AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF ENGINEERING
AND INFORMATION TECHNOLOGY

COMPUTER ENGINEERING DEPARTMENT

Hardware Graduation Project
Sudoku Solver

SUBMITTED BY

Mohammad Masood
Yacoob Assi

SUPERVISOR

Dr. Sufyan Samara

Presented in partial fulfillment of the requirements for the Bachelor's degree in Computer Engineering.

August 31, 2023

Acknowledgment

We would want to express our gratitude to our families, who have stood by us and helped us along the road from the start of our life's journey, as well as to our cherished friends. Additionally, we would like to express our gratitude to An-Najah National University's Department of Computer Engineering for their assistance during the academic years and to Dr. Sufyan Samara for overseeing our graduation project during the summer semester.

DISCLAIMER

Contents

Abstract

This project showcases a Sudoku solver that combines image processing and robotic manipulation to automate the process of solving Sudoku puzzles. By utilizing a camera connected to a Raspberry Pi, the system captures an image of a Sudoku board, extracts the numbers and their positions from the image, and employs an algorithm to solve the puzzle. A robotic arm equipped with a pen attachment then fills in the solved numbers on the physical board. This integration of image processing, robotics, and algorithmic solving demonstrates the potential of combining technologies to automate complex puzzle-solving tasks.

1. Introduction

In a world characterized by rapid technological advancement, harnessing the integration of diverse fields emerges as an innovative solution to complex challenges. This project serves as a tangible testament to the power of merging multiple domains, presenting an innovative solution to crack a Sudoku puzzle using the inverse search algorithm. Upon solving the puzzle, the solved numbers are transmitted from a Raspberry Pi to an Arduino, which operates a robotic arm to physically write the numbers on the actual board.

Sudoku, the beloved numerical puzzle, has captured the interest of puzzle enthusiasts around the globe. Its intricacy lies not only in the logical deductions required for its solution but also in the manual process of transferring solutions onto the physical board. This project takes a bold step by leveraging the capabilities of modern technology to streamline this process.

At its core, the project relies on the integration of hardware and software components. The camera seamlessly integrated with the Raspberry Pi serves as the visual interface to the Sudoku puzzle. The system captures an image of the Sudoku board, effectively converting the actual puzzle into a digital format. The magic happens within the complex image processing algorithms that extract the numerical distribution from the puzzle, paving the way for the Backtracking algorithm to take control.

The true innovation emerges when the inverse search algorithm begins to unveil the enigmatic puzzle patterns. Operating as a maestro of digital logic inferences, this algorithm dismantles the puzzle, gradually revealing its concealed solution. However, the crescendo of this automated symphony comes with the entry of a robotic arm, equipped with a pen attachment. Just as if it extracts the solution from the digital realm and materializes it into reality, this robotic manipulator carefully replicates the solved numbers onto the physical board.

Yet, this project is not merely a technical accomplishment; it's a testament to human ingenuity and curiosity. It's a journey of exploring the realm of the possible through technology, blurring the boundaries between imagination and reality. By showcasing the sequential collaboration between image processing, robotics, and programming, this project underscores the potential to simplify intricate tasks and elevate our understanding of automation capabilities.

we delve further into the details of this project, we embark on a journey that celebrates not only innovation but also sheds light on the beauty of collaboration across diverse disciplines. The subsequent sections will delve into the technical details that enabled this interdisciplinary integration, highlighting the challenges faced, the solutions devised, and ultimately, the achieved results. Through this exploration.

2. Literature Review

Indeed, our project stands in a landscape where prior software endeavors have already honed in on tackling the Sudoku puzzle through programmed solutions via backtracking. Simultaneously, separate endeavors have focused on employing software algorithms to analyze images and extract numerical data from them. Moreover, we've seen hardware initiatives, like 2D CNC machines, translate images into tangible drawings on paper.

Before we embark on the intricate technical details of our undertaking, it's pivotal to lay the theoretical groundwork on which our fusion of diverse technologies rests. Sudoku, an intricate logic-based numerical puzzle, has captivated both mathematicians and puzzle enthusiasts for decades. The aim is to populate a 9x9 grid with digits in a way that each column, row, and 3x3 subgrid accommodates all digits from 1 to 9.

The puzzle's complexity arises from the need to deduce the optimal placement of numbers using logical deduction. Conventional Sudoku solving involves iteratively applying a set of rules and logic, progressively narrowing down possibilities for each cell.

