ECE 445

Fall 2023

Senior Project Design Document

Automatic Puzzle Solver

Team 71

Eric Chen

Conor Devlin

Alex Kim

TA: Angquan Yu

1 Introduction

1.1 Problem

In the realm of recreational activities assembling jigsaw puzzles remains a popular pastime. There are many times when we are in the middle of solving a jigsaw puzzle and get stuck. For some people like children, manually solving jigsaw puzzles can be a daunting task. Many pieces look similar to the human eye, and this can be frustrating when trying to do what is deemed a relaxing activity.

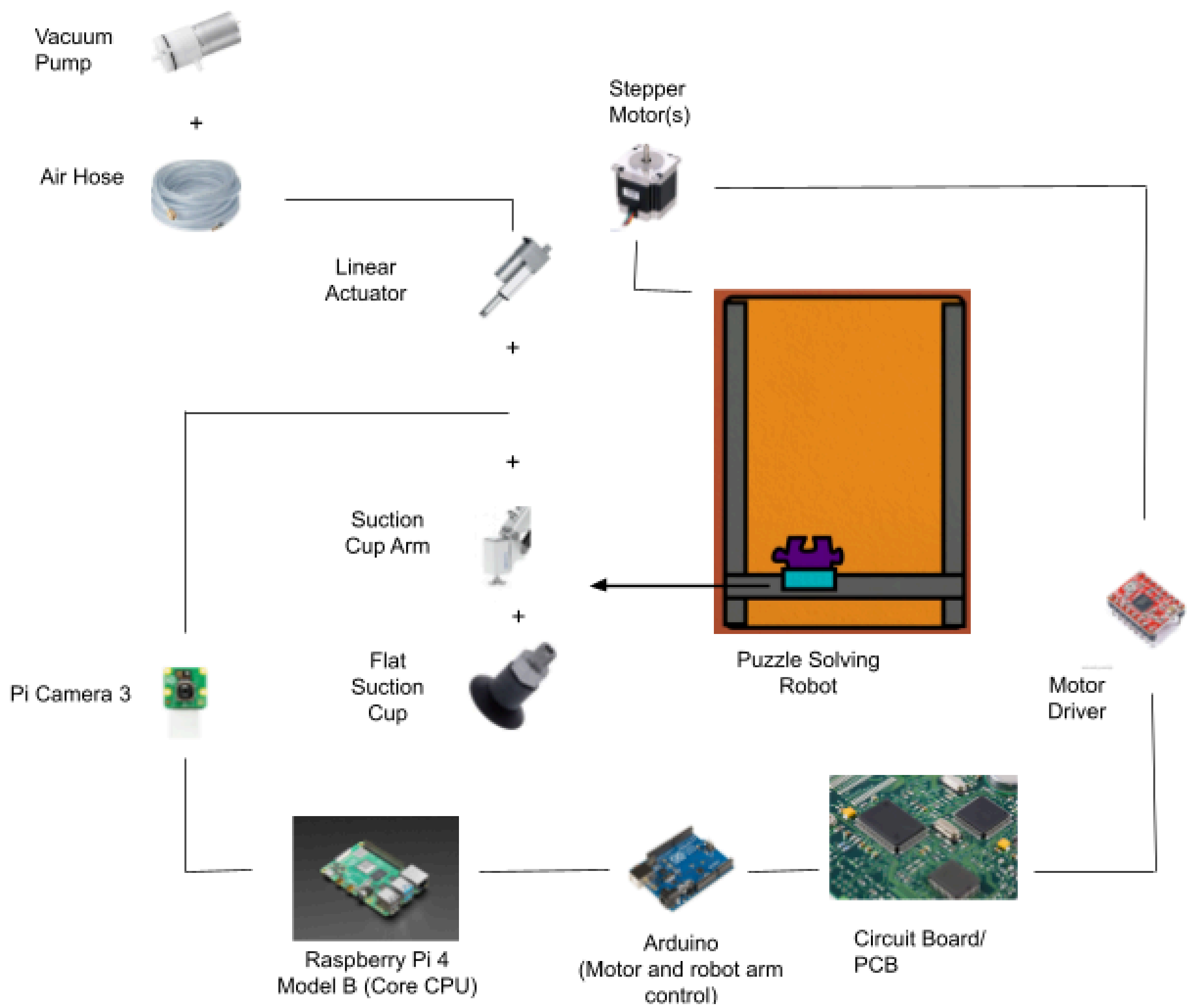
1.2 Solution

Our project aims to develop an innovative solution for puzzle enthusiasts and those facing challenges in manually assembling jigsaw puzzles. The project involves designing and implementing an Automatic Jigsaw Puzzle Solver equipped with a precision-controlled robotic arm attached to a suction device. The objective is to create a user-friendly device capable of autonomously solving puzzles of varying complexities. Our goal will be to be able to solve a 3x3 puzzle.

