***Arab American University***

***Faculty of Engineering and Information Technology***

***Computer Systems Engineering***

**SENIOR PROJECT I**

*Cube Mind   
(Rubik's Cube solver)*

*2025/2026*

|  |
| --- |
| Supervised by: *Dr. Tareq Zanoon* |

|  |
| --- |
| **Computer Systems Engineering Dept.**  **Submitted in partial fulfillment of the requirements of B.Sc. Degree in Computer Systems Engineering**  ***Insert Date Here*** |

**Students Statement**

We, the undersigned students, certify and confirm that the work submitted in this project report is entirely our own and has not been copied from any other source. Any material that has been used from other sources has been properly cited and acknowledged in the report.

We are fully aware that any copying or improper citation of references/sources used in this report will be considered plagiarism, which is a clear violation of the Code of Ethics of the Arab American University.

|  |  |
| --- | --- |
| Mohammad Atiyeh (202112724) | Ahmad khaldy (202110946) |
| Loai Hamamdeh (201811792) |  |

**Supervisor Certification**

This to certify that the work presented in this senior year project manuscript was carried out under my supervision, which is entitled:

“**SDP Title Goes Here**”

|  |  |
| --- | --- |
| *Mohammad Atiyeh (202112724)* | *Ahmad khaldy (202110946)* |
| *Loai Hamamdeh (201811792)* |  |

I hereby that the aforementioned students have successfully finished their senior year project and by submitting this report they have fulfilled in partial the requirements of B.Sc. Degree in Engineering.

I also, hereby that I have **read, reviewed and corrected the technical content** of this report and I believe that it is adequate in scope, quality and content and it is in alignment with the ABET requirements and the department guidelines.

*Dr. Tareq Zanoon,*

**ACKNOWLEDGMENT**

We would like to express our sincere gratitude and appreciation to all those who contributed to the successful completion of this senior project.

First and foremost, we extend our deepest thanks to Dr. Tareq Zanon, our project supervisor, for his invaluable guidance, continuous support, and expert advice throughout the development of this project. His mentorship and constructive feedback were instrumental in shaping our work and ensuring its successful completion.

We are grateful to the Arab American University and the Computer System Engineering Department for providing us with the necessary academic environment, resources, and facilities that enabled us to pursue this challenging and rewarding project.

We would like to acknowledge the collaborative efforts of our team members: Mohammad Atteyah, Ahmad Khaldy, and Loay Hamamdi. Their dedication, technical expertise, and teamwork were essential to overcoming the various challenges encountered during the development of our Rubik's cube solving robot.

We extend our heartfelt appreciation to our parents for their unwavering support, encouragement, and patience throughout our academic journey. Their belief in our abilities and their continuous motivation have been a source of strength and inspiration.

Finally, we thank everyone who directly or indirectly contributed to the success of this project. Their support and encouragement have made this achievement possible.

**ABSTRACT**

This project presents the design and implementation of an autonomous robot capable of solving a 3×3 Rubik's cube using Kociemba's algorithm. The primary objective of this work is to integrate and apply the knowledge and skills acquired throughout our Computer System Engineering studies while exploring new technical competencies in robotics, computer vision, and algorithm implementation.

The system consists of a Raspberry Pi as the main processing unit, coupled with cameras for cube state detection, motors for mechanical manipulation, and supporting structural components. The robot operates by first scanning and analyzing all six faces of the Rubik's cube using computer vision techniques to determine the current scrambled state. Once the cube configuration is captured and processed, the system applies Kociemba's algorithm to calculate an optimal solution sequence. The robot then executes the solution by systematically rotating the cube faces using precisely controlled motors until the puzzle is completely solved.

The project demonstrates the successful integration of multiple engineering disciplines, including embedded systems programming, computer vision, mechanical design, and algorithmic problem-solving. Developed using Python programming language, the system showcases the practical application of theoretical concepts learned during our academic program. This work contributes to the field of educational robotics and serves as a comprehensive demonstration of autonomous puzzle solving capabilities, bridging the gap between algorithmic theory and practical robotic implementation.

**TABLE OF CONTENTS**

[LIST OF FIGURES iii](#_Toc433201299)

[LIST OF TABLES iv](#_Toc433201300)

[LIST OF SYMBOLS AND ABBREVIATIONS ….. v](file:///C:\AURAK-PC30June2015\AURAK\AURAK%20Committee\Senior%20Design%20Committee\SDP%20old\SDP%20old\SDP%20I%20Template%20modified%20v1.docx#_Toc433201407)

[**1** **CHAPTER 1: INTRODUCTION** 1](#_Toc433201302)

[1.1 Problem Statement and Purpose](#_Toc433201303) 1

[1.2 Project and Design Objectives](#_Toc433201304) 1

[1.3 Intended Outcomes and Deliverables](#_Toc433201305) 1

[1.5 Summary of Report Structure](#_Toc433201307) 1

[**2** **CHAPTER 2: BACKGROUND**  2](#_Toc433201312)

[2.1 Overview](#_Toc433201313) 2

[2.2 Related Work 2](#_Toc433201316)

[**3** **CHAPTER 3: METHODS AND MATERIALS** 3](#_Toc433201318)

[3.1 System Design and Components 3](#_Toc433201319)

[3.2 Design Specifications, Standards and Constraints 3](#_Toc433201320)

[3.3 Design Alternatives 3](#_Toc433201320)

[3.4 System Analysis and Optimization 3](#_Toc433201321)

