# An-Najah National University

**Faculty of Engineering and Information Technology**

**Computer Engineering Department**

**Graduation Project II**

# "RoCu "

**Prepared by:**

**Aya Abu Ali & Waseem Ghazal**

**Supervised by:**

**Dr. Emad Natsheh**

**Presented in partial fulfillment of the requirements for bachelor's degree in computer engineering.**

# Acknowledgment:

First and foremost, we give thanks to God for blessing us and for using us to carry out this endeavor. We also pray that the knowledge we gain will be useful to the rest of the world, especially in Palestine.

We would like to thank our families for the continuous support and encouragement they provided us from the beginning until the last moment.

We thank Dr. Emad Al-Natsheh for his constant help and for providing us with a lot to complete this project.

We thank our dear Drs in the Department of Computer Engineering, who had the credit for us by reaching this stage with all that they have given us throughout these years, we will forever be grateful.

To all those and many more, who supported us through thick and thin, we give a big “Thank You” straight from our hearts, and we present you this report and project.

# Disclaimer

This report was written by Aya Abu Ali and Waseem Ghazal at the Computer Engineering Department, Faculty of Engineering, An-Najah National University.

It has not been altered or corrected, other than editorial corrections, because of the assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was Commissioned.

# Contents

# Table of Figure

## Abstract

The Rubik cube is a puzzle consisting of a plastic cube covered in multicolored squares that the player must twist and spin for all the squares on each face to be the same color. This puzzle requires high intellect and may take a lot of time to solve, so our project will be a robot that will solve it smartly and quickly. This robot will be able to solve the puzzle from any starting point by taking a picture of the six faces and recognizing the colors using image processing code, then passing it to the solving algorithm “Kociemba Algorithm” to identify the solving process and sending the moving commands to the motors via the Arduino Mega microcontroller.

A 3D printer was used to create the robot's construction in order to avoid the challenges of the mechanical side and to ensure that it measurements are accurate.

Features the application included:

1- The ability to read all 6 faces of the cube.

2- The cube can be solved by the number of moves the user enters.

3- Provide a built-in timer.

4- A sign buzzer to show that the cube has been solved.

# 1 Introduction

## 1.1 The problem:

Stumbling upon a difficult problem to solve is inevitable. A difficult problem is defined as a problem in which the solution is not easily discovered by a typical person. These problems can be like recreational puzzles such as a game of chess or a Rubik's Cube. Due to their complexity, and in some cases lack of information on the subject, it is easy to get frustrated and not try when one encounters such a problem. For this reason, it is imperative to have a skill set that one can utilize in these situations.

It will be simpler to avoid the frustration trap and carry on searching for a solution if there is a method to follow.

## 1.2 Significance:

The main objective of the project is to build a robot capable of solving Rubik's cube in a short time without trying to move the upper face of the cube.

The project contributes by providing help for players in solving Rubik's cube, in addition, "RoCu" serves the field of entertainment.

## 1.3 Objectives and Scope

The purpose of the smart robot "RoCu" is to solve Rubik's cube, and therefore it required the following services to be available:

1. The ability to read the faces of the cube and is done through the processing of digital images.

2. Artificial intelligence to figure out the best solution moves.

3. The robot's ability to move the cube faces 90 or 180 degrees.

4. Making a sign when the cube has been solved.

## Report Organization

• Second chapter: In this chapter, we covered the important subjects we learned previously, as well as the external courses and the primary constraints and obstacles we encounter while working on the project.

• Third chapter: The literature review chapter is a summary of previously published works that are similar to ours in concept. we discuss their qualities and weaknesses, as well as what sets us apart from them.

• Fourth chapter (Methodology): We talked about the mentality through which we built the application, in addition to the features that we offer, and the technology used.

• Fifth and final chapter: we talked about the results, as well as the lessons we learned from working on the project and future developments.

# 2. Constraints and Earlier Coursework

## 2.1 Constraints

### 2.1.1 Uncertainty of Image Processing

The application requires a good understanding of image processing which even when present may not give the required output, lighting and various other changes affect the process.

### 2.1.2 Inexperience

We made different simple demos and tests to make sure that the concept which we are aiming for is achievable. But all these tests took time which we could have invested in other things.

Even after all the tests, research and time spent, we still made some decisions which are considered safe bets because we could not test certain parts, and we could not afford the whole project failing.

