Self Solving/Scrambling Rubik's Cube for Learning and Training

Byron Lathi

Colin Choi

Walter Uruchima

ECE 445 Design Document- Fall 2022

TA: Li Qing Yu

Project No. 09

29 September 2022

0 | Contents

1 | Introduction

1.1 Background

Designed and created by Hungarian teacher, Erno Rubik, the Rubik's Cube has been a smash hit for puzzle solvers across the globe. Since its release to stores in 1980, over 450 million Rubik's cubes have been sold making it the most popular and best-selling toy in history. Along with these incredible numbers, Rubik's Cube has also found itself a dedicated group of avid solvers, commonly known as 'cubers'. From the beginning, these people have been learning, working, and competing for faster times in solving the cube. With the current world record for a single 3x3 Rubik's cube solve sitting at 3.47 seconds, it is clear that modern cubers have become incredibly quick at solving the once impossible puzzle.[1]

The standard Rubik's Cube consists of six faces each with a 3x3 grid of colored squares. In its solved state, all 9 of the squares per side will show the same color. In a scrambled state, the colors are scattered and each face will show an array of colors. The standard color set is red, blue, white, orange, green, and yellow. Each of the sides revolve around the center square within the 3x3 grid and can be rotated in either direction.

1.2 Problem

Naturally, the puzzle can be difficult to solve when attempting it for the first time. Beginners are often daunted by the algorithms used to solve the cube and struggle with memorizing them as well. It can almost even feel like learning a new language with all of the new terms and the strings of letters, numbers, and apostrophes that instructionalize the solving algorithms. There are many tools and resources online that can be used to aid the learning process, but these are not always perfectly clear in depicting the necessary steps required to solving the cube. Especially for a beginner, starting out on the cubing journey can be complicated.

On the other end of the spectrum, skilled cubers often find scrambling the cube to be a cumbersome task. You have to randomize which sides to turn and which way to rotate them by. There must be at least around 20 steps worth of randomization in order to achieve a high quality scramble[2]. However, years of practicing cubing leads to random biases during the scrambling process. Turns become less random and cubers often end up giving themselves suboptimal scrambles. These make for worse practice and slow down improvements to skill.

1.3 Solution

We will work towards solving the listed problems by creating a Rubik's Cube that is capable of scrambling and solving itself. Our device contains six integrated motors with controllers to turn each of the sides of the cube. It will be controlled by a microcontroller and powered by batteries run in series. Limit switches will also be used on each face of the cube to aid with precision turning control. Users will be able to interface with the cube and change from solving/scrambling settings by turning sides in a specific pattern.

The cube will be programmed to use the hardware inside to solve the cube regardless of what kind of scramble it is given. This will allow beginner cubers to visually see the necessary steps to solve the cube and gain better hands-on experience with the correct algorithms. No longer will beginners have to sift through dense tutorials for suboptimal solving algorithms.

Experienced cubers will also be able to extend their skills through better scrambles. The cube will be programmed to scramble itself to the correct extent with (pseudo) random processes. This will prevent experienced users from gravitating towards biased scrambles from their years of continued cubing.