Our project takes these foundational principles and infuses modern technology to enhance the process. The integration of a camera-equipped Raspberry Pi to capture the puzzle's image injects computer vision and image processing techniques into the equation. Insights from prior work in computer vision allow us to extract pertinent details from images, such as grid identification, number detection, and position determination.

Furthermore, introducing an inverse search algorithm to solve Sudoku puzzles showcases a spirit of innovation. Inverse search algorithms, commonly utilized in domains like cryptography, reverse-engineer from the desired solution to deduce the initial inputs. Applying this concept to Sudoku, our algorithm systematically reveals the puzzle's solution.

Robotic arms and automation have already undergone thorough exploration in fields like manufacturing and assembly. In our project, we adapt these technologies to the realm of puzzle-solving, underscoring automation's potential to simplify intricate tasks. Previous research in robotic manipulation provides insights into kinematics, control systems, and end-effector interactions, all pivotal in faithfully transcribing the solution onto the physical board.

By drawing from these diverse disciplines and existing research, our project synthesizes a unified framework that harmonizes computer vision, algorithmic reasoning, and robotics. This innovative amalgamation not only promises to solve Sudoku puzzles but also demonstrates the broader scope of interdisciplinary cooperation and automation in addressing complex challenges. Subsequent sections will delve into the nuanced technical intricacies of orchestrating these components, unveiling the hurdles overcome and the inventive solutions devised. Through this exploration, we aspire to ignite fresh avenues for technological fusion and problem-solving across domains.

3. Constraint

1. Hardware Limitations: Designing the system within the capabilities of the chosen hardware components, such as Raspberry Pi and Arduino, is essential. The computational power, memory, and interfaces of these devices might restrict certain functionalities or affect real-time processing.

Discussion: The selected algorithms and approaches were optimized to run efficiently on the hardware. This involved choosing lightweight image processing techniques and data structures that could perform adequately within the constraints.

2. Generate gcode and sending to arduino: The issue at hand is that the hardware we've utilized, resembling a CNC printer, doesn't comprehend numbers like a matrix or an image does. Instead, it only understands G-code. While there are specialized programs available for converting various formats into G-code, we'd prefer not to rely on these programs every time we operate the robot.

Discussion: The solution we implemented involves using the Inkscape software initially. We create the numerical designs in Inkscape, convert them to G-code, and store each generated G-code file for a specific number in a single folder. We've also developed a custom code that, upon sending to the Arduino, examines the upcoming number from the matrix.

Based on this number, the Raspberry pi sends the corresponding G-code file to draw that specific number. Additionally, there's a separate file dedicated to directions. The appropriate direction is sent based on the number, allowing the arm to move to the next location.

If the number is 0, the system refrains from drawing and merely instructs the arm to move.

3. Real-Time Processing: The system involves capturing and processing images in real time. Any delays or latency in processing could affect the accuracy of the solution and the robotic arm's movements.

Discussion: Image processing algorithms were optimized for real-time processing. Techniques like parallel processing and minimizing unnecessary computations were employed to reduce processing time.

4. Robotic Arm Precision: The accuracy and precision of the robotic arm's movements are crucial to accurately transcribe the solution onto the physical board. Any errors in positioning or control could lead to incorrect solutions.

Discussion: The robotic arm's kinematics and control algorithms were fine-tuned to ensure accurate movements. Calibration procedures were implemented to mitigate any inaccuracies.

5. Resource Availability: Availability of resources like power supply, physical space for the setup, and required components could impact the implementation and usability of the system.

Discussion: Resource planning was essential to ensure the availability of necessary components, power sources, and adequate space for the hardware setup.

6. Budget Constraints: Budget limitations might restrict the choice of components or technologies, affecting the overall functionality and performance of the system.

Discussion: Components and technologies were selected based on their cost-effectiveness while still meeting the project's objectives. Open-source software and affordable hardware were preferred whenever possible.

7. Motor drivers instead of shields: Using shields in our project is prohibited by the instructions of the project supervisors, so we have used a4988 drivers for stepper motors 4401.

By identifying and addressing these design constraints, the project team was able to make informed decisions and develop a system that could perform effectively within the defined limitations. Each constraint required careful consideration during the design and implementation phases to ensure a successful outcome.

