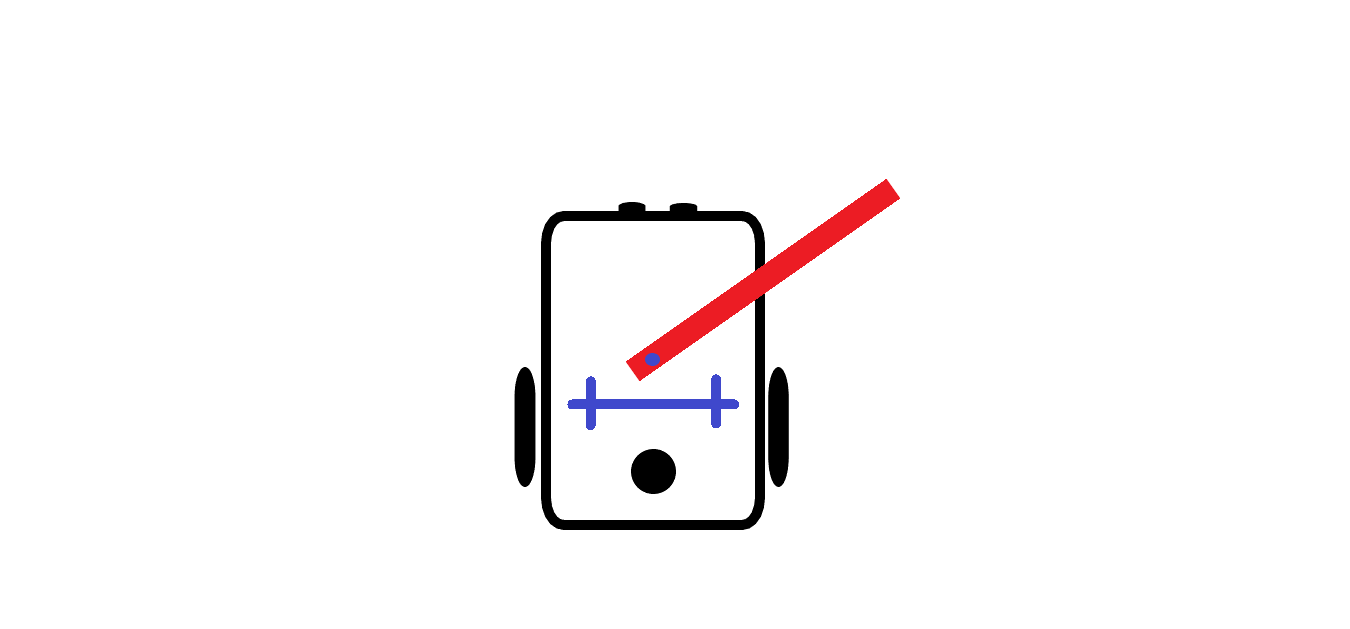
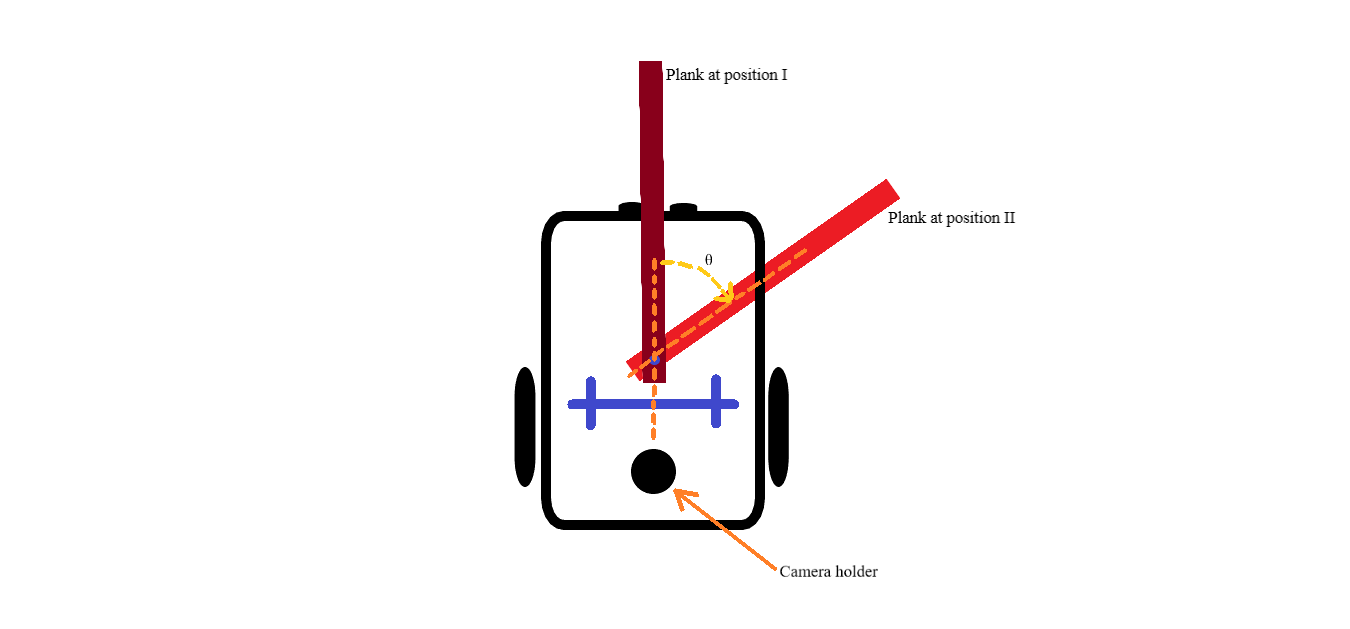
As stated in problem definition previously, the major issue in this project is to propose a method to overcome the lack of direct communication between the two robots, especially when one of them (mostly the master) is in situation of state change, such as entering a U or an L corner. Both active and passive methods should be developed to provide the indirect communication between the robots.