1.4 Visual Aid

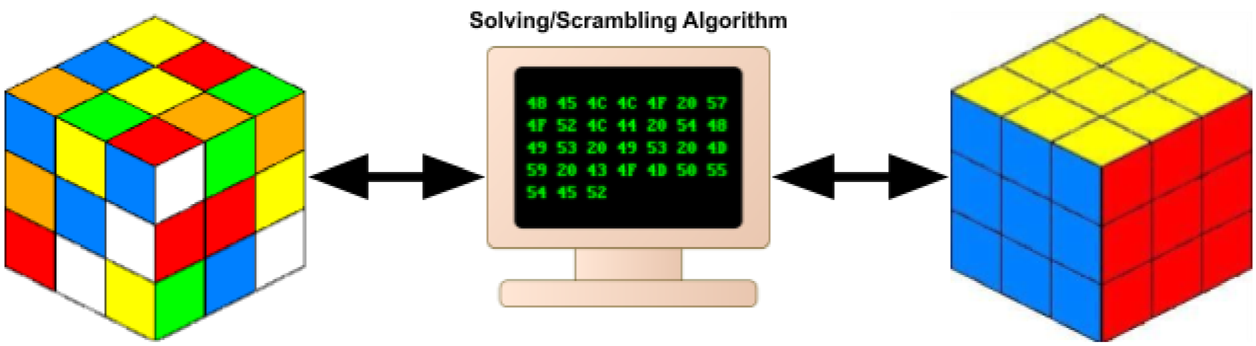

Figure 1: Project Functionality Visual Aid

1.5 High-Level Requirements List

1. The cube must be able to function as a normal Rubik's cube would, independent of the electronics inside of it.
2. The cube must be no larger than 150mm x 150mm x 150mm
3. The cube must be able to solve and scramble itself in under a minute.

2 | Design

2.1 Block Diagram

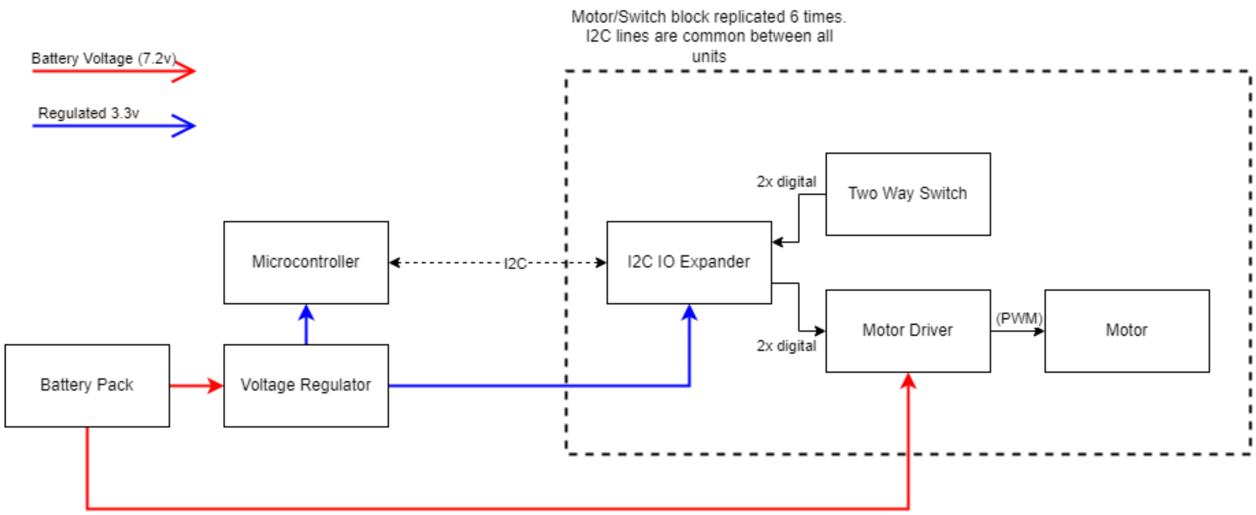

Figure 2: Block Diagram of Project Systems

2.2 Physical

Faces are aligned by way of a bidirectional switch which is activated by the shaft of the center face. These are a certain distance from the switch so as to activate them only when the face is at a 90 degree angle to the switch. Since each side is its own switch, the direction of rotation can be determined.

The cube operates in the same method as a traditional rubik's cube, with interlocking pieces anchored around the center faces, which are in turn attached to the motors with a screw and spring to provide the force that keeps the cube together while allowing it to turn. The screw threads into a hollow shaft which is rigidly fixed to the motor shaft.