### 2.1.3 Lack of funds

Some solutions needed more money to make, which would make the project too expensive and unaffordable, especially at small scale.

### 2.1.4 Lack of Time

Time is the most important factor in any project. Researching, communicating, testing, developing, understanding the requirements, planning, and the wasted time due to inexperience, all of these things require time which, no matter how tiny, quickly adds up.

### 2.1.5 Lack of Mechanical knowledge

The project has a lot of mechanical parts which we partially figured out but even after all the research, an experienced mechanical engineer was needed to answer some questions and plan the execution properly.

Some parts which are 3D printed needed some balancing and mechanical knowledge to fix, so the ideas and the initial designs were presented to the engineer to modify.

### 2.1.6 Lack of Tooling

Some special tools were needed to cut, prepare, and assemble all the parts which are not available or too expensive for home use, so a specialist was required to help with it.

Some parts required a 3D printing machine to print.

### 2.1.7 Lack of Market Knowledge

Some parts are not just store bought but are custom-built to a given specification, so more research was needed and more planning to get the correct specifications to give a correct order.

## 2.2 Earlier coursework

### 2.2.1 Digital Image Processing

The course was a big part of the project, as almost all the steps taken to handle the images to get the cube colors were from this course, let alone the important lifelong concepts that were taught.

### 2.2.2 Microcontrollers and PIC

All the basics of the Raspberry and Arduino like basic serial communication were taken in these courses.

### 2.2.3 AI

The main part of the project is the brain which takes the cube faces with its colors, an algorithm was taken to solve the cube with few possible moves and evaluate them to pick the best move.

### 2.2.4 Operating Systems

We used the Raspberry Pi OS, which we flashed on the Raspberry and used the Linux command line to install libraries and packages needed for the project. All those are some things discussed in the course.

### 2.2.5 Networks and Communication

We made the Raspberry have a static IP that connects to the hotspot of one of our laptops that shared the Wi-Fi connection with it, and we connected the two laptops to the same network and accessed the

Raspberry through a ssh connection all that couldn’t be possible without the Networks and Communication courses.

### 2.2.6 Critical Thinking and Research Skills

The research and the writing of this report were all taught in this course, and it’s one of the few non-technical courses which also is lifelong.

# 3. Literature Review

The Rubik's Cube is a well-known mechanical puzzle that has captured the interest of people all over the world due to its unique features. As a well-known and classic brain-training game.

Since its appearance in the 1970s, the Rubik's Cube has quickly become one of the most popular and difficult puzzles of all time. It has perplexed even the brightest minds. Even Erno Rubik, the professor who invented it, believed that it would be impossible to solve.

However, many algorithms for solving the Rubik's Cube have been discovered over the years and learning how to solve the Rubik's Cube is now simply a matter of following a set of steps and practicing some algorithms.

"RoCu" was designed so that people could solve the Rubik's Cube without needing to learn and remember these methods. If you have an old, scrambled cube lying in your room, if you're trying to understand how to solve it on your own and just need a "reset", or if you just want to impress your friends, "RoCu" is perfect for you.

We developed a Rubik's Cube solving robot that implements kociemba algorithm on a Raspberry Pi 4. The Raspberry Pi work with Pi camera to sequentially observe each face of the scrambled Rubik's Cube.

We implemented a mechanism for converting this raw data into a representation of the puzzle's scrambled state. We then implemented the algorithm that takes the scrambled state of the Rubik's Cube as input and outputs a sequence of moves that solve the Rubik's Cube.

This output is converted into a set of commands for five stepper motors, which interact with the Rubik's Cube. Each stepper motor, which is controlled by stepper motor drivers connected to the Arduino Mega, rotates a cube face. The stepper motors execute the steps generated by the algorithm, solving the puzzle at the end.

# 4. Methodology

This section provides detailed information about the methods and techniques we used to develop the robot, starting from designing and assembling the mechanical structure to image processing walking through AI and how they are linked together to reach the final product.

## 4.1 Mechanical Part

The mechanical part of the project is the part responsible for moving the cube faces, and this requires the presence of motors to generate movement in cooperation with a controller to direct it. To do this, it requires performing the following steps:

### 4.1.1 Mechanical Design

The first step is to design the structure that will Provides stability to the stepper motors movement and provide accurate and stable information about how they move and degrees. we found during the design stage that there was a need to print pieces (3D Printing) and we got help from specialists to print them.