It has already been concluded in the Standards Committee Meetings that the robot should stop and wait for 10 seconds if it encounters an obstacle and is going to start moving towards left or right (entering a U or an L turn). This delay time provides the other robot (mostly the slave) the information that it is in the act of state transition, indirectly. However, this passive method is obviously not enough to provide all necessary information between the robots. As a result, we have developed two different solution methods due to redundancy.

### Solution I: Image Processing for Collaboration

As a solution, we came up with use of image processing methods. When the possible movements are considered, the only detection can be done by plank and the other robot’s position. From Standard Committee, color of plank is chosen as red. Top of the walls are white while the rest of the walls are white. Thus, the walls and the plank can be observed easily.

In this solution we are planning to sense the other robot’s movements by measuring the angle of the plank using image processing methods. This is method is shown in Figure 3.



(1)

(2)

Figure 3: Illustration of Angle Sensing via Image Processing Approach

The algorithm is as follows.

1. The frame is slowed down in order not to busy the processor. Probably it will be lowered to 10 frames per second.
2. Using the blue double crosses seen in Figure 3 calibrate the zero angle.
3. Extract the region that we are interested: the plank itself and the joint point.
4. Using Otsu Thresholding Method, binary-mask the image.
5. Find the plank’s direction using tracing the boundaries.
6. Find the line that corresponds to planks orientation.
7. Find the point of intersection of plank’s direction line and the blue calibration line.
8. Using the dot product, find the angle of intersection.
9. If the angle is larger than a previously determined angle value, then this means the other robot is in turn.
10. Initial angle is known and it 90⁰ to the blue calibration line. Subtracting the current angle, θ is determined.

As can be seen above, this method leans on a simple idea. This method can be improved by adding the integrating the wall detection algorithms so that robot can observe the other one directly.

### Solution II: Processing the Information Retrieved from Plank for Collaboration

Even if a solution with image processing seems accurate, it should always be kept in mind that after the implementation, a solution may not always work as accurately as planned or may not work at all. For that reason, a back-up solution is developed in order to sense and process the movements of the other robot. The only thing that links the two robots is the plank, hence data collected from it may be manipulated and used as in the previous solution.

In this solution, it is planned to mount an encoder with Hall effect sensor and a 3-axis gyroscope to the shaft of holding point of the plank.

Hall effect sensor is a sensor which creates a voltage difference between its output terminals as a result of change in magnetic field, due to Lorentz force. This working principle is clearly illustrated in Figure 4 below.

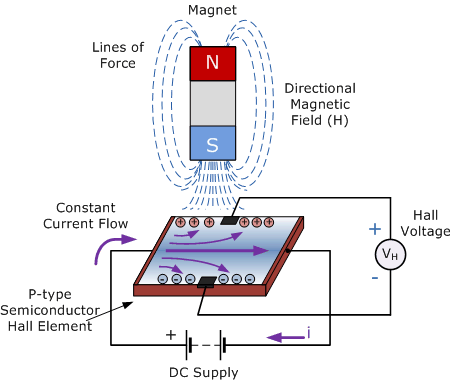


Figure 4: Illustration of Hall Effect

When such a sensor is mounted on an encoder, continuously changing magnetic field while the rotor of the motor or shaft of the holding point rotates creates continuous impulses at the output voltage. This phenomenon is also illustrated to clarify, in Figure 5 below.

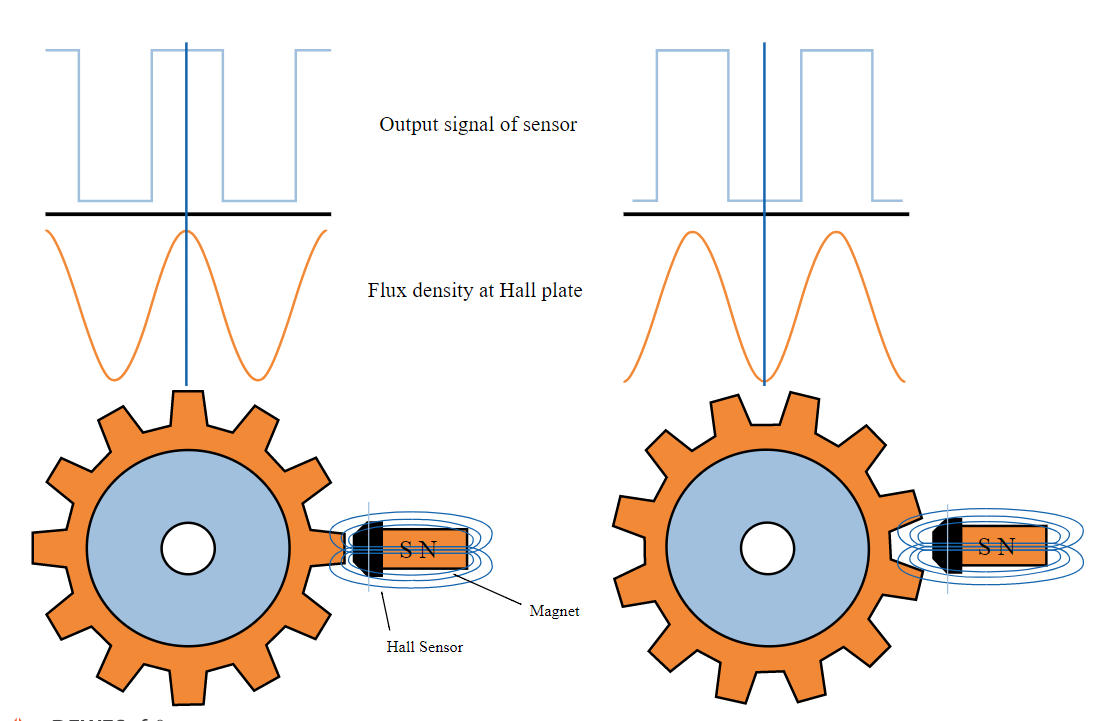


Figure 5: Output Voltage of Encoder with Hall Effect Sensor

Apparently, the rotation angle of the shaft can easily be found by fractioning the number of upcounts as the output of the encoder with Hall effect sensor, to number of upcounts corresponding to a full rotation of the encoder.

where θ is the rotation angle of the plank in degrees.