The setup will be as follows, a robot machine overlay will be placed over the jigsaw puzzle. We will have a series of pulleys and belts to move a robotic arm capable of extending in z direction via a linear actuator to pick up puzzle pieces. This robotic arm will be moved in the x and y direction using the belt and pulley system which is powered by stepper motors (reference visual aid). The robot will start by scanning each puzzle piece using an OpenCV-compatible camera. Our circuit board which is attached to a computer will then calculate exactly where each puzzle piece should be connected. Using computer vision, we will then grab each puzzle piece and move it to the desired location to complete the puzzle.

The robotic arm uses a linear actuator, suction cup, and camera to pick up the puzzle pieces. The linear actuator is needed so we can move puzzle pieces over each other and not drag the pieces along the table. The suction cup will be small enough to form a seal on the puzzle piece and it will be powered by a vacuum pump through a pipe. The camera will be used to identify the pieces and precisely place them together.

1.3 Visual Aid

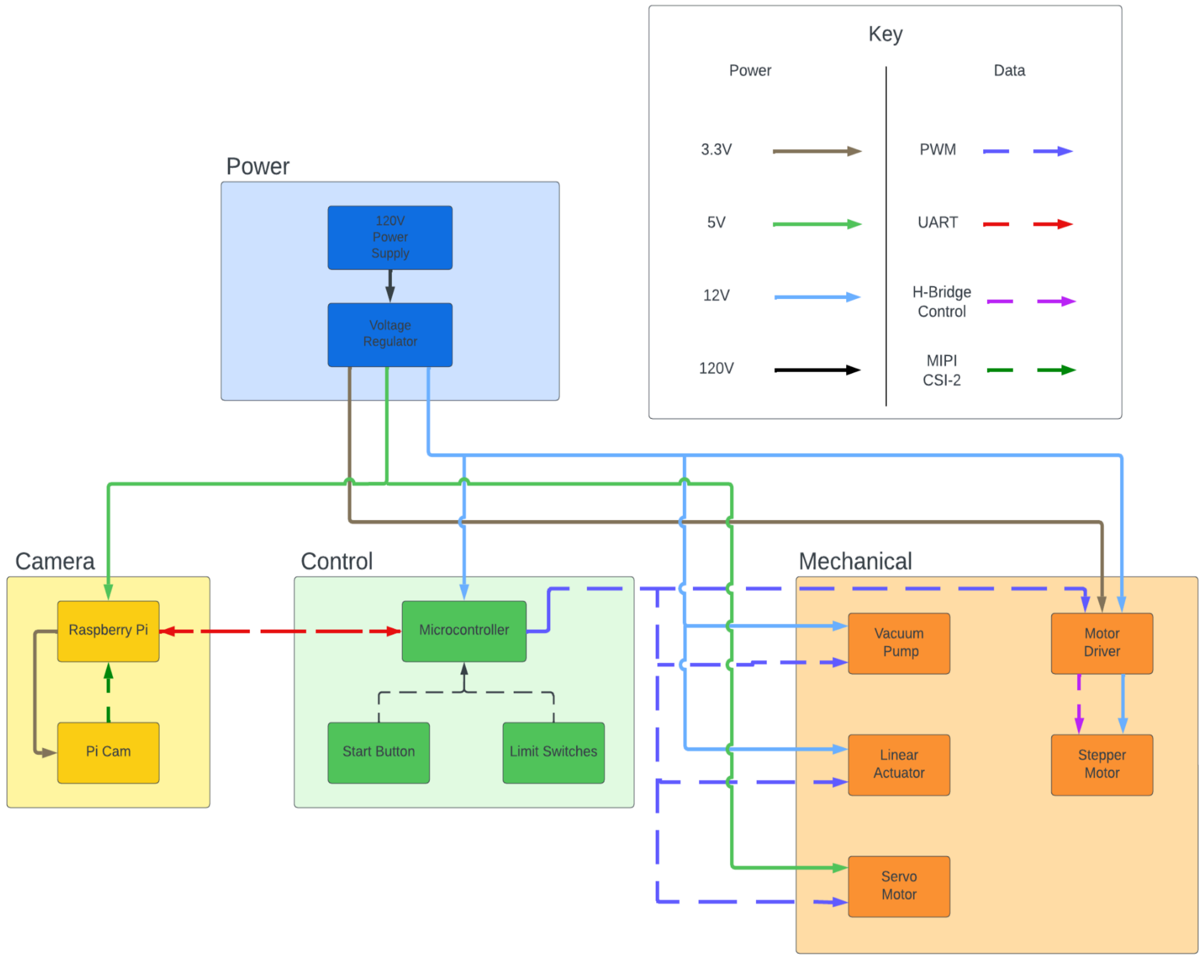


1.4 High-level requirements list