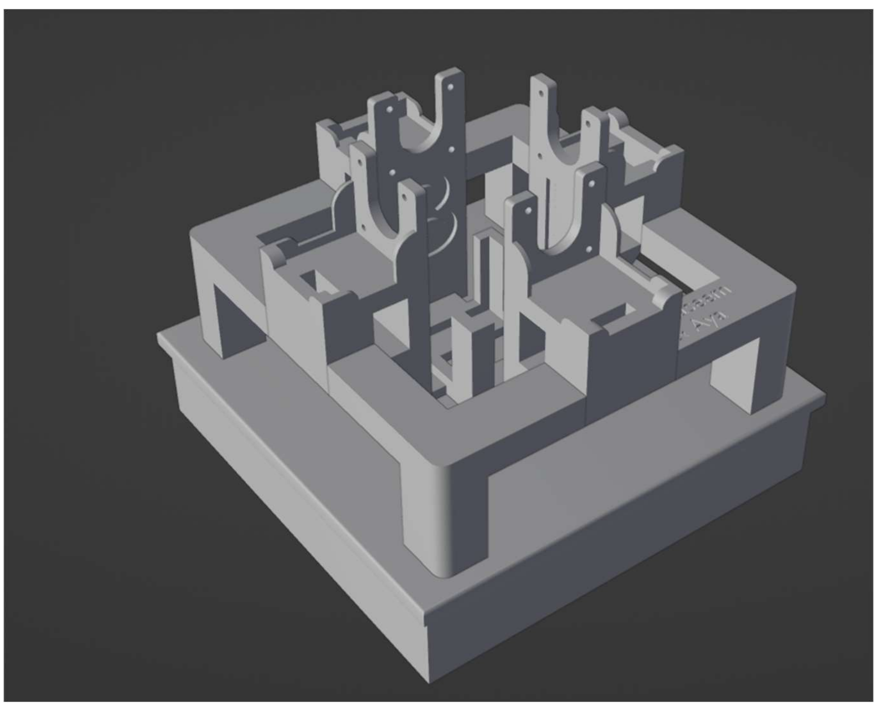


*Figure 1 : Mechanical Design*

### 4.1.2 Assembling

It is the part that requires the use of screws and screwdrivers to assemble the pieces together, and this must be done accurately for the project to work as required.