[3.5 Simulation and/or Experimental Test 3](#_Toc433201322)

[**4** **CHAPTER 4: RESULTS AND DISCUSSIONS** 4](#_Toc433201318)

[4.1 Results 4](#_Toc433201319)

[4.2 Discussions 4](#_Toc433201319)

[**5** **CHAPTER 5: PROJECT MANAGEMENT** 5](#_Toc433201335)

[5.1 Tasks, Schedule and Milestones 5](#_Toc433201336)

[5.2 Resources and Cost Management 5](#_Toc433201337)

[5.3 Lessons Learned 5](#_Toc433201339)

[**6** **CHAPTER 6: IMPACT OF THE ENGINEERING SOLUTION** 6](#_Toc433201330)

[6.1 Economical, Societal and Global 6](#_Toc433201331)

6[.2 Environmental and Ethical 6](file:///C:\\AURAK-PC30June2015\\AURAK\\AURAK%20Committee\\Senior%20Design%20Committee\\SDP%20old\\SDP%20old\\SDP%20I%20Template%20modified%20v1.docx" \l "_Toc433201429)

6[.3. Other Issues 6](file:///C:\\AURAK-PC30June2015\\AURAK\\AURAK%20Committee\\Senior%20Design%20Committee\\SDP%20old\\SDP%20old\\SDP%20I%20Template%20modified%20v1.docx" \l "_Toc433201429)

[**7** **CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS** 7](#_Toc433201340)

[7.1 Summary of Achievements of the Project Objectives 7](#_Toc433201343)

[7.2 New Skills and Experiences Learnt 7](#_Toc433201344)

[7.3 Recommendations for Future Work 7](#_Toc433201346)

[**REFERENCES** 8](#_Toc433201347)

[**APPENDICES** 9](#_Toc433201348)

[Appendix A: 9](#_Toc433201349)

[Appendix B: 9](#_Toc433201350)

[Appendix C: 9](#_Toc433201351)

***Note (Delete these lines):*** *The section titles in each chapter are guidelines and you may modify the titles or add other section titles and/or subtitles in consultation with your project advisor as there may be variations in the departmental requirements.*

*(Senior Project I chapters 1 -3),*

*(Senior Project II All chapters)*

**LIST OF FIGURES**

*Figure Page*

Fig. 1. Mercury 12

Fig. 2. Venus 13

Fig. 3. Earth 14

Fig.4. Mars 15

**LIST OF TABLES**

*Table Page*

Table. 1. Mercury 12

Table 2. Venus 13

Table 3. Earth 14

Table 4. Mars 15

**LIST OF SYMBOLS AND ABBREVIATIONS**

This is where you should put all the symbols used and abbreviations **(must be sorted)**. It is stored in a table format same as the table of contents. Use inserts a row to add more entries….etc

|  |  |
| --- | --- |
| **LIST OF SYMBOLS** | |
| **ω** | Angular velocity in rad/s |
| **ξ** | Damping ratio |
|  |  |
| **LIST OF ABBREVIATIONS** | |
| **ABET** | Accreditation Board For Engineering And Technology |
| **EC** | Examining Committee |
| **PC** | Project Committee |
| **STEM** |  **S**cience   **T**echnology   **E**ngineering   **M**athematics |
| **BFS** | Breadth First Search |
| **CNN** | Convolutional Neural Network |
| **CFOP** | CFOP: Cross, F2L (First Two Layers), OLL (Orientation of the Last Layer), PLL (Permutation of the Last Layer) |
| **DFS** | Depth First Search |
|  |  |

1. **CHAPTER 1: INTRODUCTION**

This project focuses on the design and implementation of an autonomous robot capable of solving a 3×3 Rubik's cube, integrating multiple engineering disciplines including computer vision, mechanical design, embedded systems, and algorithmic implementation. The challenge lies in creating a robust physical system capable of precise cube manipulation and reliable state detection while applying the knowledge gained throughout our Computer System Engineering studies.

We chose Kociemba's algorithm because it finds efficient solutions with a low number of moves, often under 21 steps. It's fast, well documented, and works well with the standard 3×3 cube structure. This makes it ideal for real time solving in a robotic system, where fewer moves mean less mechanical complexity and faster results. However, implementing this algorithm in a physical robotic system presents several significant technical challenges. Color detection requires accurately identifying the cube's colors using a camera under different lighting conditions, which can be particularly difficult. Mechanical control involves designing a stable and precise system to rotate the cube faces without slipping or misalignment. Integration challenges arise from ensuring that hardware components including motors and cameras work seamlessly with software systems encompassing vision processing and algorithm implementation in real time. Additionally, calibration requires precise alignment of the camera, cube positioning, and mechanical arms for consistent operation.

The scope of this project encompasses the complete design and implementation of an autonomous Rubik's cube solving robot. The main objectives include developing a robust computer vision system for accurate cube state detection, implementing Kociemba's algorithm for efficient solution generation, designing and constructing a reliable mechanical manipulation system, integrating all components using a Raspberry Pi based control system, and demonstrating successful autonomous cube solving capabilities. The expected outcomes include a fully functional robot capable of consistently solving scrambled 3×3 Rubik's cubes, comprehensive documentation of the design and implementation process, and demonstration of the practical application of Computer System Engineering principles in a real world robotics project.