- The mechanical movement will be precise to 1 mm of movement
- We can correctly identify where individual puzzle pieces are located on the puzzle
- We can complete the whole 3x3 piece puzzle in 7 minutes

2 Design

2.1 Block Diagram



2.2 Physical Design

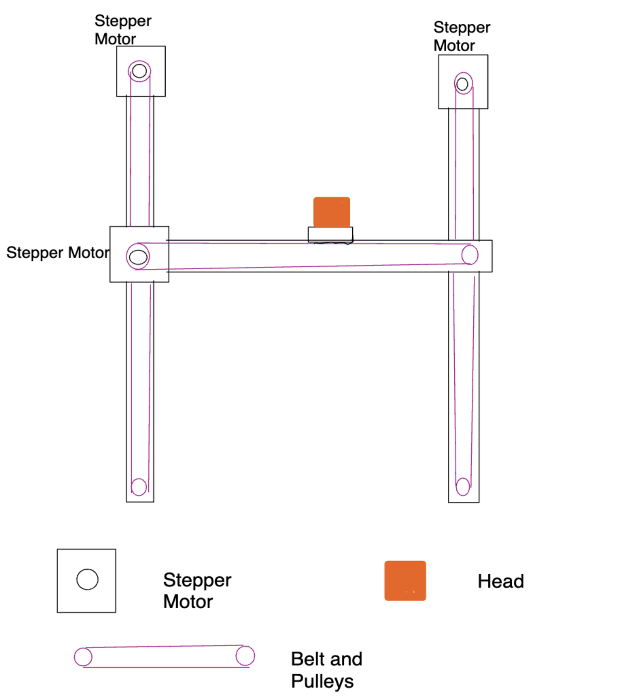


Figure 1: Robotic Movement Scheme Overhead View

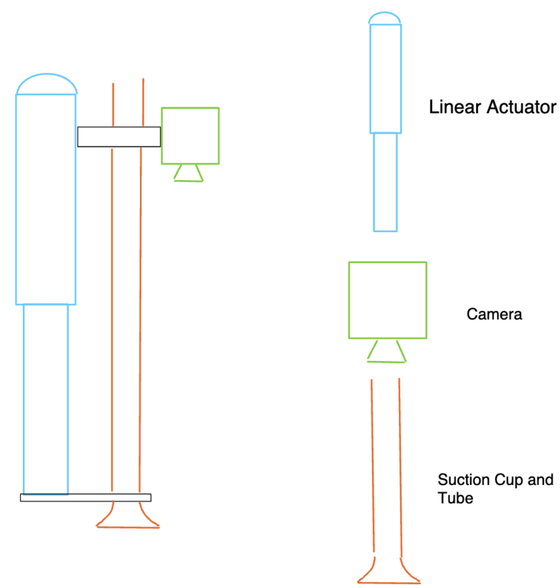


Figure 2: Robotic Arm Side View

2.2 Subsystem Overview

2.2.1 Power Subsystem

The subsystem will contain the circuitry to convert power to the needed voltage levels to power our puzzle-solving robot machine and all of its parts. This system will distribute power to the appropriate parts of the robot such as the stepper motors (powering the belts and pulleys to move the arm), the linear actuator to extend the robot arm in the z-direction, and all the components of the PCB and our camera used for computer vision.

The kit provided to us by the department can plug into the wall and supply 5V and 12V of DC power in relation to common. We will need 3.3V to power the logic in the motor controllers. We will make a power bus with 3.3V by dividing the 5V with resistors. The logic in the motor controllers will have a current from 1 to 20 microamperes.