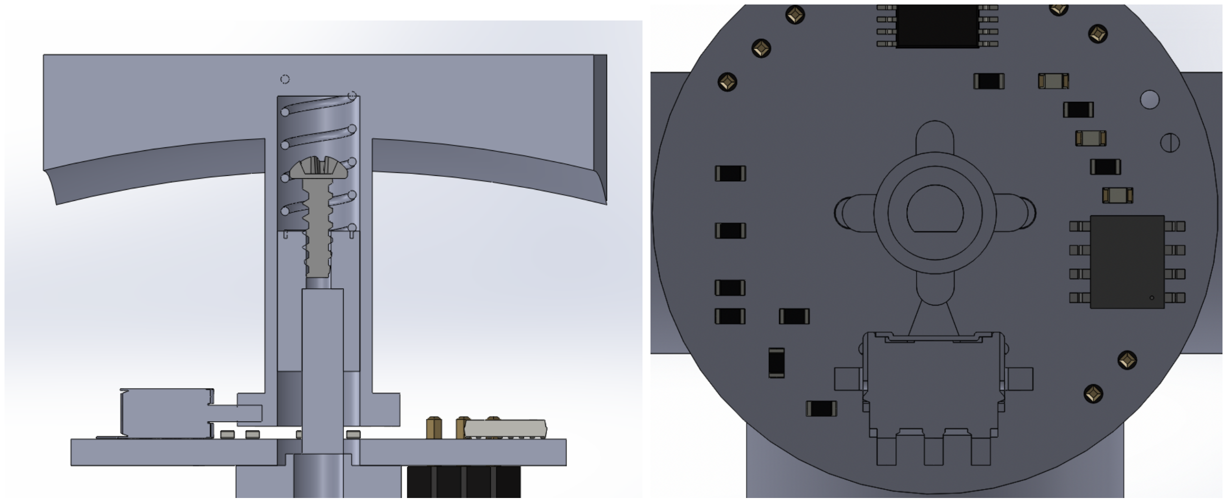

Figure 3: Center Face Connection Design (left) and Switch Module (right)

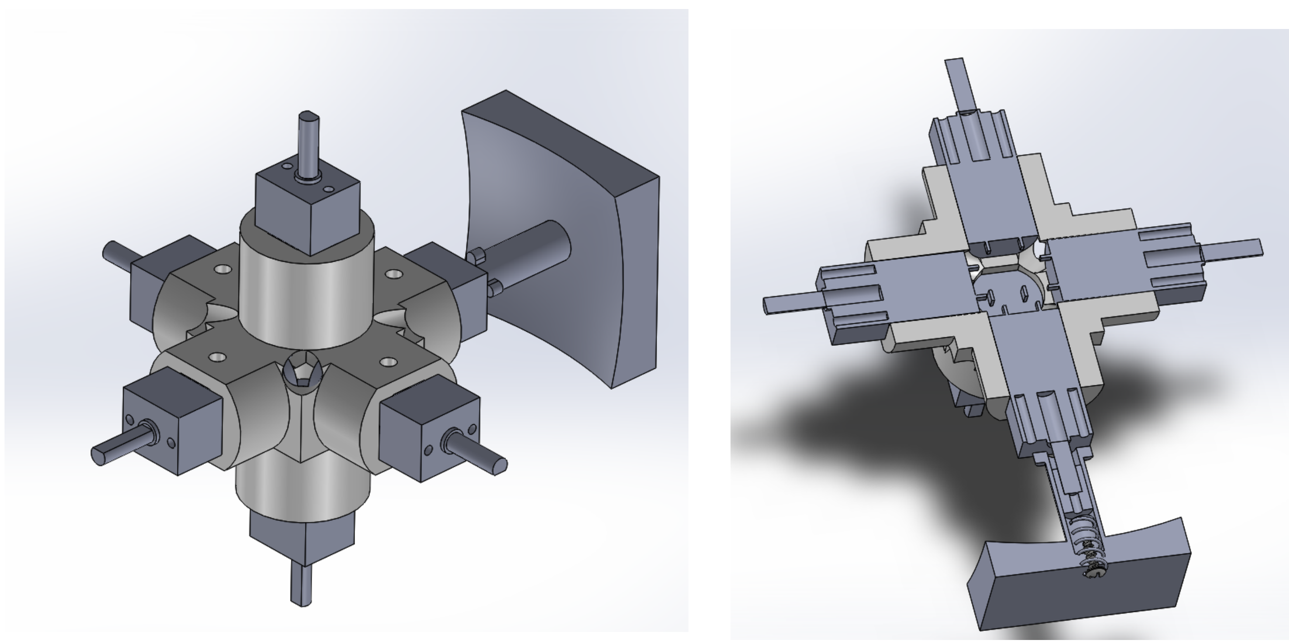

Figure 4: DC Motor Hub Core (left) and Core Sliced View (right)

