Rubik’s Cube

Custom Project Final Report

Winter 2017

Andrew Apostol

Table of Contents

[**Introduction**](#_n9o44neiquk5) **2**

[**User Guide**](#_hnsjlybf9q1n) **2**

[**Hardware**](#_vqmbsuo5p330) **3**

[Parts List](#_xtqh2tzibtrr) 3

[Pinout](#_8yptvhjzojq8) 4

[**Software**](#_4oaokf9evhlv) **4**

[**Complexities**](#_sn48u4uktu3c) **7**

[Completed Complexities:](#_tp9dnsse4u9p) 7

[**Youtube Link**](#_u9cxbem9510r) **8**

[**Source Files**](#_iy8ww0k70pag) **8**

[**Known Bugs and Shortcomings**](#_grlzpb6vy2cq) **8**

[**Future work**](#_qbv6f31drpex) **8**

# Introduction

The rubix cube was invented in 1974 by Erno Rubik, a Hungarian sculptor and professor of architecture. A rubix cube is a 3x3 cube where each of the 6 faces is covered by nine stickers, each of one solid color. The cube’s rows and columns can be pivoted, changing the colors of each face. The objective is to return the cube to it’s original form of a solid color on each face. Today, there are still puzzle enthusiasts who compete against each other in solving the cube.

My project was to implement a 3x3 Rubik’s cube on an 8x8 LED matrix. This Rubik’s cube can be altered using the buttons on the SNES controller. The objective is exactly the same as a normal Rubik’s cube - to get each face to have one solid color.



# User Guide

Color Code:

Cyan = Yellow

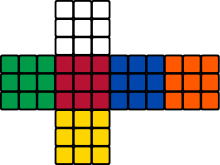
Seagreen = White

Orange = Purple

Red = Red

Blue = Blue

Green = Green



Note: The position of each Face in relation to each arrow is constant. For example, Up of Orange will ALWAYS be Seagreen.

Controls:

Left Arrow: Turn to Left Face

Right Arrow: Turn to Right Face

Up Arrow: Turn to Top Face

Down Arrow: Turn to Bottom Face

Start: Rotate Current Face clockwise

Select: Rotate Current Face counter clockwise

B + Left Arrow: Rotate Top Face clockwise

B + Right Arrow: Rotate Bottom Face clockwise

B + Up Arrow: Rotate Right Face clockwise

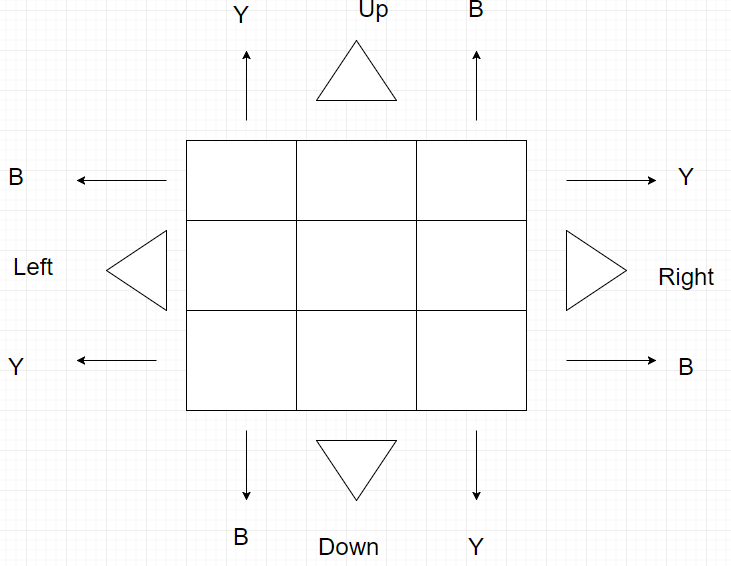
B + Down Arrow: Rotate Left Face clockwise

Y + Left Arrow: Rotate Bottom Face counter clockwise

Y + Right Arrow: Rotate Top Face counter clockwise

Y + Up Arrow: Rotate Left Face counter clockwise

Y + Down Arrow: Rotate Right Face counter clockwise



Note: B and Y must be pressed before the directional arrow when rotating non-current Faces.

# Hardware

## Parts List

* ATMega1284p microcontroller
* **8x8 LED RGB matrix**
* **SNES Controller**

## Pinout

PA0-PA3 connected to LED PIN16-19

PA4-PA7 connected to LED PIN28-31

PB5 connected to SNES PIN3

PB6 connected to SNES PIN1

PB7 connected to SNES PIN2

PC0-PC2 connected to LED PIN22-20

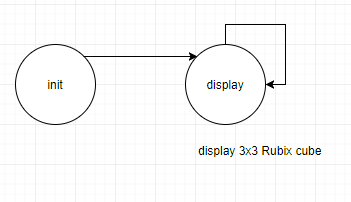
PD0-PD2 connected to LED PIN3-5

PD3-PD5 connected to LED PIN11-13

# Software

* AVR Studio 6

SM\_Display



SM\_Update

# 

# Complexities

## Completed Complexities:

* Outputting a 3x3 matrix of potentially 6 different colors on an LED matrix
* Rubik’s Cube Loic
* Reading SNES Controller Input

# Youtube Link

https://www.youtube.com/watch?v=xOlDdDdT01c

# Source Files

Main.c contains all code for the state machines, timing, and running the state machines.

# Known Bugs and Shortcomings

* It is almost impossible for somebody who is not well versed in solving Rubik’s cubes to be able to learn off of one face. Part of being able to understand a Rubik’s cube is being able to see how rotating one face changes the 4 faces around it.
* Understanding which button rotates what is confusing.
* There are some combinations of colors on a 3x3 matrix that dims a single LED.

# Future work

There are a few improvements that could make this better. First of all, I want to display the surrounding faces as well as the current one. This would make it easier to understand how each rotation affects the entirety of the cube. The is only one point of view for every face. For example, the top of orange will always be white. This is not always the case when playing with a real Rubik’s cube. Realistically, each surrounding face can be considered the “top” depending on how the user is holding the cube. That is why I also want to somehow connect each face within the code. Lastly, using the SNES controller is not very intuitive. I want to find a different controller or maybe a different combination of buttons that is easier to understand and play with.

Resources

<https://github.com/katherinegallaher/cs120b-Rhythm/blob/master/finalproj.c> for the read SNES controller function

<http://www.gamesx.com/controldata/snesdat.htm> for the SNES pinout and controller button to clock pulse table