Requirements	Verification
• The 3.3V power bus will have the correct voltage	• Use a voltmeter to measure the difference between the 3.3V bus and common. The voltage should be 3.3V +/- 0.2 V

2.2.2 Control Subsystem

The control subsystem provides the microcontroller, a ATMEGA 328 PB, necessary to efficiently operate all mechanical components including the motor drivers, linear actuator, and vacuum pump. The limit switches will be used to zero our stepper motors. We will use the start button to start and stop the machine.

Upon start up, the arduino will spin the stepper motors to move the car, which is the mechanical part that will be attached to the belt to physically move things, to the limit switch. The microcontroller will zero the stepper motors, which means the microcontroller will start recording the position by counting the steps.

Requirements	Verification
• The microcontroller will be able to record the position of the suction cup, above the table	• Turn on the machine so the stepper motors get zeroed • Move to a designated point on the board, our point will be 100 steps in the x direction and 100 steps in the y direction or (100,100), mark this position of the suction cup on the table • Move the stepper motors in a manner that is similar to how the puzzle solver will operate. Move 123 steps in the x direction and 112 in the y direction or (+123,+112), then (-50,+50), then (+73,-41), then (-7,-19). • Move the suction cups to the original position by moving (-139, -102) and mark the position of the suction cup over on the table • Verify that the marks are in the same place +/- 5 millimeters
• The microcontroller should be able to receive a signal from the raspberry pi and react	• Have the raspberry pi send a signal to lower the linear actuator • Within 3 seconds of the pi signal being

Requirements	Verification
accordingly	sent the microcontroller should lower the linear actuator

2.2.3 Camera Control Subsystem

Our camera subsystem will consist of the Raspberry Pi Camera Module 3 along with its connections to our off-board module, which will be a Raspberry Pi. For the off-board controller specifically, we will use a Raspberry Pi 4 (4GB version). Our camera will be 25 × 24 × 12.4 mm in size and around 4 grams in weight. It has a 12MP, 120 Degree Wide pre-attached lens. We will use the highest resolution available which is 2304 × 1296p56 for video and still image capture. While it is stated that this camera comes with a 120-degree wide-angle lens, the true horizontal field of view is more towards 102 degrees. This camera will be directly attached to the Raspberry Pi 4 (RP4) and will stream/save video and still images directly to the RP4 to be used with OpenCV for our puzzle image detection software stack. We will be using UART protocol communication, a type of serial communication, between the RP4 and the Arduino Uno (on-board microcontroller). Specifically, the RP4 GPIO 14 (TX) pin will be directly connected to the D0 (PD0, RX) pin on the Arduino. The baud rate of the RP4 will be configured to be at 115200. The special bootloader configuration file on the RP4 will ensure BOOT_UART flags are enabled on launch. The RP4 will be sending commands to the Arduino such as next coordinates to move the robot arm to and when to pick up or put down a puzzle piece. To ensure that the Camera Subsystem is fulfilling its responsibilities for sending transmissions to the onboard control system, a requirements & verification table can be found below.

Table 3: Camera Control Subsystem – Requirements & Verification

Requirements	Verification
• When the camera subsystem detects a fatal communication failure in the ways of not receiving data from the Pi Camera 3 for 10 seconds. • Simulate a sudden camera disconnection.	• Ensure that the entire camera subsystem (RP4 + Pi Camera) and currently running in an idle state (reference software flow chart). • Cut power/turn off the Pi Camera source. Confirm that the RP4 enters a fatal communication state.
• The camera system's idle state should receive consistent feed from the Pi Camera 3.	• Verify that the camera has been initialized and is properly connected to the RP4. • Verify that the correct resolution and frame rate are still running after 5 seconds in the idle state.
• The camera system should send an error message and stop camera recording when any memory or storage errors are encountered.	• Fill up current RP4 storage with test files and verify that streaming is cut from the Pi Camera when storage is full. • Verify that a correct corresponding error message is sent to the user through the verbose terminal.
• Camera subsystem should be able to handle data corruption during transfer to control system block.	• Remove receiver to transmission wire between RP4 and Arduino mid-way through control operation. • Verify that a correct corresponding error message is sent to the user through the verbose terminal. • Verify that the operation is shut off and handled accordingly.
• The camera subsystem should be able to perform real-time video processing. • We want to aim for 95% accuracy object detection.	• After ensuring that the video capture is available and working, verify that the processed real-time video stream meets processing benchmarks. (Edge detection, object tracking, etc.) • Verify that we have 95% accuracy in puzzle recognition compared to real truth.
• The camera system should be able to send basic commands and coordinates to the control subsystem	• Verify that the UART connection is stable and the baud rate is correct between both subsystems. • Execute user tests and send subsystem

Requirements	Verification
	data to the motor control subsystem; verify that the motor control subsystem reacts accordingly ○ Example: new pose/coordinate move, emergency stop control, etc.