* 1. Problem Statement and Purpose

This project addresses the technical challenge of designing and implementing an autonomous robotic system that can detect, analyze, and solve a scrambled 3×3 Rubik's cube without human intervention. The system must integrate computer vision for cube state recognition, Kociemba's algorithm for solution computation, and precise mechanical control for cube manipulation using Raspberry Pi based embedded systems.

The purpose of this project is to demonstrate comprehensive application of Computer System Engineering principles through a complex, multi-disciplinary robotics implementation. This project provides practical experience in embedded systems programming, computer vision, mechanical design, and real time algorithm implementation, directly preparing students for industry challenges requiring integration of hardware and software systems. The project serves as an educational platform that bridges theoretical knowledge with practical engineering skills, contributing to STEM education by showcasing how multiple engineering disciplines collaborate to achieve autonomous system functionality.

* 1. Project and Design Objectives
* Solving Speed: The robot shall solve a standard 3×3 Rubik's cube in 30 seconds or less from initial cube scanning to completion.
* Success Rate: The system shall achieve a minimum 90% success rate in correctly solving randomly scrambled 3×3 Rubik's cubes across 20 test attempts.
* Color Detection Accuracy: The computer vision system shall accurately identify all six cube face colors with 95% accuracy under standard indoor lighting conditions.
* To develop our skills in building similar and diverse robotic systems by combining various hardware and software engineering techniques.
  1. Intended Outcomes and Deliverables

We intend to build a Rubik’s Cube Solving Robot, an embedded mechatronic system capable of automatically solving a standard 3x3 Rubik’s Cube. The robot will consist of a mechanical structure operated by motors to manipulate the cube, a software interface to receive or detect the cube's state, and a solving engine that applies the Kociemba algorithm to determine the optimal move sequence. The key outcome of this project is to demonstrate how embedded systems, control algorithms, and mechanical design can be integrated into a fully automated problem solving machine. This system aims to showcase the practical application of engineering principles in robotics, optimization, and real time control.  
This project is intended as a learning platform and proof of concept for more advanced robotic manipulation systems. We aim to present it at engineering exhibitions and hope to inspire further development in educational robotics or puzzle solving automation.

* 1. Summary of Report Structure

This report is organized into seven chapters. The second chapter provides an overview of the project and discusses related work in the field. The third chapter focuses on the system design and its components, covering design specifications, standards, constraints, design alternatives, system analysis, optimization, and simulation or experimental testing. Chapter four presents and analyzes the project results. Chapter five details the management aspects of the project, including time management, cost management, and lessons learned. Chapter six examines the project's costs, production considerations, and its social and environmental impact. Finally, chapter seven reflects on the project goals achieved, evaluates the level of success, highlights the skills and experiences gained, and offers recommendations for future work.

1. **CHAPTER 2: BACKGROUND**
   1. Overview

The Rubic’s Cube was invented in 1974, it is a three-dimensional puzzle that has fascinated millions around the world, competitions are now held to solve it, attracting people of all ages to try to solve it, solving the cube independently using robots requires the integration of multiple engineering disciplines, including computer vision, mechanical design, embedded system, and algorithm optimization. The challenge lies in creating a system capable of accurately determining the state of the cube, calculating the optimal solation and performing precise mechanical operations to efficiently solve the puzzle.

In this project, we used the Kociemba’s algorithm, a two- phase algorithm that significantly reduces the cube’s solution space, enabling the puzzle to be solved efficiently with the fewest moves, this algorithm is widely known for its balance between speed and optimal solution, making it ideal for robotic applications.

* 1. Related Work

### AI Rubick Cube solver

This is the first project related to our project and this project uses artificial intelligence to solve the Rubik’s Cube puzzle automatically, this project relies on algorithm like BFS, but instead, we worked on the Kociemba’s algorithm because it’s faster and better. This project would require thousands of moves to solve the cube, whereas ours solves the cube in 21 moves or less, this project relied on CNN, but we used cameras.

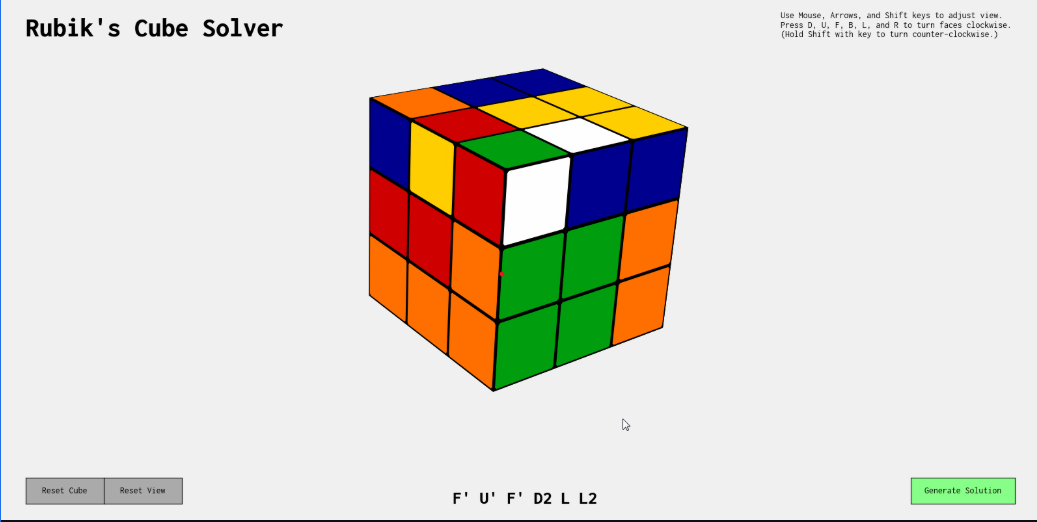


Figure 1

### Robot solves Rubic’s cube

The second project is a physical robot (not just a software simulation) that solves a Rubic’s Cube using motors and sensors. This project also aims to solve a Rubik’s Cube automatically and relies on a camera and motors, it’s similar to our project.