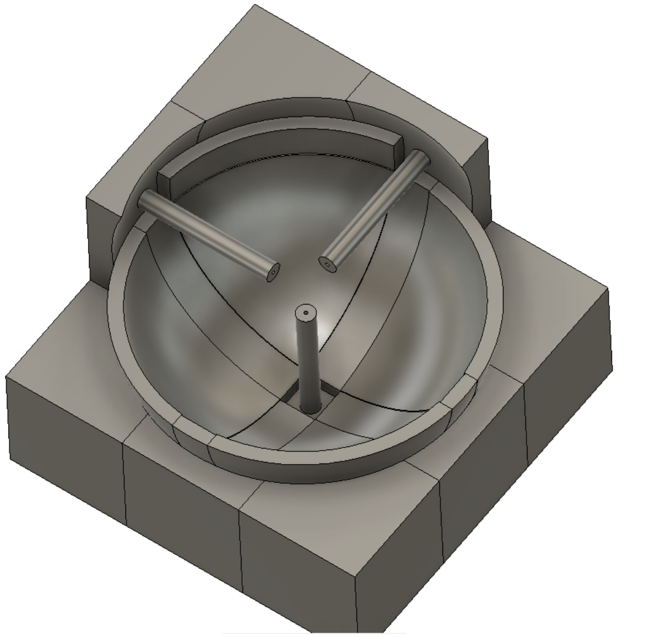

Figure 5: Outer Cube Shell

2.3 Subsystem Overview

2.3.1 Switch Module

The switch modules contain a bidirectional switch, an I2C I/O expander, and a motor driver. The switch is a Panasonic ESE24, which features separate outputs for both left and right switching directions. The I/O Expander is a PI4IOE5V9554, which has 3 address pins allowing up to 8 bit states to be used in the same system[3]. It contains 8 I/O pins, of which only 4 are needed. The ZXBM5210 Motor Driver allows the motor to be driven in forward and reverse directions as well as be set in coast and brake states[4]. This motor driver has a maximum output current of 1.5A, which is well above the 0.67 A stall current of the motors. Additionally, the driver contains an internal pwm oscillator which can be tuned to lower the motor output power if needed. The maximum supply voltage is 18 volts, which is well above the nominal motor voltage of 6V. Each motor driver will drive one mini gear motor. The motors have a 1:1000 gear ratio, rotating at approximately 10RPM at 6 V, with a stall torque of 5.5 Kg-cm. We plan to power the motors with 2 batteries, which would result in a voltage of 7.4V. As the motors will only be one for ~1.5 seconds at a time, this will not overload them. If the motors do get overloaded, the output power of the motor drivers can be reduced as mentioned above. The motors are similar to Pololu Micro Metal Gearmotors.

Requirement	**Verification**
Switch must be activated when each face is aligned 90 degrees	Rotate face 2 full rotations in either direction, ensure switch is closed when face is within 5 degrees of alignment and is open when the face is misaligned.
*Motors must have enough torque to turn the cube (est. 0.5 N*M)*	Adjust spring tension until current required to drive cube is within recommended current limit of motors (250 mA)
Motor driver must supply enough current to the motors without overloading	Measure current consumption of motors while turning cube, should not exceed 1A

2.3.2 Power System

Power is supplied by two LiPo batteries, resulting in a nominal battery voltage of 7.4V which is used to drive the motors. A 3.3v regulator is used to regulate power for the microcontroller and IO Expanders

Requirement	**Verification**
Batteries must supply enough current for the motors and control system	Measure the current consumed by the motors and control system using a bench power supply, ensure rated battery current is above that.

