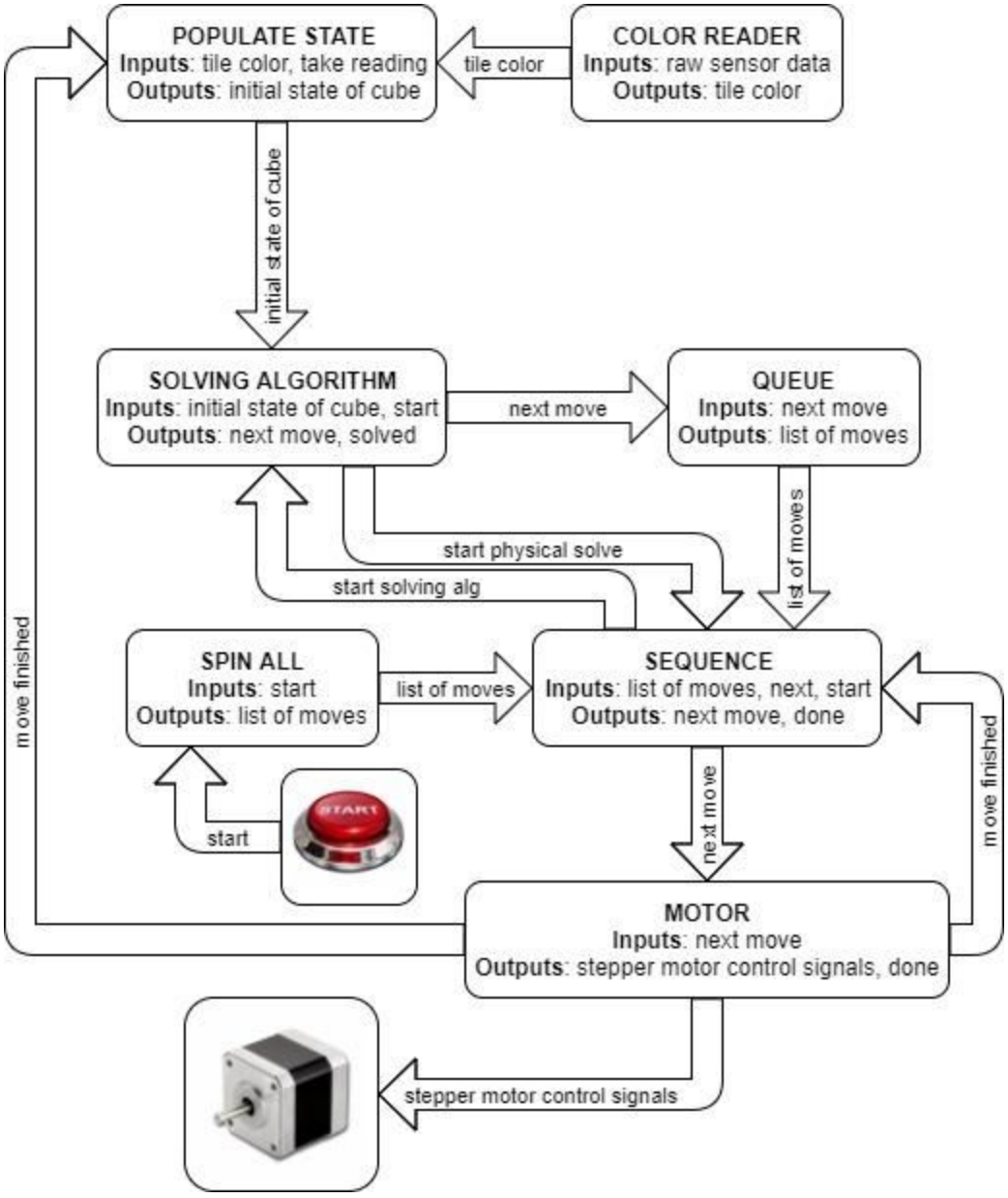
Rubik's Cube Solving Robot

Jacob Swiezy, Nathaniel Knopf
6.111 Fall 2017

Block Diagram



Descriptions of Modules

Color Reader

The color reader module will take the raw input of the TCS34725 RGB color sensor produced by reading a single sticker and map it to one of the six colors on the Rubik's Cube. This mapped color will be continuously outputted and wired as an input to the populate state module in order to determine the initial state of the puzzle.

Populate State

The populate state module will be responsible for determining the original state of the scrambled Rubik's Cube. The inputs to this module will be *tile color*, outputted from the color reader module, and a *take reading* signal from the motor module, signifying that the Rubik's Cube is not turning. When the *take reading* signal goes high, the populate state module will store the value of the *tile color* in its internal representation of the puzzle's state.

Once the internal representation of the Rubik's Cube has been completed, the populate state module will send the initial state of the Rubik's Cube to the solving algorithm module.

Motor

The motor module will translate moves outputted by the queue module into signals for the stepper motor drivers on the robot. The motor module will receive a *ready* signal, signifying that a new move is ready for the motors, as well as a value corresponding to the move to execute next. When the *ready* signal goes high the motor module will output the corresponding signals to the stepper motor drivers. Once the motors have been activated and turned according to the instructions given, a *done* signal signifying the motors have finished turning will be asserted.