2.2.4 Mechanical Subsystem

The mechanical subsystem executes physical actions based on instructions from the control system to manipulate the arm. This subsystem contains stepper motors controlled by the motor driver to precisely control the movement of the belt system. The linear actuator provides vertical movement for the arm, which in turn allows a flat suction cup to pick up the puzzle pieces. The suction will be provided by an air hose connected to the vacuum pump.

Requirements	Verification
• The vacuum pump and suction cup will be able to pick up puzzle pieces.	• A puzzle piece will be held up to the button of the suction cup. • The vacuum pump will be turned on and the puzzle piece should be held to the suction cup with just suction.
• The linear actuator should be able to reach puzzle pieces on the table, with the suction cup, when fully extended.	• A puzzle piece will be placed on the table directly below the linear actuator and the linear actuator will be lowered. • The suction cup connected to the linear actuator will touch the puzzle piece
• The stepper motors should receive an amperage between 0.85A-1.36A from the drivers to ensure proper usage.	• Turn on the machine and run all three motors • Use an ammeter to measure the current to the motors from the drivers. • Ensure it is within range and adjust voltage accordingly
• To protect the board from destructive LC voltage spikes we will put a large (at least 47 μF) electrolytic capacitor	• Place >47μF electrolytic capacitor across VMOT and ground • Turn on the machine and monitor voltage spikes across motor power and ensure it

Requirements	Verification
across motor power (VMOT) and ground somewhere close to the board.	is under the 45V maximum
● A heat sink will be placed on the motor driver in order to cool the IC.	● Attach heat sink to IC ● Turn on the machine and run 7 minutes worth of movement with the motors. ● Use a thermocouple to measure the heat of the IC throughout the run to keep temperatures under 60°C

2.3 Software Design

One core part of our project involves the use of OpenCV to detect puzzle pieces and analyze them in order to determine the proper orientation and position in which each individual piece should be rotated and moved to. Our inputs to the camera system are simply the live photo and video feed that will be fed into our software stack.
An HSV algorithm is used in combination with OpenCV to process the color images fed into the computer from our camera subsystem. A simple flow chart of the software stack can be seen below (Figure 10).

Once the camera is initialized, it will capture frame-by-frame data in an executive loop while in our “Capture Image” phase. I will be using a standard capture rate which will cap out at around 30 frames per second, any more and there risks unnecessary performance drops in exchange for a only a slightly smoother system. After capturing a frame from our live video feed, the computer will be using cv2’s cvtColor functionality to convert the entire frame into the HSV color space while also creating a mask for each color space that we are trying to identify. The mask (cv2.inRange) contains the upper and lower bounds for each individual color on the HSV scale for use of identification. A local database of color masks will be stored on the computer (see Figure 3) with all nine corresponding colors that will be on our finished puzzle grid (see example grid, Figure 2). Once our color is located, the system will then find the contours of the detection areas (cv2.findCountours) to identify our puzzle piece. To account for false positives and have better filtering, only the contour piece with the largest area will be selected as the “puzzle piece”. This should cover most cases as the view of our camera is not significantly wide and will be quite close to the table/viewing area. There would be a very small chance that the largest (red, blue, green, etc.) piece in the view of the camera is not a puzzle piece. The detected puzzle piece will then be highlighted in frame and labelled according to the color that is determined by the HSV algorithm. This process will then continue running in an executive loop until a user interrupt occurs to send a robot move command to the ESP32 microcontroller which will then operate the motor drivers of our robot arm accordingly.