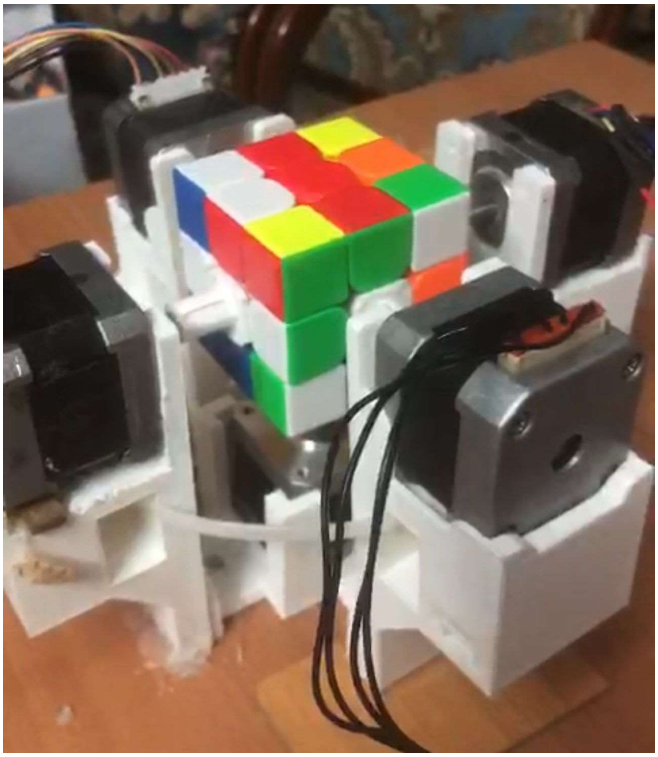
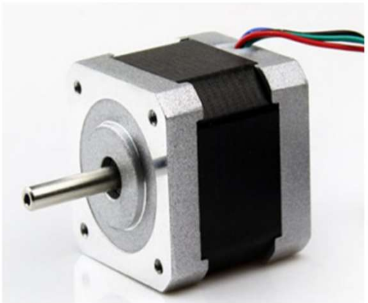
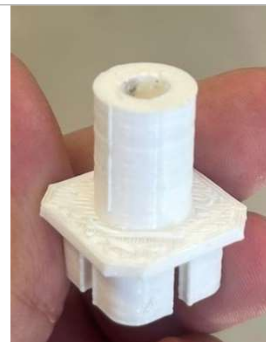
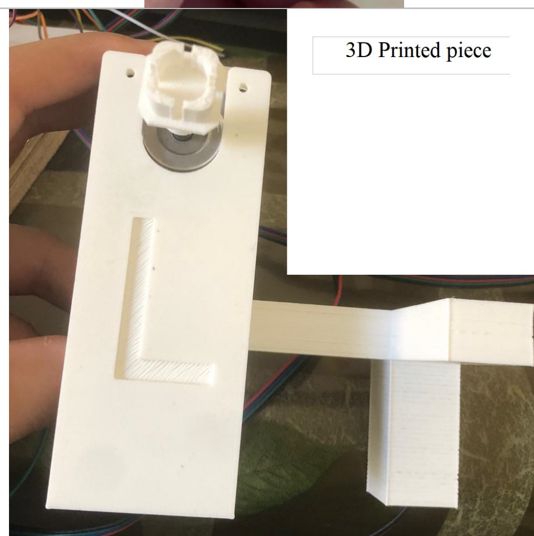
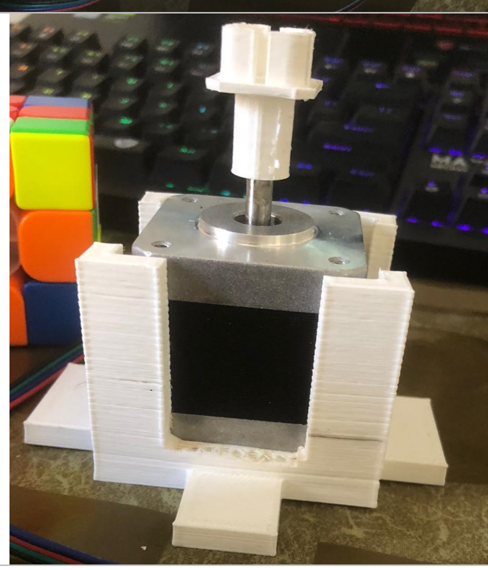


*Figure 2 : Mechanical Body*

At this point, we also tested the stability of motors to make sure that everything was moving correctly.

### 4.1.3 Parts

| Item Name | Item Image | Quantity |
| --- | --- | --- |
| Stepper Motor |  | 5 |

| | | |
|---|---|---|
| 3D Printed piece |  | 5 |
| 3D Printed piece |   | 4 |
| 3D Printed piece |  | 1 |

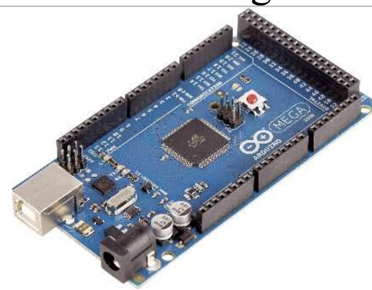
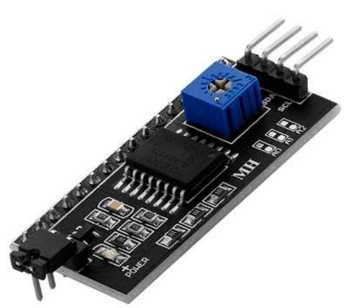
## 4.2 Controller Part

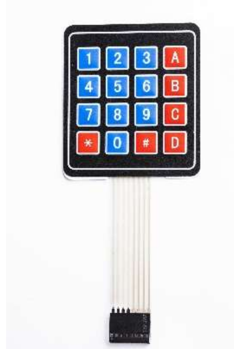
The controller part is responsible for directing the mechanical part and providing it with power, and it includes two main parts:

### 4.2.1 Arduino

We could have used the raspberry as a complete controller, but we separated the motors for two reasons, one of which is safety and the other is the separation of responsibilities so that the team can work separately.