The main problem with this solution is that even if the rotation angle of the plank can be calculated very accurately, unfortunately it does not allow the robot to sense if this angle is positive or negative. In other words, it does not provide the information whether the collaborating robot moves towards left or towards right. Using a 3-axis gyroscope, this missing information can also be obtained. The acceleration of the shaft of the holding point causes a digital output, which is positive if the acceleration is in positive x, y, z direction or negative, if the acceleration is in negative x, y, z direction.

# PLANS

## MANAGEMENT

### Responsibilities Among the Group Members

As X-Cali, we are a group of electronics engineers working in different areas. Each group member is specialized in different topics and we distributed the technical workload regarding this fact. The distribution of technical work is as given in Table 1.

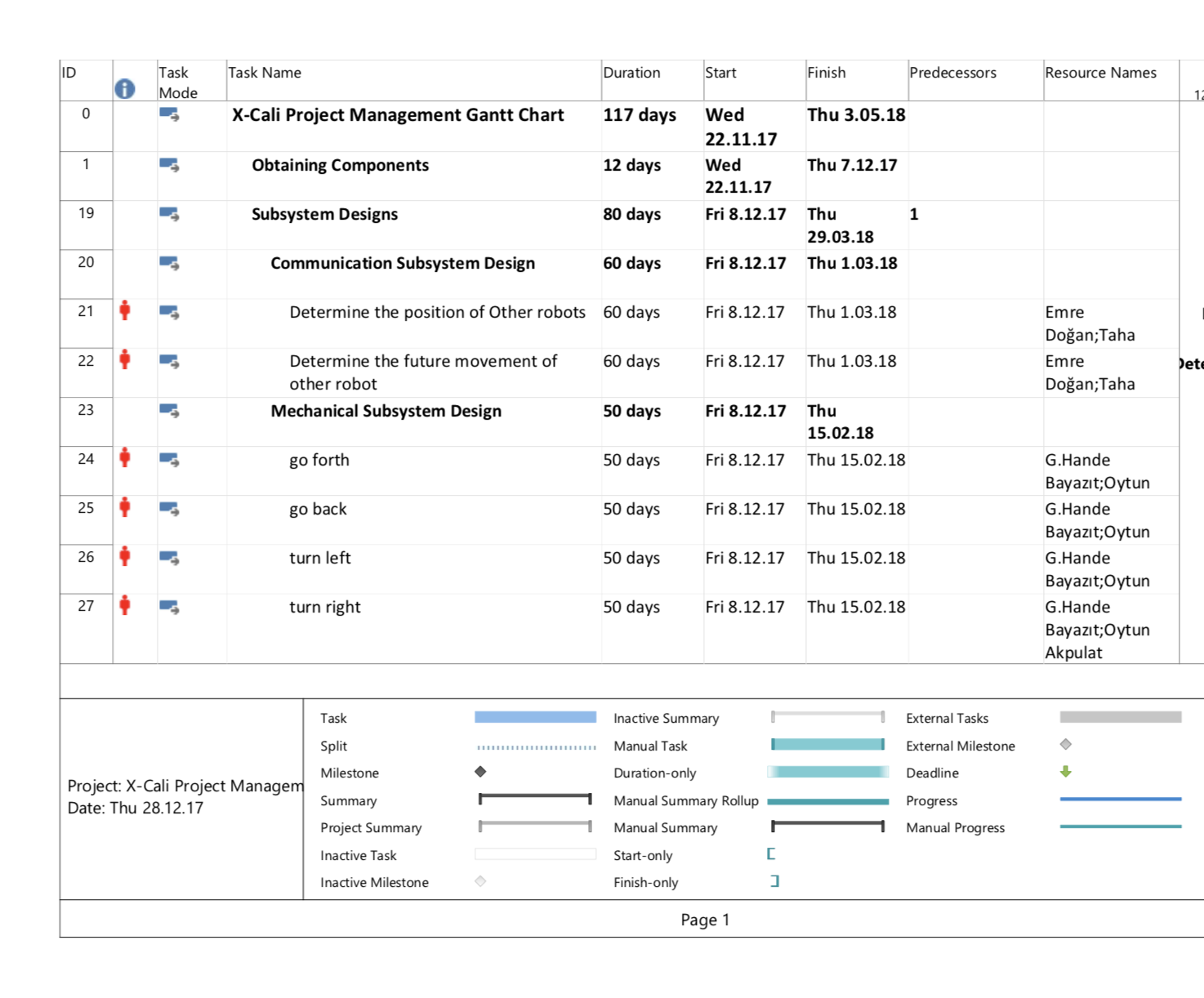
Table 1. Distribution of the Technical Workload in X-Cali