4. Methodology

4.1 Overview

The successful implementation of this project relies on a comprehensive and systematic methodology that brings together the various components of computer vision, algorithmic reasoning, and robotics.

4.2 Equipment's and Components

4.2.1 Arduino UNO

The Arduino Uno is an open source board based on the Microchip ATmega328P microcontroller, and it's a very important component of our project. There are digital IIO and Analog Input Pins 14, 6 and others .

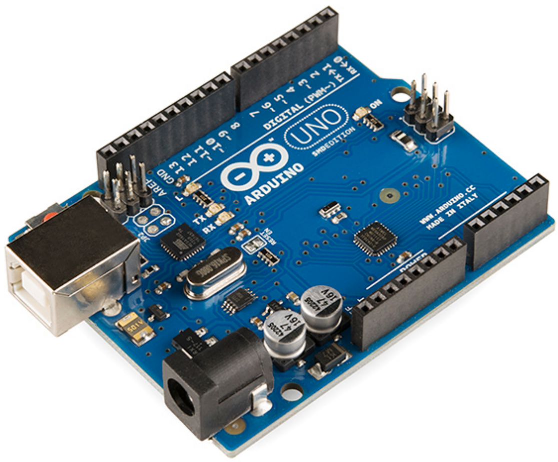

Figure 4.1: Arduino Uno.

4.2.2 Raspberry Pi 3 Model B

the raspberry pi 3 is single board computer, we uses it to access complex libraries that is essential for the camera to analyze using image processing (OpenCV), and we used it to capture an image through the camera and perform necessary image processing operations. Subsequently, we extracted the numbers from the image and applied the Backtracking algorithm to solve the Sudoku puzzle. Afterward, we transmitted the results to the Arduino in the form of customized G-code commands for each number that requires printing.

Figure 4.2: Raspberry Pi 3 Model B.

4.2.3 Camera module

This camera is used to connect to the raspberry pi to be able to capture image and video stream,it is connected on the Camera Serial Interface of the raspberry pi. We employed the camera in our project to capture images of the Sudoku board. Then, we perform the required image processing operations using the Raspberry Pi, which is connected to the camera.

Figure 4.3: Camera module.

4.2.4 Stepper Motor 4401

The stepper motor is an apparatus that moves in a sequential motion, one step at a time. For applications where precise position control is desirable, like 3D printers, laser cutting or picking and placing equipment, it can usually be applied in a number of ways. This NEMA 17HS4401 bipolar stepper motor has a 1.8º step angle (200 steps/revolution) with 40N.cm (56oz.in) holding torque.

Figure 4.4: Stepper Motor 4401.

In our project we have used two stepper motors one for x-axis and one for y-axis

4.2.5 Stepper Motor Driver A4988

For the control of stepper motors in a variety of applications, the A4988 Driver is an interchangeable driver. The A4988 has the ability to control a stepper motor with great accuracy and efficiency when using it in combination with an Arduino UNO microcontroller.

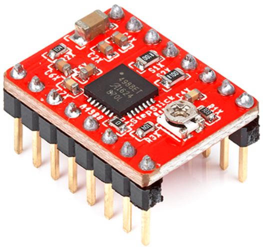

Figure 4.5: Stepper Motor Driver A4988.

4.2.6 Servo Motor SG90

Little, light, and powerful. A smaller version of a regular servo that can rotate about 180 degrees (90 in each direction). These servos can be controlled with any servo code, hardware, or library. Good for novices who wish to move objects without developing a motor controller with feedback and gearbox, especially because it can fit in tight spaces. Hardware and 3 horns (arms) are included.

Figure 4.6: Servo Motor SG90.

We have been using it to control the z-axis. We connect the pen to its white piece and we send a command to move it up, the pen goes up and so on.

4.2.7 12V Power Supply

Stepper motor and Arduino needs 12v to work, so we take the power supply that provides 12v and connect it to stepper motors and arduinos.

Figure 4.7: 12V Power Supply.

4.2.8 Breadboard

We have used two breadboards in our project, one has 2 stepper motor drivers .

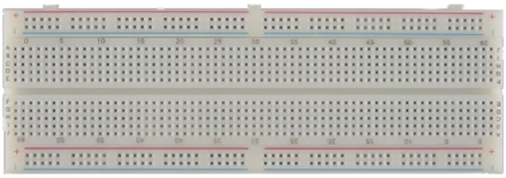

Figure 4.8: Breadboard.