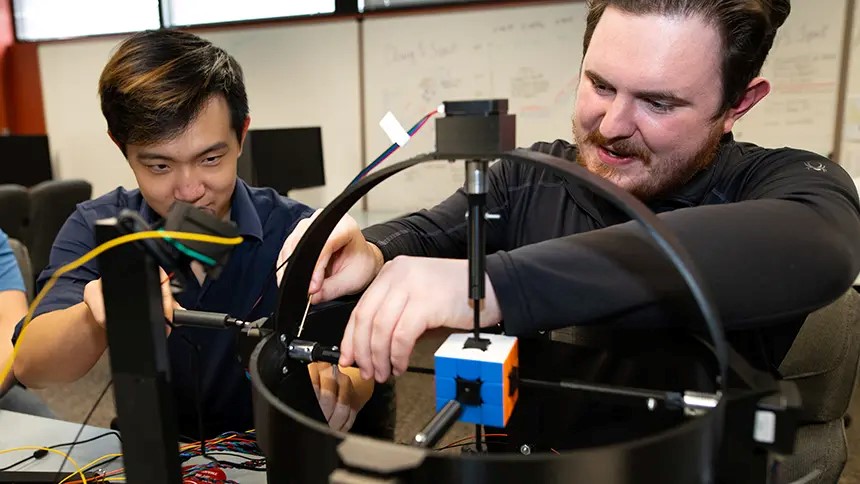


Figure 2

### CubeMoster

The third project is very similar to ours, solving the cube using several algorithms, such as CFOP, DFs and Kociemba’s, which we will use in our project. The main difference between the two projects is that they use an Arduino, a raspberry, while our project does not use an Arduino.



Figure 3

### Solving the Rubik’s cube with deep reinforcement learning and search

In this research, they created an artificial intelligence capable of solving the Rubic’s Cube without the need to study human solutions. It solves the cube in 20 moves, but it is merely a program that does not moves a real cube, this research is purely theoretical, our project is practical, and the cube will be moved by a real robot.

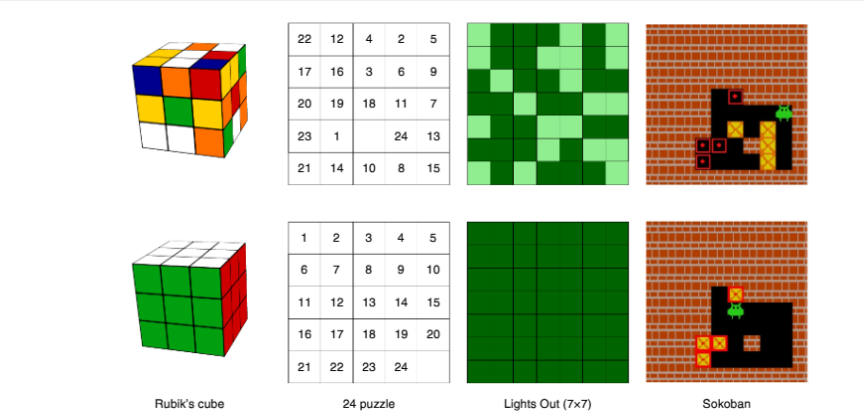


Figure 4

1. **CHAPTER 3: METHODS AND MATERIALS**

Architecture and Design of Systems

Overview of the Project

Building an autonomous robot that solves Rubik's Cubes using computer vision, servo control, and algorithmic problem-solving is the focus of our senior project. A Raspberry Pi, two cameras, six servo motors, and a mechanical base make up the system, which is used to manipulate and solve a typical 3x3 Rubik's Cube. The Python implementation of the Kociemba algorithm serves as the foundation for the reasoning behind the solution .

The project was divided into four main subsystems:

1. **Image Acquisition and Processing**  
   Two cameras were mounted on opposite sides of the cube to capture all six faces by rotating the cube as needed. Images are processed using **OpenCV** in Python to detect and map the color of each square.
2. **State Recognition and Solving Algorithm**  
   The recognized cube colors are converted into a string format that represents the cube state. This state is passed into the **Kociemba algorithm**, which returns an optimized sequence of moves to solve the cube.
3. **Hardware Control and Cube Manipulation**  
   Six **servo motors** are mounted around a custom base to grip and rotate each face of the cube. The Raspberry Pi generates **PWM signals** to control the motors based on the solution sequence.
4. **System Coordination and Execution**  
   A control program on the Raspberry Pi orchestrates the full process: capturing images, interpreting the cube state, computing the solution, and commanding the servo motors to execute the moves.

Addressing Issues and Difficulties

Accuracy of Color Detection: Due to lighting, early tests revealed uneven color recognition. In order to reduce shadows and reflections, we added diffused lighting and calibrated the RGB color ranges.Servo Precision and Alignment: During rotation, a few servos misaligned the cube. With careful motor placement and delay timings, we created a 3D-printed base that enables complete face rotations prior to the subsequent command.Camera Angle and Visibility: Using two fixed cameras to capture every face was challenging. To minimize the number of cameras needed, we used motors to rotate the cube so that each face was exposed in a precise order.Mechanical Stability: During rotation, the cube occasionally slipped. For improved grip, we added rubber padding and redesigned the gripper arms to apply uniform pressure.