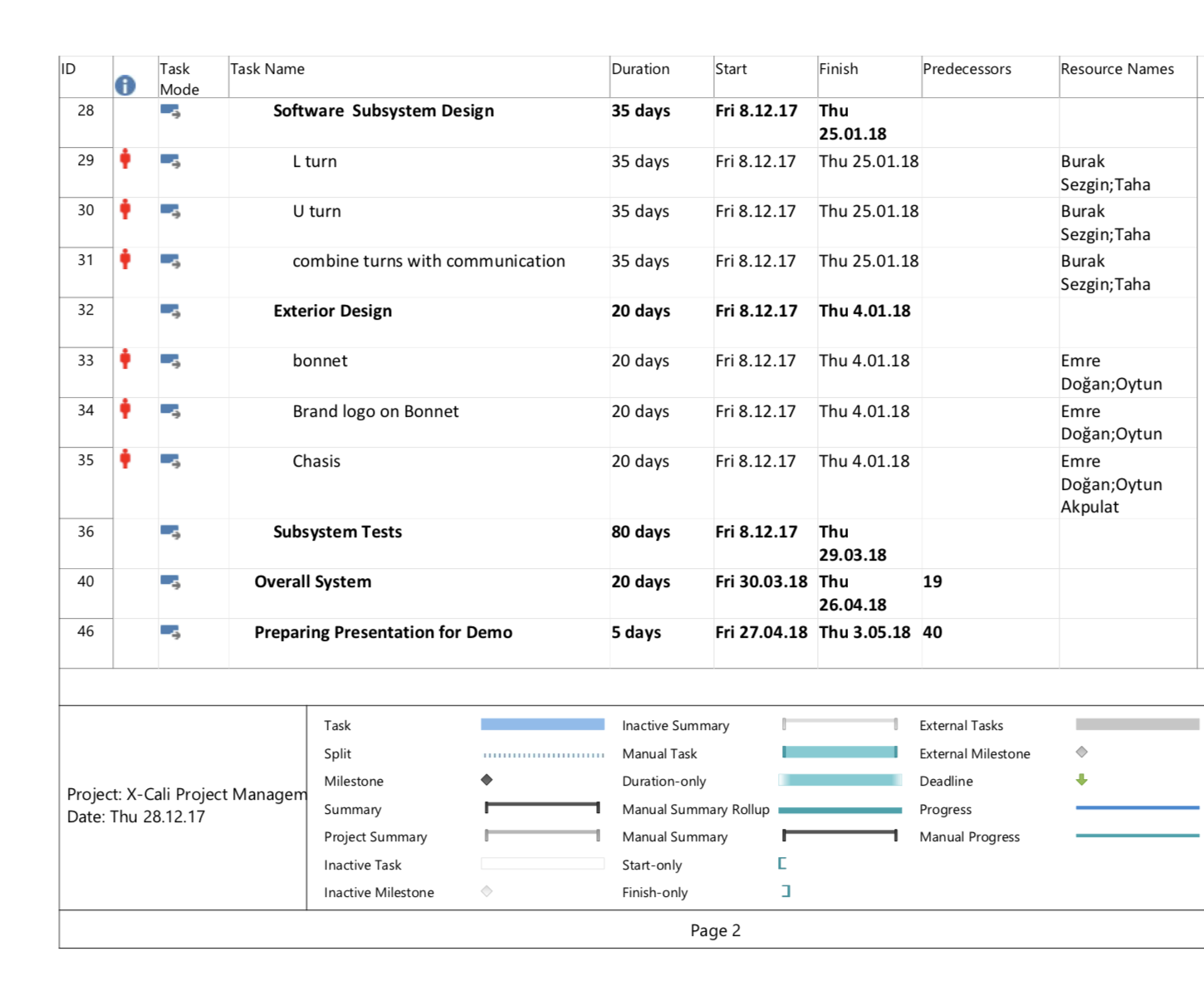
|  |  |
| --- | --- |
| The Subject | Member(s) in charge |
| Microprocessor Programming | Burak Sezgin  Taha Doğan |
| Algorithm Design | Taha Doğan  Emre Doğan |
| Sensors & Communication | Emre Doğan  Burak Sezgin |
| Control System Design | G. Hande Bayazıt  Oytun Akpulat |
| Mechanical System Design | G. Hande Bayazıt |

