Computer Science
Senior Design Laboratory

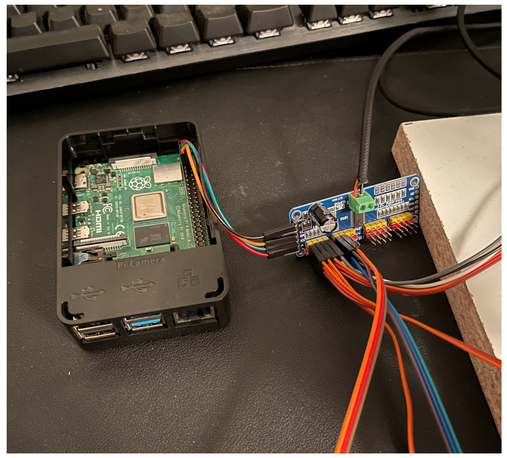
University of Kansas

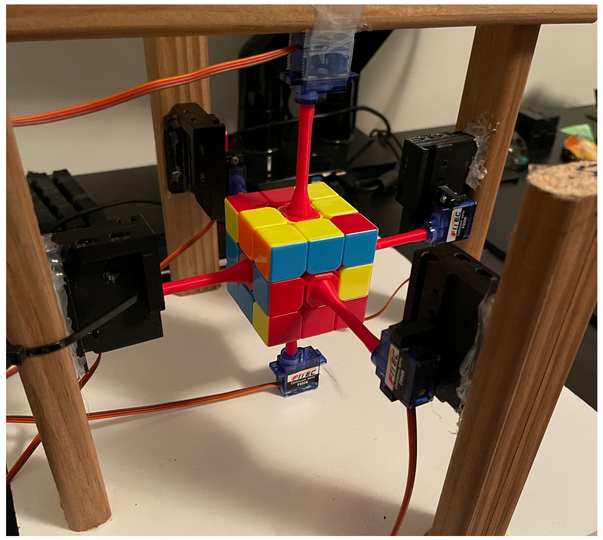
$Solved^3$ (Team 3)

Luis Belmontes (CS), Diego Garcia (CS), Karen Liu (CS), Ankeet Prasai (CS), Xingtong Zhang (CS)

Description & Purpose

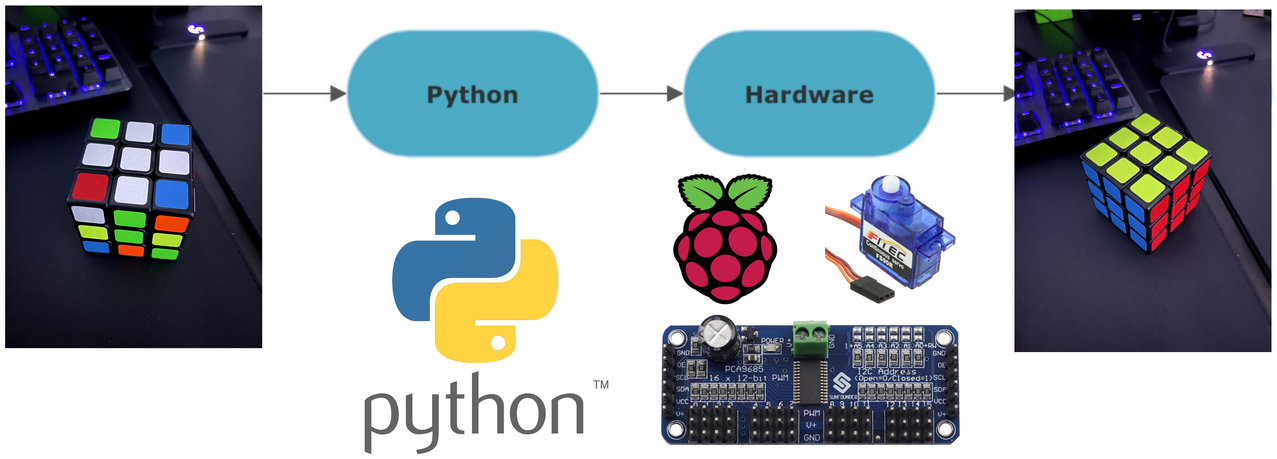
Description: A Rubik's Cube solver application that uses software and hardware to physically solve a cube.
Reason: We chose this project because we thought it was a great way to use the skills we already have and take it even further by implementing a hardware component. Hardware implementation isn't as common in a CS track, this project allowed us to extend our knowledge using hardware.
Purpose: The purpose of this project is to have a visual representation of the cube being solved. This helps beginners or even those unfamiliar with these puzzles to visualize the steps needed to get from a random scramble to the solved state.
Result: A Rubik's Cube solving algorithm that uses the F2L (a common beginner method). This algorithm is then visualized using the hardware components, solving the cube one move at a time





- Raspberry Pi + PCA9685
- 6 continuous servo motors
- 3D printed pivots to connect motors to the cube
- Used wood for the housing that was screwed/glued together

Design



Ethical & Intellectual Property Issues

Ethical Issues: Having a solver might discourage those who are trying to learn how to solve a Rubik's Cube as they might be intimidated by how efficient an automatic solver is. This may make them think that they would have to learn all the algorithms just to solve the cube, when only about 3 or 4 actual algorithms are needed for a beginner to solve a cube.

Intellectual Property Issues: For the first two layers, our solver gets each piece's position and performs moves to place it into the correct position without misplacing any others. This way our solver solves the first two layers in a more natural way like a human would. But for the last layer, using algorithms was something that we had to do as that is the way that the cube would be solved by a human. We used algorithms that were compiled by Feliks Zemdegs, a well-known speed-cuber. This made it easier for us to implement them into our solver.