Battery voltage must be within 6.0 to 8.0v with 1A output	Measure voltage of battery while drawing 1A.
Voltage Regulator must be able to supply 3.0v to 3.6v with enough current to the ICs	Measure the control system only and ensure the rated current from the regulators is above that.

2.3.3 Software System

The software system will control the motors driving the faces of the cube, as well as be responsible for keeping track of the state of the cube at all times. This will be taken care of by the STM32 microcontroller which is capable of communicating over the I2C bus. The microcontroller can read the positions of each of the six switches and set the motor states over the I2C bus.

Two forms of user interaction will be available from its solved state. A user will be able to freely scramble the cube, or will be able to trigger a solve or scramble interaction by rotating the cube faces quickly. The default mode for the motors will be set to “coast” so that the user will be able to rotate the cube freely. In the event of a user freely scrambling, the software system will keep track of the state of the cube by having a 2-dimensional array for each face of the cube. Elements in the array will change based on which face piece is occupying that space on the cube. Each face will have a unique ID ranging from 1-6.

In the event of a user interaction to initiate a scramble, the software system will initiate a random number generator determining the face ID number that will be turned clockwise. The motor setting will then be changed to “forward”, while all others will be set to “brake”. A scramble will be completed after 20 turns, and motors will be set back to “coast”. During a user trigger solve, the software system will take into account the current state of the entire cube and identify which faces to turn and in which direction. During the solve, the cube will update the state of the cube. Motor control will be the same as when the cube is scrambling, but will have the ability to be set to “forward” or “backward”.

Requirement	**Verification**
Must be able to distinguish between a full face rotation from all other partial rotations	Will check the current state of the cube of the internal software system to see if it matched the outer cube appearance during a simulated partial turn
Must be able to keep track of states of full cube	Will check the current state of the cube in memory against the physical state of the cube after numerous rotations, and resets.
Must be able to identify user control rotations	Will check the current flag of the user interaction to see if it identifies as user control rotation
Must be able to scramble cube randomly	Match cube face rotations to random number generated rotation string
Should be able to solve the cube from any state it is in	From the unsolved rubix state, trigger the solve algorithm and check that the end state results in solved faces.

2.4 Tolerance Analysis

The block with the greatest risk to implement is the motor and feedback system. Each switch is bidirectional, with contacts for left and right. The cube will have some ability to center itself, but it is important for the switches to activate only when the face is aligned, and for the controller to stop the movement as soon as the switch is hit. Any distance the face spins after the switch is activated will not be detected, and the motor will have to realign itself if this happens.

There is no official torque specification for a rubix cube, although 0.05 N-M for speed cubes and worst case 0.5 N-M for original cubes are good estimations. We plan to 3D Print our own cube which will allow for optimizations to reduce the torque required. Even in the worst case, the micro gear motors have at least 0.5 N-M stall torque at 6V.

For battery consumption, each motor has a maximum stall torque of 360mA. With an estimated speed of 10RPM, it takes 1.5 seconds to make a single rotation. This means a single turn consumes 0.15mAh of charge. In a suboptimal solution with 200 turns, this results in 30mAh of charge being consumed. In reality this is an overestimate since this assumes the motors are working at stall current the entire time. If this were the case, then it means the motors are underpowered. The 200 turn solution would take 5 minutes to solve. With a more optimal solve of 60 moves, the cube would take 1.5 minutes to solve. With a perfect solution of 20 moves, it would take 30 seconds to solve.

3 | Cost and Schedule

3.1 Cost Analysis

The total cost of the project can be separated into two expenses. Mainly, the labor and materials cost. Labor can be calculated based on the total hours being put into the project and the average salary for electrical and computer engineers. Assuming an average salary of $40/hr * 2.5 hrs/day * 75 days * 3 members = $22,500 in total labor costs[5].

In addition to the total cost of parts, we have an additional cost towards 3D printing the chassis for the cube. The base for 3D printing is set at around $5 with an additional price of $0.15/gram[6]. Given 125 mm sides and around 15% infill printing, we're given 288 cmˆ3. This gives us an estimated cost of around $40 per print of the cube. We'll likely need one or two iterations to work out the kinks so our total estimation would be closer to around $80. So all costs accounted for, the total cost of the project would be $22,500 + $80 + $120 = $22,700.