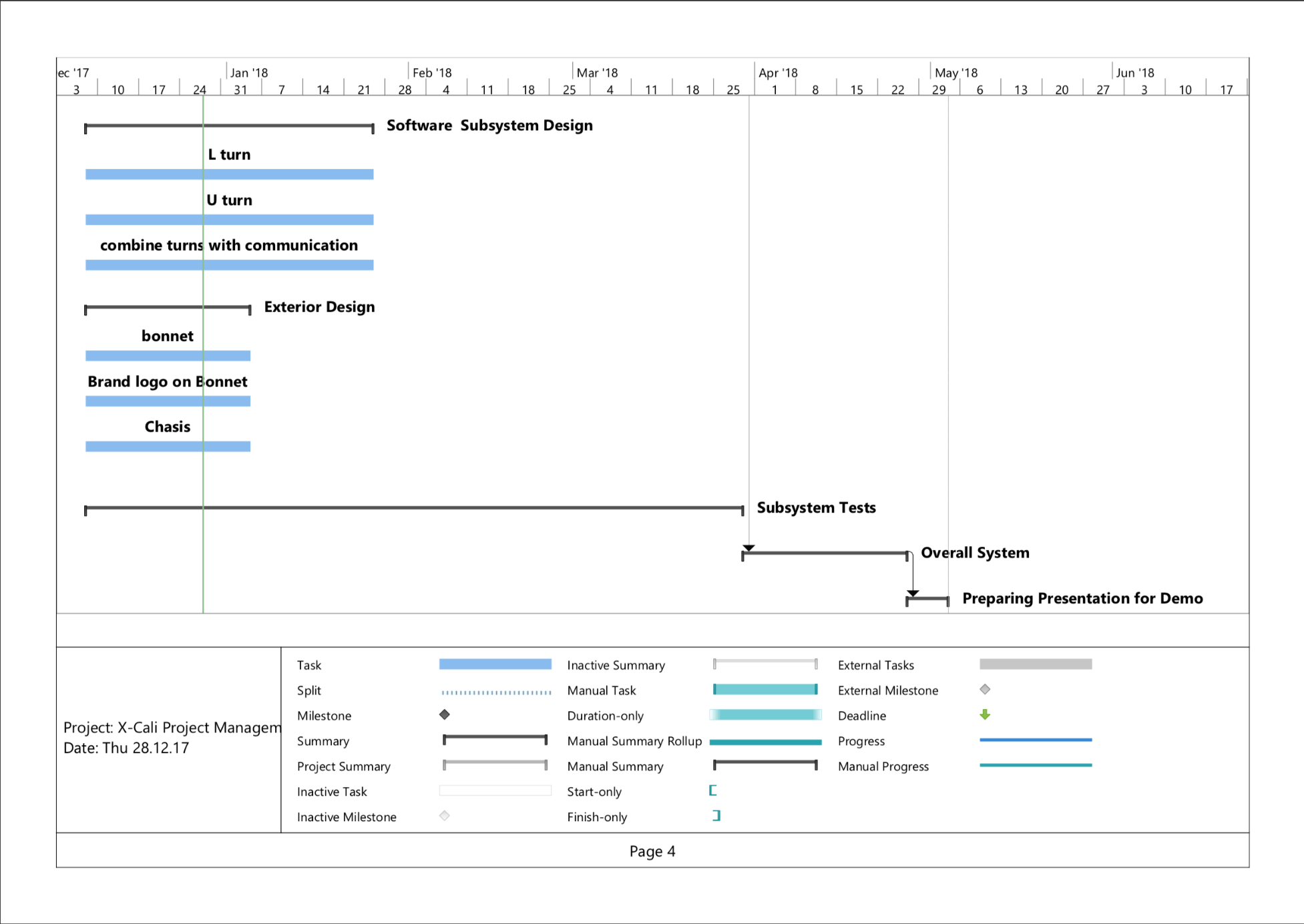
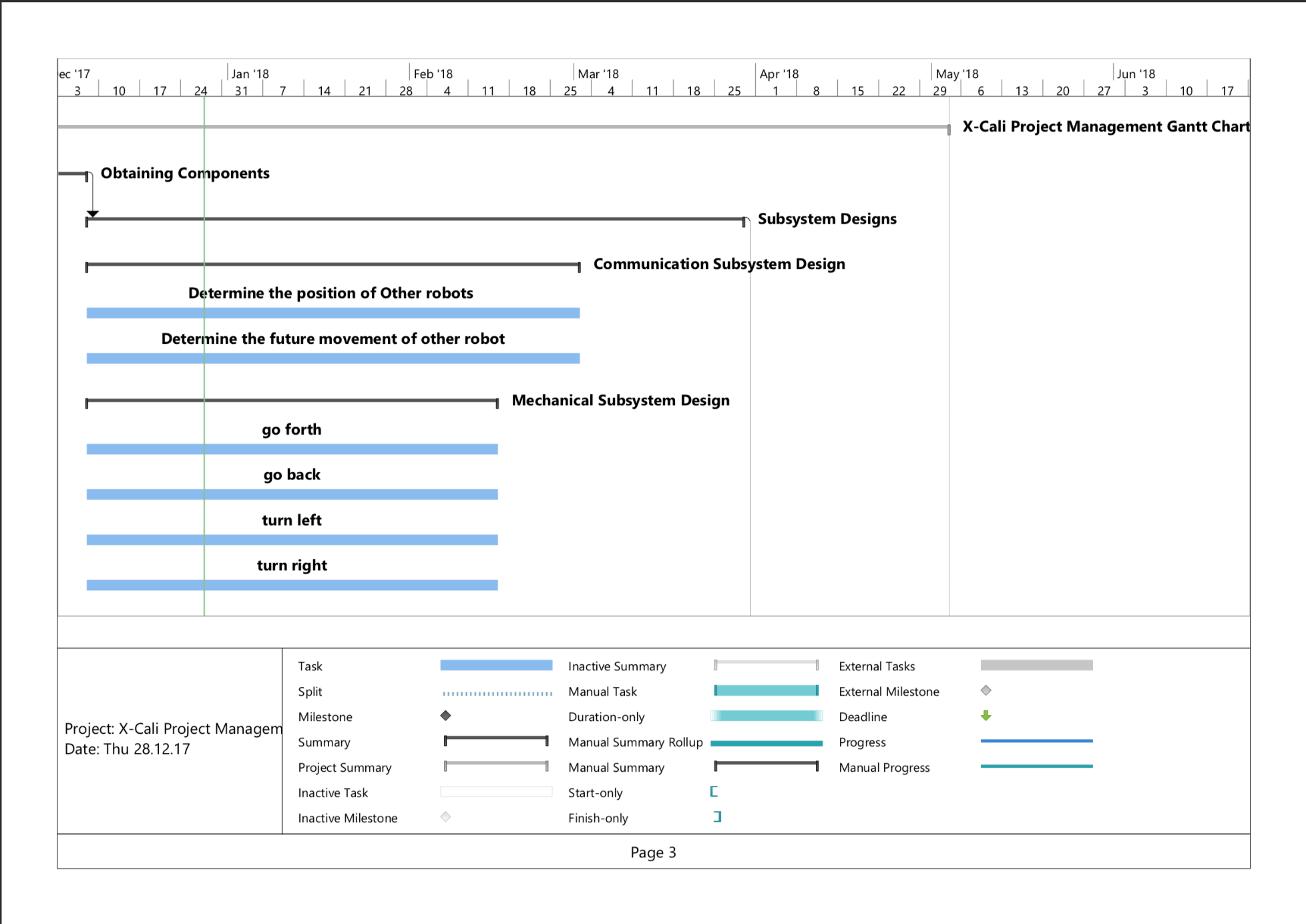
Although we separated the technical workload for different group members, we mostly work together in all tasks. This situation is quite important for us as we have a chance to gain the habit of interdisciplinary team work.