#### 4.2.1.1 Parts

| Item Name | Item Image | Quantity |
| --- | --- | --- |
| Arduino Mega |  | 1 |
| A4988 Driver |  | 3 |
| L298N Motor Driver |  | 2 |
| LCD Display 16*2 |  | 1 |
| I2C LCD Adapter |  | 1 |

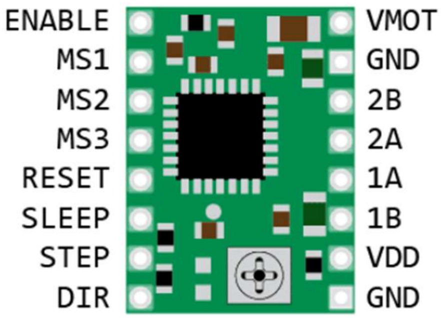
| | | |
|---|---|---|
| Keypad |  | 1 |
| Button |  | 3 |
| Buzzer |  | 2 |

**4.2.1.2 Driving Motors**

We have 3 motors are controlled using the A4988 driver, which is an inexpensive and readily available microcontroller that offers many features.

Stepper motors have 1.8° steps meaning 200 steps per revolution, referring to full steps. Higher resolutions are possible with a micro-stepping driver, such as the A4988, because intermediate step locations are supported. This is accomplished by supplying intermediate current levels to the coils.

To control the motor, we give it two input signals, the direction indicating the spinning direction and the step pulse which drives the motor to rotate.



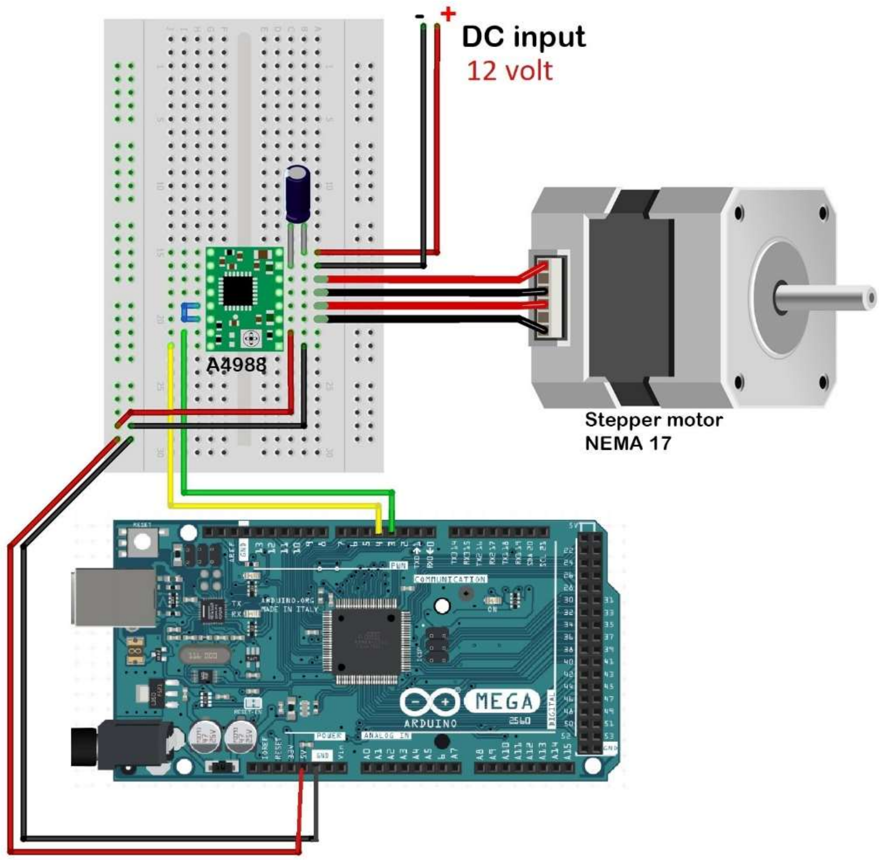




*Figure 3 : A4988 Driver Circuit*

We controlled the other stepper motors with L298N motor driver, the L298N module has two H-Bridges. Each H-bridge drives one of the electromagnetic coils of a stepper motor.

By energizing these electromagnetic coils in a specific sequence, the shaft of the stepper can be moved forward or backward precisely in small steps.

However, the speed of the motor is determined by how frequently these coils are energized.