3.2 Schedule

Week	Task
Week 1: 8/22	• Initial Web Board Post
Week 2: 8/29	• Lab Safety Training • Early Project Approval • CAD Assignment
Week 3: 9/5	• Project Approval
Week 4: 9/12	• Project Proposal
Week 5: 9/19	• Design PCB
Week 6: 9/26	• Design Document Checkup
Week 7: 10/3	• Design Review
Week 8: 10/10	• Order PCB • Initial 3D print
Week 9: 10/17	• Initial Testing
Week 10: 10/24	• Refining Code and Assembly
Week 11: 10/31	• Second Round of PCB if necessary • Individual Progress Report
Week 12: 11/7	• Finalize Project
Week 13: 11/14	• Mock Demo
Week 14: 11/21	• Fall Break
Week 15: 11/28	• Final Demo
Week 16: 12/5	• Final Presentation and paper

4 | Ethics and Safety

4.1 Ethics

As engineers working towards a brighter future and a better planet, we adhere to the IEEE Code of Ethics throughout all engineering practices. For the duration of the project and going forward during work in the real world, we will ensure integrity, responsibility, and ethical practices in our work environment. All members of our team are mutually respected and treated as such. We will work to ensure the ethical nature of our project is not tainted and it will likewise refrain from producing unethical outcomes for others. We do not foresee many points of ethical contention with our project, but that will not prevent us from remaining weary of any possible conflicts[7].

4.2 Safety

There are a few main safety concerns with our project. Namely with all electronics there is danger of electrical shock. As such we will work to create a non conductive housing for the electrical components to prevent injuries or accidents of this nature. This too comes with considerations for battery safety. We must be cautious when working with such components as they are prone to cause major accidents when not kept a watchful eye over. We will work to make sure batteries are in good, working condition when in use and stored away properly when not. This will all ensure a safe workplace for our team and others around us. We will also consider implementing temperature sensors and a hard limit switch to turn off the device in case the battery reaches hazardous conditions.

As a final safety concern, we recognize that there is a potential issue for people with long hair. Our project contains strong turning motors with crevesaces that allow for hair to get caught. Users with longer hair are at a higher risk for an accident of this nature. To prevent this as much as possible, we will work to minimize gaps in the hardware and warn users of this hazard when applicable.

5 | Parts List

Description	Quantity	Total Price	Link
Micro Gear Motor	6	$72	Link Link (alternative)
Motor Driver	6	$7.26	Link
I2C Expander	6	$8.16	Link
STM32F4 MCU	1	$10	Link
3.3V LDO	1	$1.6	Link
Battery	2	$20	Link

6 | References

[1] "The Inventor of the Rubik's Cube Took This Long to First Solve It." *CBCnews*, CBC/Radio Canada, 13 Mar. 2021, https://www.cbc.ca/radio/undertheinfluence/the-inventor-of-the-rubik-s-cube-took-this-long-to-first-solve-it-1.5945283#:~:text=Soon%2C%20Rubik's%20Cube%20become%20part,the%20pandemic%20has%20boosted%20sales.

[2] "God's Number Is 20." *God's Number Is 20*, http://www.cube20.org/.

[3] *How Much Do Electrical Engineer Jobs Pay per Hour? - Ziprecruiter* https://www.ziprecruiter.com/Salaries/Electrical-Engineer-Salary-per-Hour.

[4] *Description Pin Configuration - Diodes Incorporated*. https://www.diodes.com/assets/Datasheets/PI4IOE5V9535.pdf.

[5] *ZXBM5210 Description Pin Assignments - Diodes Incorporated*.

https://www.diodes.com/assets/Datasheets/ZXBM5210.pdf.

[6] “What We Offer.” *Illinois MakerLab*, https://makerlab.illinois.edu/pricingservices/#printing.

[7] “IEEE Code of Ethics.” *IEEE*, https://www.ieee.org/about/corporate/governance/p7-8.html.