Hardware Components of the Detailed System Architecture:

Raspberry Pi 4: Central processing unit that runs Python code and manages every component.

Take pictures of the Rubik's Cube from various perspectives using two USB cameras.

Six servo motors—one for each face—control the rotation of the cube faces.

Power Supply: Provides the Raspberry Pi and motors with the proper voltage.

The cube is held in place and all motors are mounted firmly by the custom base and grippers.OpenCV is a software component used for color detection and image capture.

Detected cube state is converted to a solving sequence using Python and the Kociemba Module.

GPIO Library (Rpi.GPIO): Provides servo motors with PWM signals from the Raspberry Pi.

The main control script is in charge of timing, sequencing, and coordinating motor control, vision, and algorithms.

Overview of the System Workflow

Startup: All cameras and motors are initialized by the system.

Image Capture: The cube is photographed by cameras, with faces rotated as necessary.

Color Mapping: Each sticker's color is determined by the software.

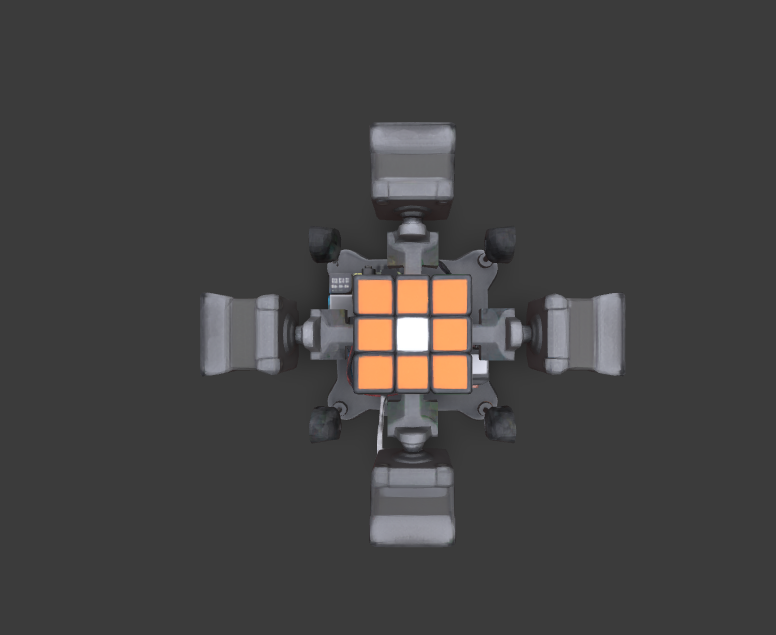
Solving: The Kociemba algorithm receives the current state as input and outputs a solution.

Execution: In accordance with the solution sequence, servo motors rotate the cube gradually.

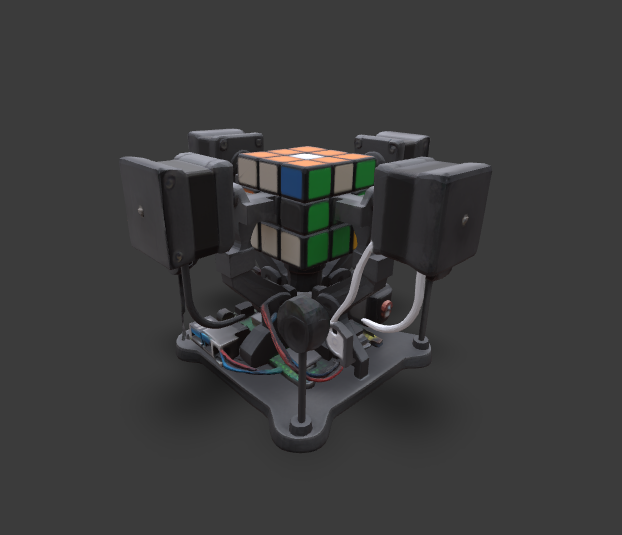
Completion: The system stops or resets when the cube reaches the solved state.

* 1. System Design and Components

### SYSTEM DESIGN



Top



Front



Bottom

### components of project

### Raspberry Pi

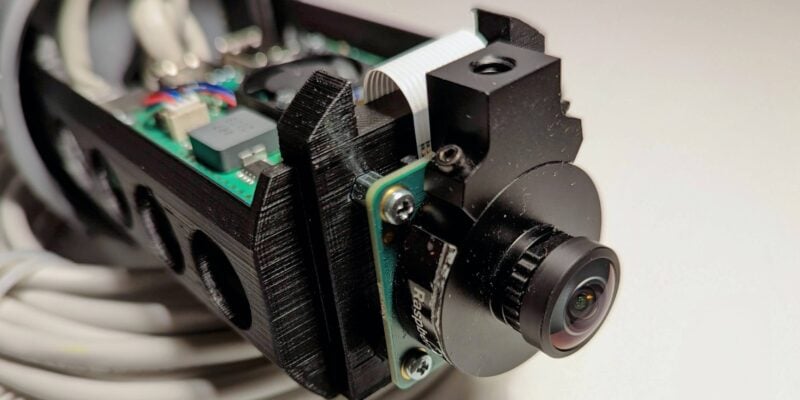
### 

The Raspberry Pi acts as the central processing unit, coordinating all hardware and software operations of the Rubik’s Cube solving robot.