Beyond the technical issues, there are some other important management subjects to handle. Our financial control specialist is Taha Doğan. He is in charge of the all financial issues due to the project.

### Detailed Version of Gantt Chart

******

******

******

### Cost Analysis

After a detailed research process, we checked the market to be able figure 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 Per 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₺ | 4 | 91.00₺ |
| Logitech C270 Webcam | 115.75₺ | 1 | 115.75₺ |
| L298N Motor Driver | 12.00₺ | 1 | 12.00₺ |
| Ball transfer unit | 4.20₺ | 1 | 4.20₺ |
| Robot Chassis & Wheels | 50.00₺ | 1 | 50.00₺ |
| 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₺ |
|  |  | **TOTAL PRICE** | 607.75₺ |

It does not seem like that we will need other components. But if we need to add some new parts to our robot, we still have almost 40% of our project budget.

### Deliverables

The expected deliverables of the work packages of our project process can be observed in the Table 3.

Table 3: Expected Deliverables of the Work Packages.

|  |  |  |
| --- | --- | --- |
| The Work Package | Corresponding Deliverable | STATUS |
| Research | Tentative Report | Completed. |
| Component Tests | Results and analysis of the component tests | 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. |

Beyond the deliverables of the R&D process, the package of our product will include the main body of robot, a plank, user manual, 2 spare tires, a back-up battery and a remote controller deciding the robot to become master or slave.

The size of the robot is not strictly unalterable as the company will not have a stockpiling policy. After your order, your product will be prepared with respect to your requests and sent you in 10 weekdays.

Our product will have a low-power consumption so that charging will not be an issue for users. Rechargeable batteries with higher charging cycles will be preferred in the design process. But still, for the degeneration problems due to battery, we are sending a back-up battery within the package.

In the user manual, customers will be able to find all necessary information about the setup of the robot, methods of changing the tires and battery and switching the robot’s duty (master or slave). All this information and some extra contents will be available on our company website <http://www.xcali.ml/> . You can easily leave a message from the contact tab to ask anything about our company and products.

Our company cares about customer satisfaction. For this reason, we are offering a 2-year warranty for all products except the batteries. After 2 years, you can extend your existing warranty with a small amount of money. Our maintenance and repair service will be in Ankara. But you can send your broken product with our negotiated logistic partner, a well-known shipping company from all around the world.

## ENGINEERING

### Test Plans

As we completed two subsystems, Proximity Sensor Sensing and DC Motor Driving, we have three test plans, including these subsystems and integrated version of them, prepared up to now. These plans are given in the following order:

#### 1st Test Plan: Proximity Sensor Measurements

HC-SR04 model of ultrasonic proximity sensor is used to sense the maze walls. Before the integration of proximity sensors, they are tested. The test specifications are provided below.

|  |  |
| --- | --- |
| ***Environment*** | : Living Room, silent |
| ***Driver*** | : Raspberry Pi 3 |
| ***Obstacle Object*** | : a thick book |
| ***Temperature***  ***Measurement Tool*** | : 24℃  : Tape measure  N |

***Measurement Method***

First, the distance between the proximity sensor and the obstacle is set to specific values such as 5cm, 10cm… up to 75cm with a tape measure. Then the distance is measured with the proximity sensor and the values are saved. Five samples are collected at each distance value.

Measurement is done with RaspberryPi3 and the results are observed on the monitor, on the RP3 GUI.

The measurement results are given in Table 4.

Table 4: Measurement results of HC-SR04 model prox. sensor with given measurement specifications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Measurement #1 | Measurement #2 | Measurement #3 | Measurement #4 | Measurement #5 |
| Real Distance | Measured Distance | Measured Distance | Measured Distance | Measured Distance | Measured Distance |
| 5 | 5.9 | 5.4 | 5 | 5.2 | 5.2 |
| 10 | 9.7 | 9.67 | 9.74 | 9.74 | 9.71 |
| 15 | 15.2 | 15.16 | 15.16 | 15.5 | 15.18 |
| 20 | 19.62 | 19.58 | 19.62 | 19.58 | 19.6 |
| 25 | 24.56 | 24.56 | 24.64 | 24.59 | 24.61 |
| 30 | 29.15 | 29.7 | 29.15 | 29.57 | 29.14 |
| 35 | 34.12 | 34.21 | 34.16 | 34.12 | 34.12 |
| 40 | 39.41 | 39.44 | 39.32 | 39.42 | 39.39 |
| 45 | 43.92 | 43.88 | 43.88 | 43.93 | 43.9 |
| 50 | 49.7 | 48.82 | 48.88 | 48.15 | 48.22 |
| 55 | 53.71 | 53.68 | 53.64 | 53.68 | 53.59 |
| 60 | 58.92 | 58.93 | 59.36 | 59.77 | 58.92 |
| 65 | 63.18 | 63.6 | 63.5 | 63.56 | 63.43 |
| 70 | 68.53 | 68.1 | 68.18 | 68.48 | 68.12 |
| 75 | 73.74 | 74.19 | 74.6 | 73.76 | 73.28 |

HC-SR04 has 4 pins

* VCC
* TRIG
* ECHO
* GND

TRIG and ECHO pins are connected to the RP3 pins 23 and 24 respectively. The code, sense.py, for measurement is provided in our project’s GitHub Repository.