4.2.9 Linear Guide Rail HGR20 and Linear Motion Guides HGH20CA

It is used for movement in each axis.

Figure 4.9: Linear Guide Rail HGR20 and Linear Motion Guides HGH20CA.

4.2.10 Pulleys and belts

We have used 2 meter belts one meter for each axis and 4 pulleys 2 for each axis, in each axis there is 1 pulley on each side to make the linear guide motion moves.

Figure 4.10: Pulleys and belts.

4.2.11 Wires

We used wires to be able to connect the components to each other, so we can provide necessary signals for required parts.

Figure 4.11: wires.

4.3 Implementation

The following sections outline the step-by-step approach used to achieve the project's objectives.

4.3.1 Build the Structure

First, we built the structure of our project:

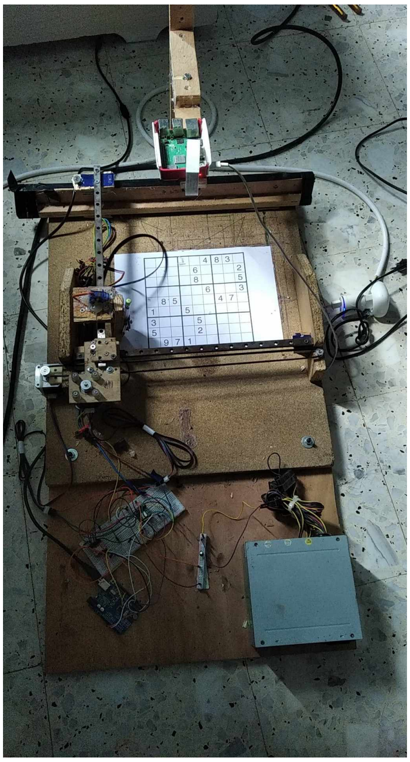

Figure 4.12: project design.

We establish each axis using linear guide rail and linear motion guides. and establish each motor on its axis and establish the pulleys with belt for each axis , then we placed a stand for the Raspberry Pi in a suitable position to capture an image of the board that the project will solve. Then, we placed the Pi and the camera in their appropriate places.

4.3.2 Test the movement

We test the movement by installing simple code on Arduino to the movement of motors.

4.3.3 Took a picture of the board:

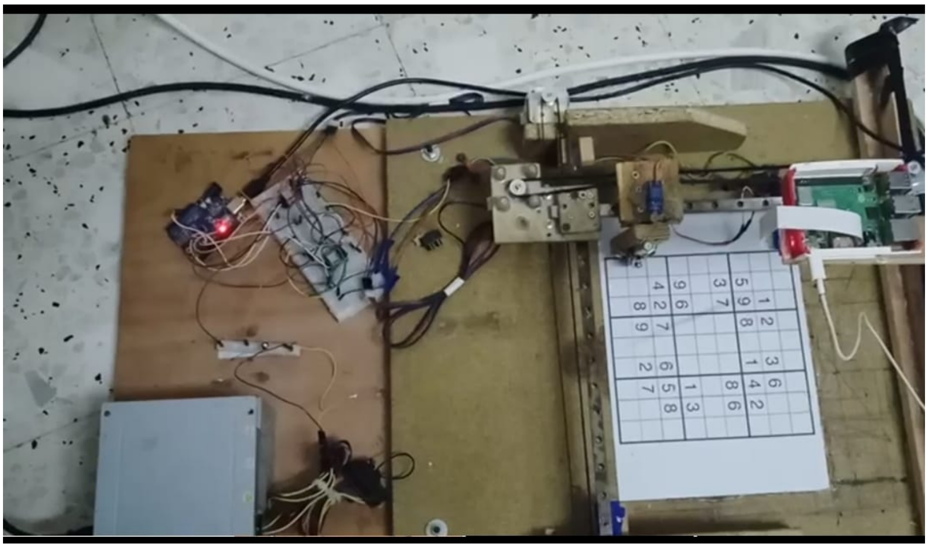

Figure 4.13: took picture.

4.3.4 Image processing:

1. Image Acquisition and Preprocessing: The process begins with the Raspberry Pi equipped with a camera capturing an image of the Sudoku puzzle. This image is subjected to preprocessing techniques, such as edge detection, thresholding, and noise reduction, to enhance the visibility of the puzzle grid and numbers.