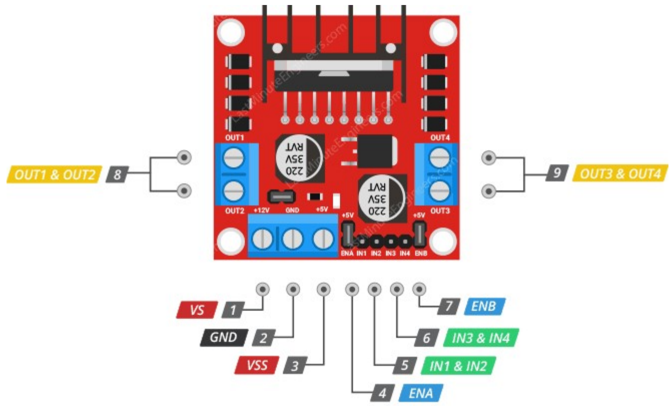



*Figure 4: L298N Module Pinout*

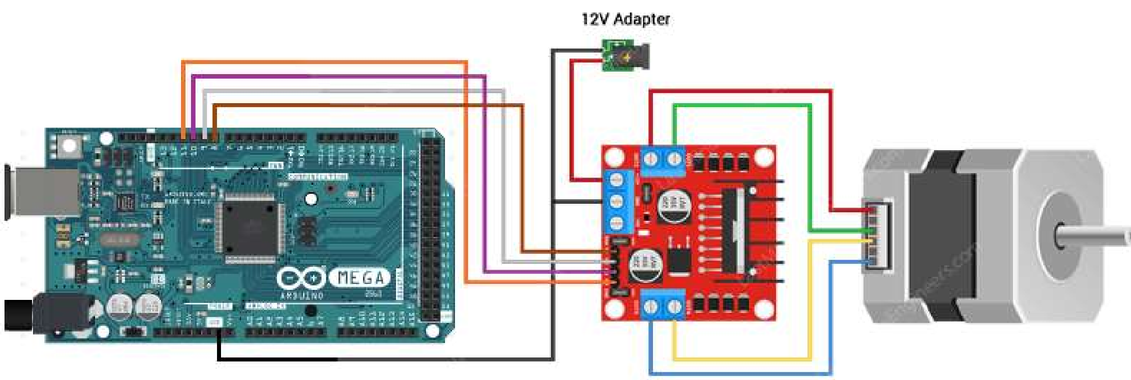



*Figure 5 : Control Stepper Motor with L298N Motor Driver*

## 4.2.2 Raspberry Pi

We used Raspberry Pi 4 for image processing, as well as writing the AI code in Python. The Raspberry has 8 megabytes of RAM and Linux 10 as operating system.

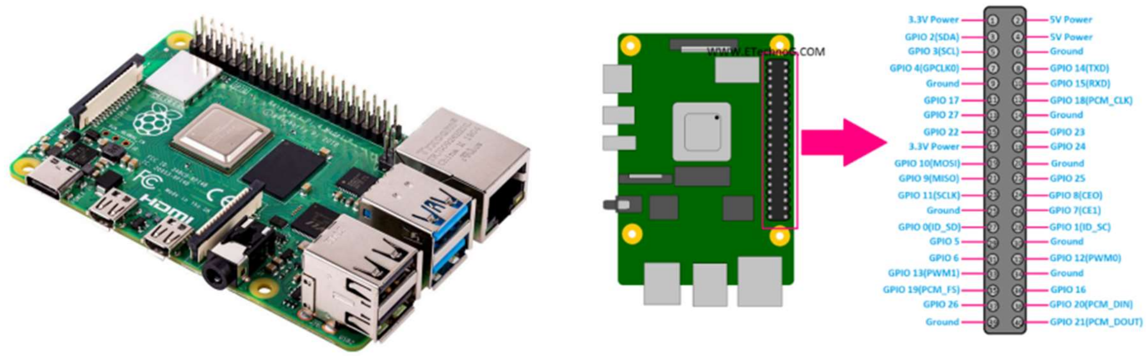



*Figure 6: Raspberry Pi 4*

#### 4.2.2.1 Image Processing

The goal of the digital image processing part is to know the colors of the cube faces to be transferred to the AI, where it responds to it and directs the motors to move the faces in the right way to solve the cube.

The program takes 6 pictures of the cube at the beginning, and then in each picture it searches for the squares colors to separate them from what is around it, and by using range functions the Colors are detected in the image.