The following graphs are obtained using MATLAB.

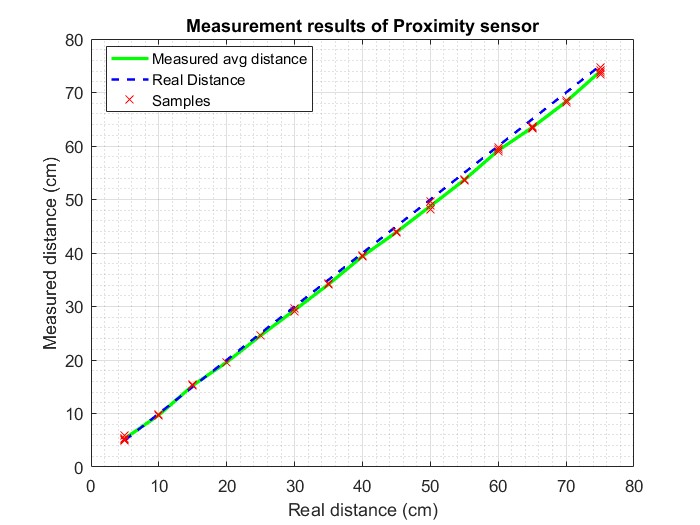


Figure XX: Measurement Results of HC-SR04 proximity sensor

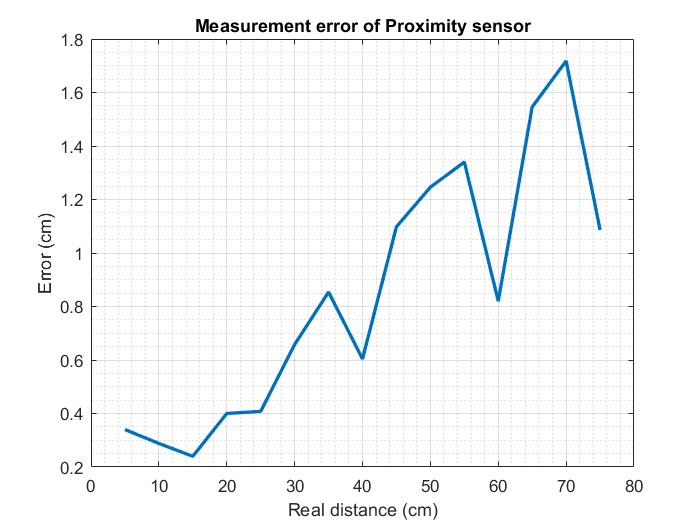


Figure XX: Measurement error of HC-SR04

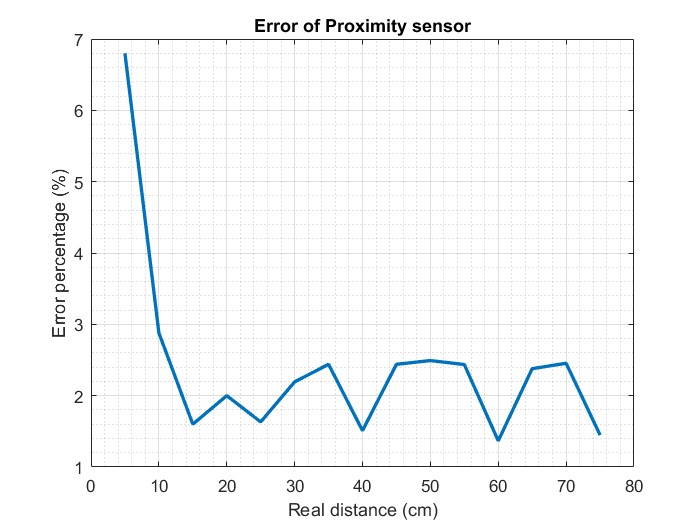


Figure XX: Error Percentage of HC-SR04

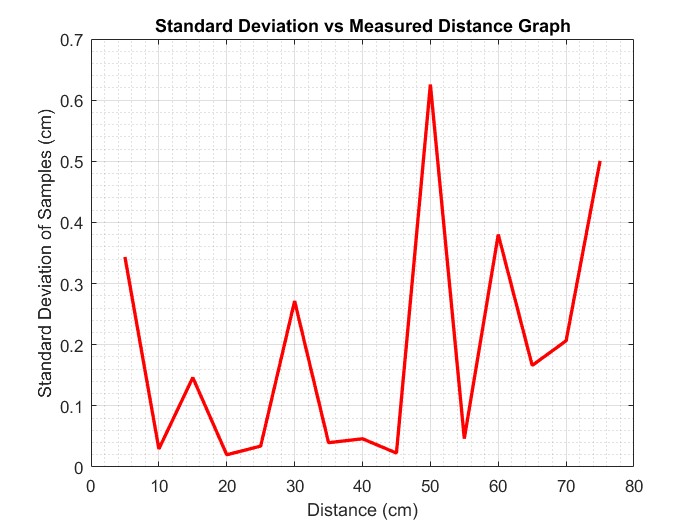


Figure XX: Standard Deviation of HC-SR04

***Result***

When the error percentages of the component at different distances are monitored, it is seen that up to 400 millimeters, there is not a significant error. But after this amount, the amount of error is more than 10 millimeters. With all these data, we decided that our robot will stop when the distance to the wall is 150 millimeters and make a decision at this point. So, this component is suitable for our system.

#### 2nd Test: DC Motor Starting Current Test

**Test Conditions**

***Location*** : Design Studio

**Date**  : Dec. 17, 2017

***Temperature*** : Room temperature (appx. 23-25◦ C)

***Environment***  : Loud, luminous

**Test Procedure**

The aim of this test is to measure the zero-speed current (maximum current) of the DC motor. To achieve this, we used the DC Power Supply’s Current Limiter property. We directly connected the power supply to the motor. And by fixing the voltage level, we changed the current limit value step by step. The corresponding current values can be seen in the following table.