It captures images from the two cameras, processes them using computer vision, and determines the cube's color configuration.

Based on the solved sequence from the Kociemba algorithm, the Raspberry Pi sends control signals to the servo motors to rotate the cube accordingly.

1. Cameras :



To determine each square's location and color, the cameras take high-resolution pictures of each face of the Rubik's Cube.

The Raspberry Pi receives the captured images and uses OpenCV for image processing and color recognition.

Before solving, the system can accurately reconstruct the cube's current state thanks to precise visual data from the cameras.

1. Servo Motors :



Commands from the Raspberry Pi cause the servo motors to physically rotate the Rubik's Cube's faces.

With precise angular movement, each of the six servo motors is in charge of grasping and rotating a particular cube face.

They carry out the algorithm-calculated solution steps, enabling the robot to solve the cube on its own.

### BAse :



The base supports the mounting of all six servo motors and offers a sturdy framework for safely holding the Rubik's Cube.

When the motors rotate each face, it makes sure the cube stays correctly aligned throughout rotation, avoiding slippage or misalignment.

Additionally, the base aids in the orderly arrangement of the parts, which makes the system small, strong, and simpler to put together and maintain.

### software

**Kociemba algorithm**

The Rubik's Cube can be solved from any jumbled state by using the Kociemba algorithm to determine an effective and ideal series of moves.

After image processing has identified the cube's current state, the algorithm uses this state as input in string format to produce a step-by-step solution.

The servo motors are then able to carry out each move and solve the cube by converting the output of the Kociemba algorithm into motor control commands.

* 1. Design Specifications, Standards and Constraints

- Design specifications:

* Hardware Requirements:  
  Raspberry Pi 4
* Six servo motors
* USB camera or Raspberry Pi Camera Module
* Power supply
* Mechanical frame supporting
* Standard 3×3 Rubik's cube compatibility

-Performance Specifications:

* Cube solving time : ≤30 seconds
* Success rate : ≥ 90% across 20 test attempts
* Color detection accuracy : ≥ 95% under standard indoor lighting
* Rotation precision: ±2 degrees per servo movement
* Operating temperature: 0°C to 40°C

-Applicable Standards:

Hardware Standards:

* IEEE 1394 (FireWire): For high-speed data transfer between camera and processing unit
* USB 2.0/3.0 Standards: Camera interface and peripheral connectivity
* GPIO Standards: Raspberry Pi GPIO pin configurations and electrical specifications
* PWM Standards: Servo motor control signal specifications (50Hz frequency, 1-2ms pulse width)

Software Development Standards:

* POSIX (IEEE 1003): Unix-like operating system compatibility for Raspberry Pi OS
* Python Enhancement Proposals (PEP): Code style and structure guidelines (PEP 8, PEP 257)
* OpenCV Standards: Computer vision library implementation for image processing
* IEEE 754: Floating-point arithmetic standard for algorithm calculations

Communication Standards:

* I2C Protocol: Inter-integrated circuit communication for sensor interfaces
* SPI Protocol: Serial peripheral interface for high-speed device communication
* UART Standards: Universal asynchronous receiver-transmitter for serial communication

Open Source Standards:

* Linux Kernel Standards: Raspberry Pi OS (Debian-based) compliance
* GNU General Public License (GPL): Open-source software licensing
* Python Software Foundation License: Python interpreter and libraries
* OpenCV BSD License: Computer vision library usage

Quality and Safety Standards:

* ISO 9001: Quality management system guidelines for project development
* IEC 61508: Functional safety standards for electrical systems
* RoHS Directive: Restriction of hazardous substances in electronic components

Design Constraints:

Physical Constraints:

* Maximum system footprint: 300mm × 300mm × 400mm
* Power consumption limitation: ≤15W total system power
* Operating noise level: ≤60dB during cube manipulation
* Component accessibility for maintenance and calibration

Performance Constraints:

* Real-time processing requirements for computer vision algorithms
* Servo motor response time limitations (0.15 sec/60° rotation)
* Memory constraints of Raspberry Pi 4 (4GB RAM maximum)
* Processing power limitations for algorithm optimization

Environmental Constraints:

* Indoor lighting conditions (minimum 200 lux illumination)
* Temperature stability requirements for consistent servo performance
* Vibration isolation to prevent mechanical interference during operation
* Electromagnetic interference considerations for camera and motor operation

Economic Constraints:

* Project budget limitations for component selection
* Cost-effective material choices for mechanical frame construction
* Balance between performance requirements and component costs

Technical Constraints:

* Kociemba's algorithm implementation complexity and optimization requirements
* Computer vision accuracy limitations under varying lighting conditions
* Mechanical precision constraints due to servo motor tolerances
* Integration challenges between hardware and software components

Time Constraints:

* Project development timeline restrictions
* Testing and validation phase limitations
* Documentation and presentation preparation requirements
  1. Design Alternatives

### **Solver algorithm**

Our initial alternative design for the project involved developing our own algorithm to solve the cube. However, we ultimately decided to implement Kociemba’s algorithm instead, as it offers greater efficiency and reliability. Creating a new algorithm from scratch would have required significantly more time and effort without providing any substantial benefits—essentially reinventing the wheel.