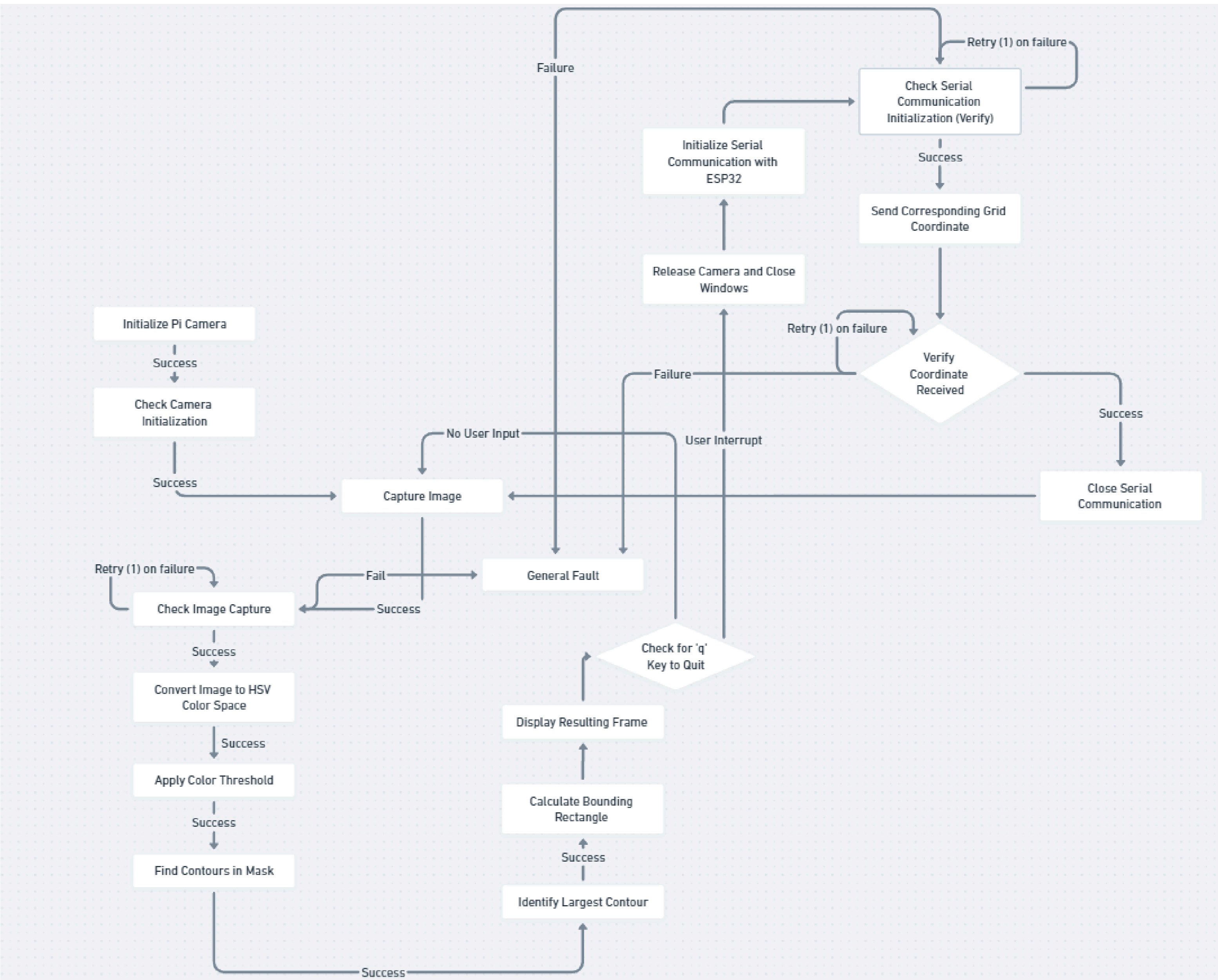


Figure 10: Camera Subsystem Communication Flow Chart

Color	Upper Boundary	Lower Boundary
Red	[10, 255, 255]	[0, 120, 70]
Blue	[130, 255, 255]	[110, 50, 50]
Light Green	[80, 255, 255]	[40, 40, 40]
Orange	[15, 255, 255]	[5, 50, 50]
Yellow	[30, 255, 255]	[20, 100, 100]
Violet	[160, 255, 255]	[130, 50, 50]
Magenta	[170, 255, 255]	[140, 50, 50]
Dark Green	[90, 255, 255]	[60, 50, 50]
White	[180, 30, 255]	[0, 0, 200]

Figure 11, HSV Color Mask Database

2.4 Tolerance Analysis

Mechanical Precision: One of our high-level requirements is that we will be able to move the suction cup with 1mm precision in any direction. We will be moving the arm with the suction cup attached to it with a belt and pulley system powered by stepper motors. We have chosen a stepper motor with 200 steps per revolution. The steps are the base unit of movement of the stepper motor. The distance of one revolution is determined by the radius of the pulley the belt is attached to.

$$2\boldsymbol{\pi}r = C$$

$$C/steps = distance\ of\ step$$

$$2\boldsymbol{\pi}r/200 = 1mm$$

To get this level of precision we will need the radius of the pulley to be less than 31.83mm. We plan to have pulleys with a radius of only 10-20mm.

