University California Riverside

CS122A Final Project

Fall 2016

Cube Solver

Author:
Stanley Cohen

Professor:
Jeffery McDaniels

December 7, 2016

Abstract

The Cube Solver is a Rubik's Cube solver that uses a Raspberry Pi and OpenCV for reading in a cube. The solution is then found using Kociemba's algorithm and passed over to an Atmega1284 to solve the cube for the lazy layman.

1 User Guide

Turn on the Atmega1284 and ensure that the SPI lines are connected to the Raspberry Pi. Launch the *cube_reader.py* Python script on the Pi. Show the camera the sides of the cube listed in the terminal and press enter to lock in a side. Once all sides have been entered, type "done." Place the cube into the arms of the solver and press the button on the breadboard to begin running the solution!

2 Technologies and Components

- AVR Studio 6
- Atmega1284
- 8 Car Grade Solenoids
- 4 Stepper motors
- 8 Relays (to form H-Bridges for the solenoids)
- Raspberry Pi
- Pi Camera
- OpenCV

3 Demo Video

Demo video can be found here.

4 Source Code

4.1 Raspberry Pi

4.1.1 cube_reader.py

This script is used to read the cube via the Pi Cam with the help of OpenCV and then convert the 6x3x3 array of cubies into a cube string to be passed to Kociemba's algorithm. It will then convert the returned solution into a set of moves that can be executed on my machine (only L, R, F, B moves) and pass the moves over SPI to the Atmega.

4.1.2 myspi.cpp

This C++ code will send whatever value it's passed over the Raspberry Pi's SPI. The majority of the code is from MontaVista Software. I modified their template to fit my purpose.

4.2 Atmega1284

4.2.1 queue.h

Used to store the set of moves we need to execute. Allows us to receive multiple moves and hold them until a button is pressed so the user can have time to place the cube into the machine.

4.2.2 joystick.h

Used for debugging to put moves into the queue so the Raspberry Pi does not need to be on and hooked up at all times to test everything.

4.2.3 main.c

Listen for moves sent over SPI and add them to the move queue. Control the 4 stepper motors over 2 shift registers. Control the 8 solenoids to pull back the correct arm after executing a move. Execute all moves in the move queue on button press.

5 Wiring

Wire	Atmega	Pi
SPI MOSI	PB5	PIN19
SPI MISO	PB6	PIN21
SPI CLK	PB7	PIN23
GND	GND	PIN25
SHIFT1 RCLK	PC1	-
SHIFT1 SRCLK	PC2	-
SHIFT1 SRCLR	PC3	-
SHIFT1 SER	PC0	-
SHIFT2 RCLK	PC1	-
SHIFT2 SRCLK	PC2	-
SHIFT2 SRCLR	PC3	-
SHIFT2 SER	PC4	-
MOTOR1	SHIFT1[3:0]	-
MOTOR2	SHIFT1[7:4]	-
MOTOR3	SHIFT2[3:0]	-
MOTOR4	SHIFT2[7:4]	-
SOLENOID1 H-BRIDGE	PD[1:0]	-
SOLENOID2 H-BRIDGE	PD[3:2]	-
SOLENOID3 H-BRIDGE	PD[5:4]	-
SOLENOID4 H-BRIDGE	PD[7:6]	-
BUTTON	PA5	-
JOYSTICK	PA[1:0]	-