2. Grid Detection and Extraction: Computer vision algorithms are employed to detect and extract the grid structure of the Sudoku puzzle from the preprocessed image. This involves identifying the intersections of the grid lines and segmenting the puzzle into individual cells.

3. Number Recognition:
The segmented cells containing numbers are further processed using optical character recognition (OCR) techniques. These techniques analyze the images of the numbers to convert them into digital text representations.

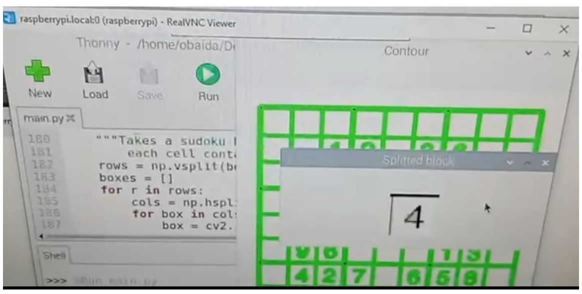

Figure 4.14: Number Recognition.

4.3.5 Backtracking Algorithm :

Backtracking is an algorithmic technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time .
. The extracted numbers are fed into the inverse search algorithm, which aims to deduce the original configuration of the puzzle. This algorithm employs logical reasoning and constraint satisfaction techniques to iteratively fill in the blank cells of the Sudoku grid until a solution is found.

4.3.6 Robotic Arm Interaction :

Once the solution is found, the Raspberry Pi directs the solved numbers to the Arduino, which controls the robotic arm. The system is capable of recognizing the specific General G-code (G-code) associated with each number, supplied to the Arduino by the Raspberry Pi. The robotic arm, equipped with a pen attachment, is then guided to write the solved numbers on the actual Sudoku board.

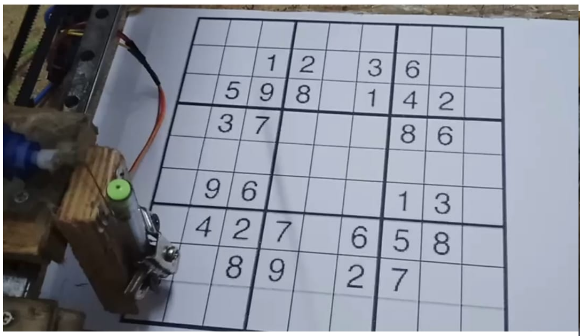

Figure 4.15:

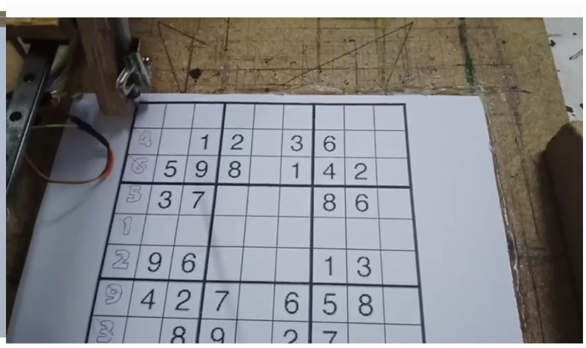

Figure 4.16:

We utilized Inkscape software to generate specific G-code for each number. Then, we employed the UGS software to define the arm's motion direction. Subsequently, a code was developed that retrieves the number from the solution matrix. Based on the number, the appropriate G-code file is sent to the Arduino, and the arm is directed according to the motion data obtained from UGS. The arm then proceeds to the next number. If a square is empty, the arm moves to the next number without performing any writing action. In this case, the value in the matrix is 0.

4.3.7 Iterative Improvement:

Based on the analysis, iterative improvements are made to enhance the accuracy and efficiency of the system. This could involve refining preprocessing techniques, optimizing algorithm parameters, or fine-tuning the robotic arm's control.

4.4 Programming languages

4.4.1 Python

We utilized Python on the Raspberry Pi for tasks such as image capture, image processing, solving Sudoku puzzles, and sending G-code commands to the Arduino..

4.4.2 C++

We utilized C++ to program the Arduino, enabling it to receive G-code and control the stepper motors and servo motor for drawing purposes.

By following this methodology, the project aims to showcase not only the technical feasibility of integrating these technologies but also the synergy between different disciplines. The intricate interplay of computer vision, algorithmic reasoning, and robotics culminates in a tangible demonstration of how innovation can arise from unexpected connections between fields.