Table 5: Current values corresponding to the different trials.

|  |  |  |  |
| --- | --- | --- | --- |
| Trial Number | Voltage Level | Current Level | Is the limit exceeded? |
| #1 | 7.8 V | 3.00 A | NO |
| #2 | 7.8 V | 2.74 A | NO |
| #3 | 7.8 V | 2.30 A | YES |
| #4 | 7.8 V | 2.60 A | NO |
| #5 | 7.8 V | 2.40 A | YES |
| #6 | 7.8 V | 2.50 A | NO |
| #7 | 7.8 V | 2.49 A | YES |

The DC power supply makes a clique sound and turns the CC (Control and Current) LED on when the current limit is exceeded. By this method, we found the maximum current level under the voltage level 7.8 Volts which is about 2.49 Amperes. Below this value, all current levels will exceed the limit during the zero- speed tests.

A similar test is applied while the robot is moving. Two motors took 0.92 Amperes in this test.

**Conclusion**

These tests are performed in order to observe and measure the current rating of the motors at worst case (i.e. at start and at zero-speed issues). Comparing the test results, it can be concluded that the amount current drawn by motors is far from being harmful for the battery, the motor drive and the rest of the system.

#### 3rd Test:  Response Time and Turning Angle Test

-Integration of Motors and Proximity Sensor-

**Test Conditions**

***Location*** : Design Studio

**Date**  : Dec. 17, 2017

***Temperature*** : Room temperature (appx. 23-25◦ C)

***Environment***  : Loud, luminous

***Measure*** : Turning time and turning angle

***Measurement Tools***: Protractor, ruler, pencil

**Test Procedure**

This test aims to observe the behavior of integration of two subsystems, which are motors and proximity sensor.

In the previous tests, we have observed successive results. In this test, the turn angle corresponding to different run times of the code that performs turning to right. Two subsystems integrated on Raspberry Pi 3. The scenario is that the robot continues on its route until it encounters an obstacle, in fact in our case a wall, and turns right first due to Standard Committee Regulations.

The test is performed as follows: The run time of the code is set to different values in order to observe corresponding turning angles of the tyres. Starting point of the vehicle, the point that the vehicle senses the obstacle and stops, the point after the vehicle turns right and stops and the final point that the vehicle stops are marked. These four points form two approximately perpendicular lines and the angle between them is measured with a protractor. This setup is shown in Figure X, below.

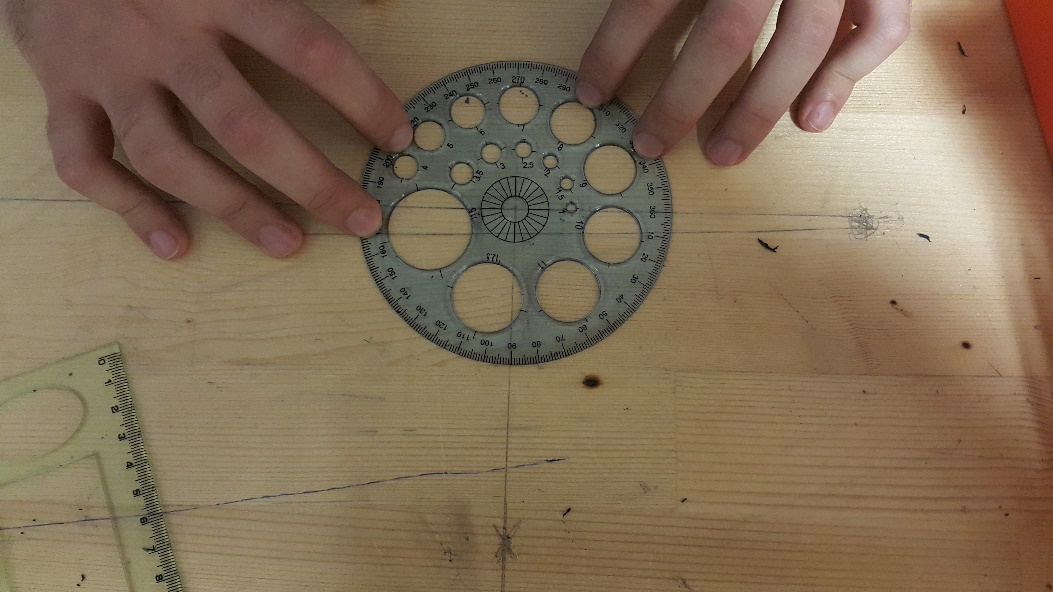


Figure XX: Experimental Setup

**Test Results**

Specifications

*Power Supply to motors, constant:* 8 Volts

*Controller:* RP3

*Motor drive:* L298P Shield

Test results are provided on Table 1, below.

Table XX: Turn Angles Corresponding to Different Right-Turn Run Times

|  |  |  |  |
| --- | --- | --- | --- |
| **Run Time** | 0.5 sec | 0.4 sec | 0.45 sec |
| **Trial 1** | 104 o | 85o | 88o |
| **Trial 2** | 93 o | 86 o | 83 o |
| **Trial 3** | 103 o | 80 o | 90 o |

**Conclusion**

These tests are performed in order to observe integrated system response. As can be seen above, proximity sensor reacts very quickly. The time required to turn robot approximately 90⁰ takes around 0.45 sec. However, please note that there is not neither gearbox in motor setup nor speed control software. Thus, motors are run at top speed at given voltage.

# CONCLUSION

# REFERENCES

# APPENDICES