### **Arduino**

The second design alternative was to use an Arduino board to handle the project. However, we found that it lacks the processing power and memory required to run the AI module we intended to use. As a result, we decided to use a Raspberry Pi instead, as it offers significantly more computational resources and memory, making it better suited for our project's AI requirements.

### **number of cameras**

The third alternative design we considered involved the number of cameras used in the system. Initially, we planned to use six cameras—one for each face of the cube. However, we later decided to use only two cameras positioned at the corners of the cube, allowing each camera to capture three faces simultaneously. This approach significantly reduced the project's cost and complexity while still providing sufficient visual coverage for our needs.

### **data mining algorithms**

The final design alternative involved our choice of motor. Initially, we planned to use a stepper motor, but we ultimately decided to switch to a servo motor. This decision was based on performance—servo motors are significantly faster than stepper motors, which aligns with our project's main goal: achieving high-speed operation. By choosing servo motors, we were able to optimize the system for better efficiency and responsiveness.

* 1. System Analysis and Optimization

The development of the Cube Mind Rubik's Cube solver involved a multi-faceted approach to system analysis and optimization, addressing various components critical to its autonomous operation. The primary analyses focused on the efficiency and reliability of the computer vision system, the precision and speed of the mechanical manipulation system, and the performance of Kociemba's algorithm within the embedded Raspberry Pi environment.

Computer Vision System Analysis:

Initial analysis of the computer vision system centered on its ability to accurately detect and interpret the colors of the Rubik's Cube faces under varying lighting conditions. Experimental work involved testing different camera angles, distances, and ambient light settings to identify optimal parameters for consistent color recognition. This included evaluating various image processing techniques, such as color space conversions (e.g., RGB to HSV) and thresholding, to enhance color differentiation and minimize noise. Performance metrics included the percentage of correctly identified stickers and the time taken for a complete cube scan. Early challenges included glare from reflective surfaces and the subtle variations in color shades, which necessitated iterative refinement of the vision algorithms. Further analysis is required to implement adaptive lighting compensation or more robust color calibration routines to ensure performance across a wider range of environmental conditions.

Mechanical Manipulation System Analysis:

The mechanical system, responsible for physically rotating the cube faces, underwent rigorous analysis to ensure precision, speed, and reliability. This involved evaluating the choice of motors (e.g., stepper motors for precise angular control), the design of the gripping mechanism to prevent slippage, and the overall structural stability of the robot. Experimental trials measured the accuracy of each turn, the time required to execute a sequence of moves, and the repeatability of operations. Backlash in gears and slight misalignments were identified as areas requiring optimization. Future work in this area includes exploring alternative mechanical designs for faster and smoother transitions between moves, as well as implementing real-time feedback mechanisms (e.g., encoders) to correct for minor positional errors during operation.

Kociemba's Algorithm Implementation and Performance Analysis:

Kociemba's algorithm, chosen for its efficiency in finding solutions with a low number of moves, was analyzed for its computational performance on the Raspberry Pi. This involved profiling the algorithm's execution time for various scrambled states and assessing its memory footprint. Optimization efforts focused on reducing the search space and improving the lookup table efficiency to ensure real-time solution generation. Experimental tests involved feeding the algorithm with a diverse set of scrambled cubes and measuring the time taken to generate a solution sequence. While the algorithm itself is highly optimized, its integration with the vision and mechanical systems introduced latency. Remaining work includes further optimizing the inter-process communication between the vision, algorithm, and control modules to minimize overall solving time. Additionally, exploring parallel processing techniques on the Raspberry Pi could further enhance the algorithm's responsiveness.

System Integration and Overall Performance:

The holistic system performance was analyzed by integrating all components and measuring the end-to-end time from cube placement to a solved state. This involved identifying bottlenecks in the workflow, such as the time taken for image capture, processing, algorithm execution, and mechanical movements. Initial trials revealed that the sequential nature of these operations contributed significantly to the overall solving time. Future work will involve exploring concurrent execution of certain tasks where feasible, such as pre-processing images while the mechanical system is executing a previous move. Comprehensive error analysis, including instances of misidentification or failed mechanical operations, was also conducted to identify failure modes and inform system improvements. Remaining work includes developing a more sophisticated error recovery mechanism to handle unexpected events during the solving process.

In summary, the system analysis and optimization efforts have laid a strong foundation for the Cube Mind robot. While significant progress has been made in understanding and improving individual components, further work is required in enhancing the robustness of the computer vision system, refining the precision and speed of the mechanical manipulation, and optimizing the overall system integration for faster and more reliable Rubik's Cube solving.

1. **CHAPTER 4: RESULTS AND DISCUSSIONS**
   1. Results

Summarize the results and draw conclusions based on those results.  Detailed results (e.g., printouts) can be placed in the Appendix to improve readability of the report. experimental results.

* 1. Discussions

Discuss your result that obtains in above section.

1. **CHAPTER 5: Project Management**
   1. Tasks, Schedule and Milestones

You are required to develop the schedule of your senior project which involves identifying the main and sub tasks, investigating task sequences, resource estimations, and task duration estimates to generate the project schedule that also includes milestones which are precise points used to measure the progress toward the final goal. A sample of a project schedule represented in Gantt chart format is shown in figure 5.1.