Spin All

The spin all module will contain a sequence of moves that can be applied to the Rubik's Cube in such a way that every sticker is exposed to the TCS34725 RGB color sensors. As input, it will take a *start* signal, which goes high when the robot should begin the process of solving the Rubik's Cube. When the *start* signal goes high, the spin all module will output the list of moves necessary to determine the puzzle's state to the sequence module.

Solving Algorithm

The solving algorithm module will take the current state of the Rubik's Cube and a *start* signal as inputs. When the *start* signal goes high, the solving algorithm module will use an algorithm for solving the Rubik's Cube to determine the next move that should be applied to the puzzle. This move will be outputted to the queue module. The solving algorithm will also output a

solved signal that goes high when the puzzle has been reduced to a solved state, as well as the state of the Rubik's Cube after the new move has been applied. This outputted state will be wired into the solving algorithm module as the new state for calculation of the next move.

The algorithm used by this module to solve the Rubik's Cube will be based off a beginner's method solution to the Rubik's Cube. This method breaks solving the puzzle down into seven different steps. The solving algorithm module will contain a finite state machine, where the state represents which step of the method the module is currently completing. The module will then use this state combined with the current state of the Rubik's Cube to determine the next move to execute.

If this algorithm proves to be too complex to implement on the FPGA, we will investigate alternative solutions to this problem. One possible solution is to have a computer running a more powerful Python script that determines the moves to execute on the Rubik's Cube. The FPGA will then receive serial communications from the Python script and store the sequence of moves produced by Python in the queue module.

Queue

As the solving algorithm module outputs the moves that need to be applied to the Rubik's Cube, the queue module will receive these outputted moves as an input and keep a running queue of these moves. This list of moves will be outputted by the queue module and wired into the sequence module. This list will be appended to as new moves are produced by the solving algorithm module such that the sequence module is always receiving a list of all the moves calculated so far.

Sequence

The sequence module will receive a list of moves from the queue module, as well as a move finished signal from the motors module indicating that the motors have finished moving and are ready for new instructions. When this signal goes high, the sequence module will send the next move on the inputted list from the queue module to the motor module and send a new move signal to the motor module indicating that the instruction to the motor module has changed.

Additionally, during the portion of the solve in which the robot is determining the starting state of the Rubik's Cube, the sequence module will receive a list of moves from the spin all module. Its functionality will then be identical to that given above, with the module passing new moves on to the motor module when prompted by the move finished signal.

External Components

TCS34725 Color Sensor

In order to read the initial state of the cube without user intervention, the robot needs to be able to determine the color of each of the cube's tiles. This part of the robot will be implemented via the TCS34725 RGB color sensor. The sensor uses special filters in conjunction with a photodiode array to return 16-bit values for Red, Green, and Blue. Using these values, we can determine the color of a tile on the cube. By rotating the cube through a specific sequence of moves, we can use just two of the color sensors to obtain the state of the entire cube. The sensor is mounted to a breakout board including an onboard LED to ensure consistent lighting conditions. The FPGA will communicate with the sensor via the I2C communication protocol.

Stepper Motor

The robot will utilize six stepper motors to physically implement the solving algorithm on the cube. Each stepper motor will control the rotation of one face of the cube. The stepper motors will only be attached to the center tile of each face of the cube, so that each face can rotate freely. The bipolar stepper motors each have 4 wires, 2 wires per coil. By energizing the coils in a predetermined sequence, the stepper motors will step in the desired direction. The FPGA will interface with the stepper motors via stepper motor drivers which are described in the following section.

A4988 Stepper Motor Driver

The stepper motor driver converts logic signals into appropriate signals on the 4 wires of a stepper motor to drive the motor in the desired direction. The motor driver requires 2 control inputs from the FPGA to drive each stepper motor, step and direction. A rising edge on the step input causes the stepper motor to rotate in the direction specified by the corresponding input. Because it is only possible for us to rotate one face of the cube at a time, our implementation will tie all of the direction inputs together allowing us to reduce the number of FPGA outputs from 12 to 7 (6 step outputs, 1 direction output).

Rubik's Cube

For our Rubik's Cube, we will use a Dayan ZhanChi Stickerless Speed Cube. This is a type of Rubik's Cube engineered to have low turning friction and a low probability of locking up - both of which are useful properties for reducing strain on the stepper motors. This Rubik's Cube will also be treated with an oil based lubricant to decrease turning friction.

Robot Frame

The robot frame will be constructed of custom designed, laser-cut acrylic pieces. The robot frame is responsible for holding the stepper motors which actuate the cube. The frame will be

constructed of 6 individual pieces each holding a stepper motor. The frame will only consist of 2 uniquely designed parts (4 side pieces, 2 top plates). The frame will be designed in Solidworks before construction to ensure that all of the pieces fit together properly before construction.

Stepper Motor to Cube Adapter

To physically connect each of the stepper motors to the appropriate Rubik's cube face, we will 3D print 6 small adapter pieces. The adapters, which will be designed in Solidworks, will attach only to the center tile of each face of the cube. The adapters will have a friction fit with the shafts of the stepper motors.