Vacuum Suction: We will be using a vacuum suction cup to pick up the puzzle pieces. The vacuum holding the puzzle piece will need to be strong enough to hold the puzzle piece when we are lifting it off the table since that is when the vertical acceleration will be greatest. The force needed to lift the piece will be

$$F = m * (g + a)$$

where m is the mass of the puzzle piece, g is the acceleration of gravity, and a is the acceleration of the puzzle piece being lifted. The large 3x3 puzzle we are going to use is listed on Amazon with a total weight of 4.6oz or 0.5111oz per piece. This is equivalent to about 14.49 grams which we will round to 15 grams or 0.015kg. The acceleration of gravity is 9.81 m/s^2. We approximate that we will pick up the piece with a max acceleration of 0.5 m/s^2.

$$F = 0.015 * (9.81 + 0.5) = 0.15465$$

We will need 0.15465 N to pick up the piece. The pressure of the vacuum needed is determined through this equation

$$P = F/A$$

The pressure we need is based on the size of the suction cup we choose. We are thinking of using a suction cup with a .2 inch diameter which is a 5.08mm diameter and 2.54mm radius. The area of the suction cup is

$$A = \pi r^2$$

so the area of the suction cup will be 0.00002026829m^2. With this force and area, the pressure of the vacuum pump will need to be 7630.14 Pa. The vacuum pump we plan to use will have a max vacuum pressure of 420 mmHg or 56 kPa. The parts we plan to use will be able to lift the puzzle pieces.

3 Cost and Schedule

3.1 Cost Analysis

The total cost for parts as seen below before shipping is $115.44. Estimating 5% shipping cost adds another $5.77 and 10% sales tax adds $11.54. We are all in Computer Engineering and the expected starting yearly salary of a CompE graduate from UIUC is $109,176. This means we can expect a salary of $52.49/hr per team member. $52.49 x 2.5 x 60 hours x 3 team members = $23,620.50 in labor cost. The machine shop quotes a price of $50/hr for 40 hours a week for 2 weeks = $4,000. This comes out to be a total cost of $27,753.25.

Description (Part #)	Manufacturer	Qty	Price	Links
Raspberry Pi Camera Module 3 Wide Lens (5658)	Raspberry Pi (Adafruit)	1	35.00	Link

Raspberry Pi 4 Model B - 4 GB (4296)	Raspberry Pi (Adafruit)	1	55.00*	Link
Arduino Uno Rev3 (A000066)	Arduino	1	27.60***	Link
Limit Switch 10-pack (3-01-1546)	HiLetgo	1	5.99	Link
Micro Air Pump - DC 12V Micro Vacuum Pump (B07FGFPKNS)	DEWIN	1	10.49	Link
2" Stroke Micro Electric Linear Actuator (UYG2206LAS50MM-1)	UYGALAXY	1	35.99	Link
Stepper Motor Driver 3 Pack (DRV8825)	WWZMDiB	1	7.99	Link
Nema 17 Stepper Motor (17HS4401S)	Usongshine	2	19.98	Link
Standard Servo Motor (HS-311)	Hitec	1	13.49***	Link
Suction Cup	McMaster-Carr	1	9.89**	Link
3m Silicone Tube 2mm ID x 4mm OD Flexible Silicone Rubber Tubing Water Air Hose Pipe Transparent (B08H1ZD5VZ)	Gikfun	1	8.18**	Link
DC DC CONVERTER 0.8-6V (LMZ12008TZ/NOPB)	Texas Instruments	1	17.94***	Link
			115.44	

* = Purchased individually ** = Purchased by the machine shop *** = From lab or shop

3.2 Schedule

1. **Schedule:**

Include a time-table showing when each step in the expected sequence of design and construction work will be completed (general, by week), and how the tasks will be shared between the team members. (i.e. Select architecture, Design this, Design that, Buy parts, Assemble this, Assemble that, Prepare mock-up, Integrate prototype, Refine prototype, Test integrated system).