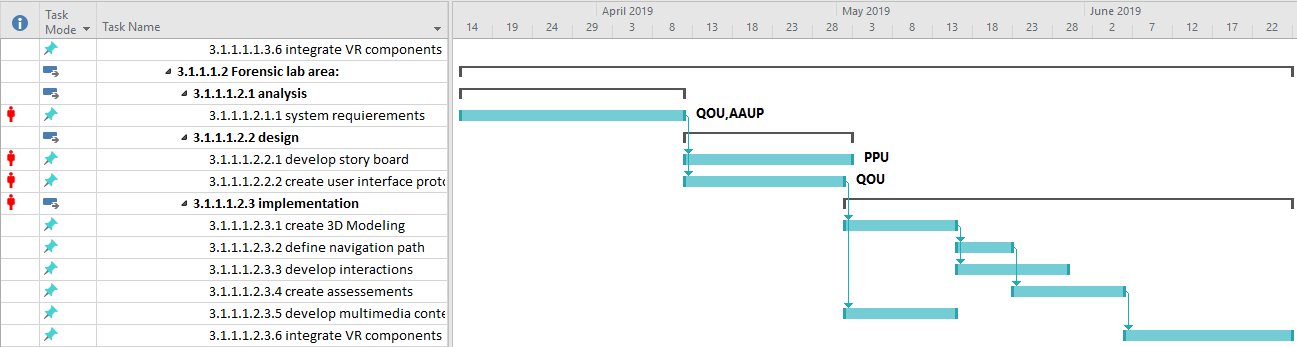


Fig. 5.1 A sample project schedule

* 1. Resources and Cost Management

In this section you have to estimate the needed project resources which involves assessing how many resources—people, tools, equipment, and materials—a project team should use to accomplish project tasks. You have to specify the cost of theses resources where you can fill these costs and calculate the budget of developing the project in a table such as the table below.

Table 5.1 Sample table to fill in total cost of senior project items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | # Unit | Unit Cost | Subtotals | Comments |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Total senior project estimate |  |  |  |  |

* 1. Lessons Learned

You have to document information that reproduces both the positive and negative practices of the senior project.

1. **CHAPTER 6: Impact of the engineering solution**

Sustainability consideration and constraints includes economic, environmental, and social (equity) aspects that need to be evaluated and taken into account in project research and development. There is a strong relationship between these three pillars of sustainability. These need to be considered and incorporated in this section with a discussion on their design constraint and the positive and negative effects of the project within this scope.

* 1. Economical, Societal and Global

Economics (cost) impact: should consider, when relevant,

1. Prototype design and production cost, including the manner in which production cost can be reduced, when applicable.
2. Device cost in mass production, including materials, operations, supports etc.
3. Cost saving of the product should be considered when appropriate. For example, energy savings compared with the use of other products, water saving, reduction in operation cost, etc.
4. Tax incentives to be considered towards final product cost. For example, renewable energy and energy efficient products tax incentives, carbon footprint reduction, etc.
5. Environmental aspects, such as availability of resources, may affect the product cost and therefore price and their market vulnerability.

Social impact of the product: when relevant, please consider

1. How can the developed product impact people lives. Is it a positive or negative impact?
2. What community or personal needs does it address?
3. Is the product going to change consumption patterns?
4. Is the product automating a task currently preformed manually and therefore might impact employment?
5. Does the product create new jobs or fields?
6. Safety aspects and health concerns
7. Regulation constraints that address social and environmental concerns
   1. Environmental and Ethical

Environmental impact of the product: when relevant, please consider

1. Increase or reduction in emissions obtained through modifications in processes that emit greenhouse gasses (GHG) or products that do so.
2. Change in consumption or use patterns, which effect the environment such as use of water, food, energy, wood, etc. (positive or negative affect).
3. Reliance on resources that are scarce (such as precious material) or abundant. For examples, some fuel cells technologies use rare material while other use abundant ones. This will have an impact on the availability of these materials as well as their prices.
4. Project production and operation effect on natural resources availability and competition on the planet resources. Considering their availability in nature and the impact of their consumption on the balance of nature.
5. Environmental regulation

* 1. Other Issues

1. **CHAPTER 7: CONCLUSIONS and RECOMMENDATIONS**
   1. Summary of Achievements of the Project Objectives

Draw conclusions about the level of success of your work.  Did you substantially meet your objectives?  What did you learn in the process?

* 1. New Skills and Experiences Learnt
  2. Recommendations for Future Work

This is one of the most important sections in the Final Report, at least for those who may pick up where you left off.  Knowing what you know now, given the same problem, what would you do if you had it to do all over again?  Understanding that problems of significance are rarely if ever solved perfectly the first time, what would you suggest as the next step toward finding the answer?

**REFERENCES**

1. Rashid M. H., *Power Electronics: Circuits, Devices & Applications*, 4th Ed., Upper Saddle River, New Jersey: Prentice Hall, 2013.
2. Foschini G. J. and Gans M. J., “On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas,” *Wireless Personal Communications,* Vol. 6, No. 3, pp. 311-335, March 1998.
3. Gans M. J., "A power-spectral theory of propagation in the mobile radio environment,” *IEEE Transactions on Vehicular Technology*, Vol. 21, Issue 1, pp.27-38, 1972.
4. Dahl G., Suttrop F., “Engine Control and Low-NOx Combustion for Hydrogen Fuelled Aircraft Gas Turbines,“ *International Journal of Hydrogen Energy,* Vol. 23, No. 8, pp. 695–704, 1998.
5. Watermark 200SS soil moisture sensor specification manual. Available at http://www.irrometer.com/sensors.html

**APPENDICES**

**Appendix A:**

**Appendix B:**

**Appendix C:**