5. Results And Discussions

Our findings hold great promise, with the algorithm's efficiency showcasing remarkable performance. The execution of image analysis and numeral extraction demonstrated good results, particularly in cases of higher resolution images. Regarding the movement of the robot, it exhibited a high degree of satisfaction by accurately inscribing numbers in designated positions, even despite minor deviations on occasion.

In summary, we can concisely affirm that the Backtracking algorithm displayed outstanding performance in successfully reaching the correct solution. In terms of image analysis, improvements in image clarity led to faster and more accurate solutions. Concerning the movement of the robot's arm and the precision of numeral inscription, we surmounted challenges and achieved a range of accuracy from good to excellent.

In a broader perspective, our project yielded a Sudoku Solver Robot that captures images of Sudoku boards, performs image analysis, extracts numerals from images, applies the Backtracking algorithm to these numerals, and ultimately transmits the outcome to the Arduino. The Arduino then guides the arm to complete the missing numbers in the Sudoku puzzle.

This achievement is a testament to the dedication and effort poured into designing and implementing the system. These results serve as a robust foundation for potential future applications grounded in the same technological underpinnings and innovative approaches. This success underscores the potential of amalgamating diverse disciplines to craft an exceptional solution for a conventional technical challenge.

Figure 5.1: Result

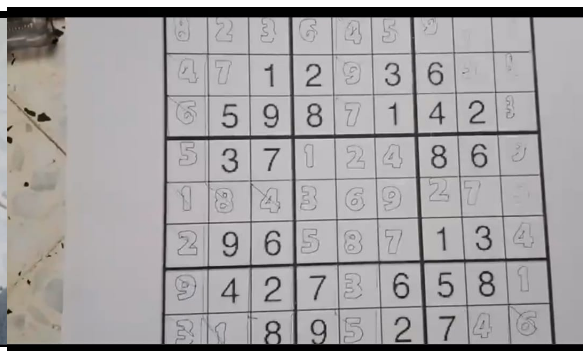

Figure 5.2: final result

6. Conclusion

In a rapidly evolving technological landscape, this project serves as a testament to the power of interdisciplinary integration in tackling complex challenges. The successful implementation of a system that combines image processing, algorithmic intelligence, robotics, and user interaction underscores the potential of harnessing diverse fields to create innovative solutions.

The project's journey from capturing a Sudoku puzzle through image processing, solving it using the inverse search algorithm, and finally transcribing the solution onto a physical board through a robotic arm highlights the synergy of technology components working in harmony.

Through this endeavor, the boundaries between the digital and physical worlds are blurred. The manipulation of data in the digital realm finds expression in the precise movements of the robotic arm as it carefully prints solutions onto the physical board.

However, this project is more than just a technological feat; it is a testament to human ingenuity and curiosity. It represents an exploration of the realm of possibilities, challenging the line between imagination and reality. By showcasing the orchestrated collaboration between image processing, robotics, and programming, the project emphasizes the potential to simplify complex tasks and elevate our understanding of automation.

As the project delved into the technical details of this integration, it shed light not only on innovation but also on the beauty of collaboration across diverse domains. The subsequent sections elaborated on the technical intricacies enabling this fusion of disciplines, illuminating the challenges faced, the solutions devised, and ultimately, the achieved outcomes.

This exploration seeks to unveil the future possibilities that arise when we dare to intertwine the threads of technology in unexpected ways. As technology continues to advance, this project serves as a reminder that creativity, collaboration, and the willingness to push boundaries are essential drivers of progress.

In conclusion, this project exemplifies how disparate technologies, when thoughtfully combined, can result in a harmonious symphony of automation, pushing the envelope of what is achievable and inspiring new horizons of technological innovation.

7. References

https://2u.pw/LSCCds (Arduino Documentation).

https://2u.pw/EZyW8Y (a4988 Stepper motor driver doc).

https://2u.pw/1sUScH (Nema 17 4401 stepper motor doc)

http://www.ee.ic.ac.uk/pcheung/teaching/DE1$_{E}E/stores/sg90_{d}atasheet.pdf$ (Servo motor S90)
https://www.raspberrypi.com/products/raspberry-pi-3-model-b/ (Raspberry Pi 3 Model B)
https://www.geeksforgeeks.org/backtracking-algorithms/ (Backtracking Algorithm)