Week	Class Task	Conor	Eric	Alex
2/26	Design/Peer review PCB planning and review Order parts	Get stepper working with arduino, pcb planning	Code outline and libraries we will use	Pcb planning and design
3/04	PCBway orders (pass audit) Teamwork evaluation, Final machine shop design	Finalize machine shop design, Linear actuator, vacuum pump, and servo work as intended	Arduino and Pi can communicate	Submit PCB design
3/18		Can recognize color and edges with openCV	Can recognize color and edges with openCV	Test PCB and see if it works
3/25	Individual progress reports	Connect hardware to machine and be able to move all machinery	Work on CV algorithm for puzzle piece recognition	Continue testing PCB and work on CV algorithm for puzzle piece recognition
4/01		Integrate PCB with Pi	Finish puzzle piece finding algorithm	Integrate PCB with Pi
4/08		Work out bug in preparation for mock demo	Work out bug in preparation for mock demo	Work out bug in preparation for mock demo
4/15	Mock demo Team contract fulfillment	Prepare for mock demo/ make puzzle more complicated	Prepare for mock demo/ make puzzle more complicated	Prepare for mock demo/ make puzzle more complicated
4/22	Final demo Mock presentation	Prepare for demo and finishing touches to project	Prepare for demo and finishing touches to project	Prepare for demo and finishing touches to project
4/29	Final presentation	Finish paper	Finish paper	Finish paper

	Final paper Lab checkout Lab notebook	and presentation	and presentation	and presentation

3 Ethics & Safety

There are a few ethical and safety concerns that we must consider when building our automatic puzzle solver.

3.1 Ethical Concerns

● Privacy (IEEE 8.7, ACM 1.2): Camera data might capture sensitive information unintentionally.

○ Mitigation: We will minimize data collection, anonymize data wherever possible, obtain informed consent, and adhere to data privacy regulations.

● Transparency and Explainability (IEEE 8.3, ACM 2.7): Understanding the robot's decision-making process is crucial.

○ Mitigation: We will develop explainable algorithms, provide visualizations of the robot's reasoning, and allow for human intervention if needed.

3.2 Safety Concerns

● Physical Safety (IEEE 8.1, ACM 2.1): Moving parts and potential pinch points pose risks to people and property.

○ Mitigation: Ensure safe operation around people, adhering to safe distances and clear communication protocols.

● System Reliability (IEEE 8.8, ACM 2.3): Malfunctions or errors could lead to safety hazards.

○ Mitigation: Rigorous testing, fail-safe mechanisms, and regular maintenance are crucial.

References:

Amazon.com, Inc. (n.d.). Air Pump Electric Treatment Instrument. Amazon. https://www.amazon.com/Air-Pump-Electric-Treatment-Instrument/dp/B07FGFPKNS

Amazon.com, Inc. (n.d.). UYGALAXY Stroke Electric Linear Actuator. Amazon. https://www.amazon.com/UYGALAXY-Stroke-Electric-Linear-Actuator/dp/B0B4BJD4HS/ref=sr_1_4?keywords=mini%2Blinear%2Bactuator&qid=1707237976&sr=8-4&th=1tuator/dp/B0B4BJ5GLL/ref=sr_1_4

ArduCam. (n.d.). Raspberry Pi Pinout. https://www.arducam.com/raspberry-pi-camera-pinout/

Handson. (n.d.). 17HS4401S Datasheet. DatasheetsPDF.com. https://datasheetspdf.com/pdf-file/1310364/Handson/17HS4401S/1

Hitec RCD USA. (n.d.). HS-311 Servo Motor Datasheet. University of Texas at Austin. https://users.ece.utexas.edu/~valvano/Datasheets/ServoHS311.pdf

IEEE. (n.d.). IEEE Code of Ethics. https://www.ieee.org/about/corporate/governance/p7-8.html

Pololu Corporation. (n.d.). Pololu - A4988 Stepper Motor Driver Carrier. Pololu. https://www.pololu.com/product/2133

Texas Instruments. (n.d.). LMZ12008 SIMPLE SWITCHER® Power Module Datasheet. Texas Instruments. https://www.ti.com/lit/ds/slvsa73f/slvsa73f.pdf?ts=1708583617637&ref_url=https%253A%252F%252Fwww.google.com%252F

Texas Instruments. (n.d.). LMZ12008 SIMPLE SWITCHER® Power Module Product Information. Texas Instruments. https://www.ti.com/general/docs/suppproductinfo.tsp?distId=10&gotoUrl=https%3A%2F%2Fwww.ti.com%2Flit%2Fgpn%2Flmz12008

University of Illinois Urbana-Champaign. (n.d.). Salary Averages. Electrical and Computer Engineering at Illinois. https://ece.illinois.edu/admissions/why-ece/salary-averages