#### 4.2.2.2 AI

The AI used in this project uses kociemba algorithm that its implementations use IDA*, which combines a heuristic with iterative deepening. The method attempts to follow a path that appears to have a shorter solution rather than just randomly walking through all the possible solutions. This reduces the need for more searching by assisting in the earlier discovery of shorter solutions.

Understanding how a Rubik's Cube works are essential to its solution, there are 6 sides to a Rubik's Cube, and each one has 9 pieces. The sides can be rotated in many ways, but the centerpieces cannot be moved while the cube is being solved.

The algorithm exposes just one function solve(), which accepts a cube definition string and returns a solution string.

```
            |************|
            |*U1**U2**U3*|
            |************|
            |*U4**U5**U6*|
            |************|
            |*U7**U8**U9*|
            |************|
************|************|************|************
*L1**L2**L3*|*F1**F2**F3*|*R1**R2**R3*|*B1**B2**B3*
************|************|************|************
*L4**L5**L6*|*F4**F5**F6*|*R4**R5**R6*|*B4**B5**B6*
************|************|************|************
*L7**L8**L9*|*F7**F8**F9*|*R7**R8**R9*|*B7**B8**B9*
************|************|************|************
            |************|
            |*D1**D2**D3*|
            |************|
            |*D4**D5**D6*|
            |************|
            |*D7**D8**D9*|
            |************|
```

*Figure 7: Names of the cube's facelet positions*

Letters stand for Up, Left, Front, Right, Back, and Down
A cube definition string "DFR..." means that in position U1 we have the D-color, in position U2 we have the F-color, in position U3 we have the R color, etc.

A solved cube is defined as follows:

UUUUUUUUURRRRRRRRRFFFFFFFFF
DDDDDDDDDLLLLLLLLLBBBBBBBBB

Solution string consists of space-separated parts, each of them represents a single move:

- A single letter means to turn that face clockwise 90 degrees.
- A letter followed by an apostrophe means to turn that face counterclockwise 90 degrees.
- A letter with the number 2 after it means to turn that face 180 degrees.

e.g., L U B’ D R U2 R’ B2
The algorithm returns a solution string then sends the solution moves to the motors so they could move the cube faces.

# 5. Results and Discussions

## 5.1 Movement

The initial goal we set to complete was to give the player the impression that the robot is smart, and it can solve the cube, which was achieved wonderfully, the feeling the player gets when the cube have been solved is indescribable.

## 5.2 Ai

The Ai is a simple solution, but we sadly could not build from scratch as it requires extensive math knowledge and a lot of testing to catch all the bugs and our math and knowledge are sadly not up to bar.

## 5.3 Image Processing

The image processing was not 100% accurate at some times, that reads some colors as white color because of the dim lighting, but the problem was solved by adding a light next to the camera.

## 5.4 Final Results

The result is a great project that works well in most cases that we tested and achieves all the goals that we set out to achieve.

# 6. Conclusion

## 6.1 Summery

We completed built a working smart robot that can easily solve Rubik's cube in a short time with smart moves, the robot might require certain improvements, which we were unable to carry out due to a lack of time, resources, or even knowledge.

## 6.2 Improvements

Some improvements are needed in the image processing section as it is not 100% accurate all the time, making the AI fail.
Tidy up the 3D structure, make big holes to wire everything nicely

### 6.3 Future Work

### 6.3.1 Make it faster

We are currently using the slow version but precise option in the motors as a proof of concept, but we could make it faster but less precise, and that would require some various adjustments.

## 6.4 Outcome

We made a smart robot that works well, not reliant on a smart device nor on a connection, the player can use it to solve some moves or whole moves for the cube.

# Bibliography


Overview of Rubik’s Cube and Reflections on Its Application in Mechanism (2018) Retrieved from https://cjme.springeropen.com/articles/10.1186/s10033-018-0269-7

Rubik’s Cube Solving Robot (2017) Retrieved from http://web.mit.edu/6.111/www/f2017/projects/jeswiezy_Project_Final_Report.pdf

Kociemba – PyPI (2019) Retrieved from https://pypi.org/project/kociemba/

Interface an I2C LCD with Arduino Retrieved from https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/

Control Stepper Motor with L298N Motor Driver & Arduino Retrieved from https://lastminuteengineers.com/stepper-motor-l298n-arduino-tutorial/

arul's utilities Retrieved from https://aruljohn.com/blog/python-raspberrypi/

The Robotics Back-End Retrieved from https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/#Serial_via_USB