Automatic cake decorator
ECE 445 Senior Design

By
James Zhu, Muye Yuan, Rui Gong

5/1/2024
Team 47
Instructor: Viktor Gruev
TA:Jason Zhang

Abstract

This report presents the design of an automatic cake decorator. The project integrates the mechanical structure, the self-designed PCB, and the robust program that allows users to customize shapes or texts and draw them on the cake. This document focuses on the engineering processes involved, including the block diagram representation, detailed design methodology, cost analysis, and performance. Our project met all predefined high-level requirements, demonstrating functionality, and reliability. This successful implementation illustrates the potential for similar automated food decorating systems in various industrial settings.

Contents

1. Introduction

1.1 Problem

The current challenge lies in manual cream decoration on cakes, prompting the need for an automated solution. Traditional methods often result in variations in cream thickness, coverage, and overall quality due to the nature of manual application. This not only demands skilled and creative workers but also leads to increased production costs and the potential for human errors. Moreover, labor costs can be a significant factor in the overall production costs.

1.2 Solution

We decided to make an automatic cake decorator machine, which puts creams with fancy shapes and curves on the top surface of the cake. The automation is not only presented in the "putting cream on the cake" process, but also the design: it could adapt the decoration according to the size and shape of the cake, eliminating the need to design or modify manually. And it provides different styles to choose from according to user preference.

The mechanical structure of the machine resembles that of a cartesian robot, or a 3D printer, which is two perpendicular sliding rails (powered by linear motors) connected to each other, able to move its tips to arbitrary x-y positions. A large syringe with cream inside is mounted at the tip, leaving a trail of cream when pushed by a motor. A sonar may be installed to detect the height of the cake, and another DOF on the z axis of the machine can be added so that the syringe tip can be adjusted up and down automatically to near the surface of the cake. The user connects to and operates the control flow of the machine on a laptop.

1.3 Virtual Aid

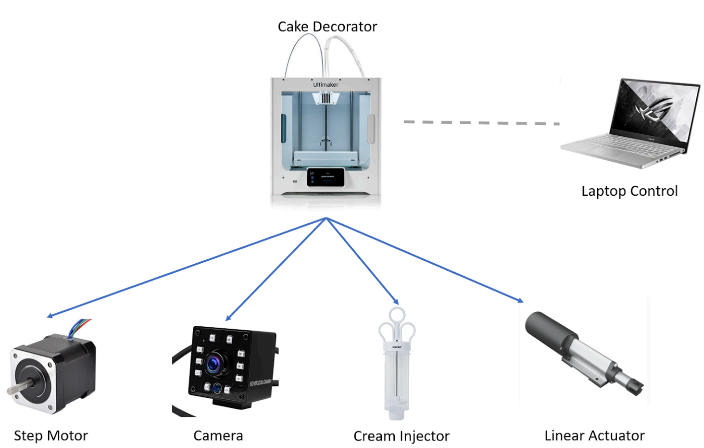

Fig. 1 Visual Aid

1.4 High Level Requirements

- Cake shape and edge is detected successfully for more than 90% of the trails, within 5 mm of range from the edge in the camera view, and not distracted by any other objects, or confused by patterns already presented on the cake.
- In the user program, at least four decoration line shapes or styles for the user to choose from.
- The movement of the motors are accurate enough to navigate the cream injector, with a maximum of 1 cm of deviation from the designed trajectory.

2 Design

2.1 Block Diagram

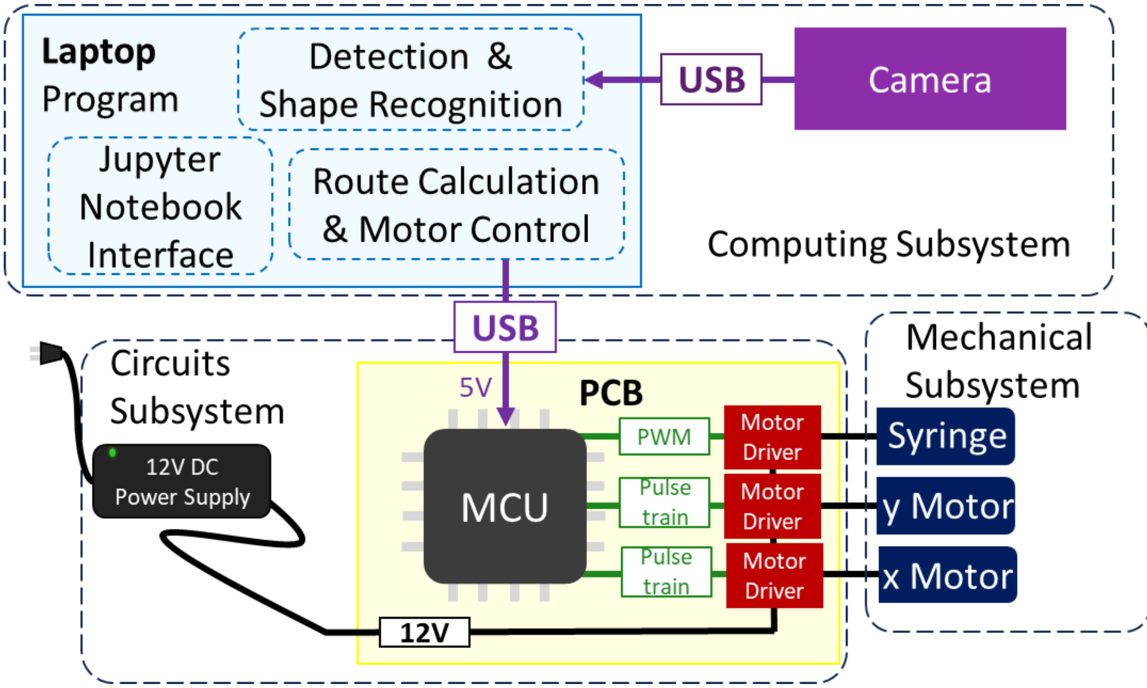

Fig. 2 Block Diagram, design process of specific subsystems will be elaborated later

2.2 Computing Subsystem

We proposed the parts of Shape recognition, User Interface, and Route calculation for computing subsystems in the design document. For the user interface, the computer program is coded in python, it's natural to make it into a jupyter notebook. This creates an easy-to-use environment, as users can navigate through the program's functionalities, execute various segments referencing the headings, and change the parameters according to the needs. This serves as a replacement for a dedicated UI system.

For recognition and route calculation, the program is constructed in a structured function flow, following this sequence:

1. Calibration for mapping between pixel and physical coordinates
   We take a picture of a standard chessboard paper. It should be placed on a specific location, thus we know the coordinates of each corner in the physical world. A built-in recognition algorithm with also find the pixel coordinates of these corners. The correspondence of these four corner points allow us to solve the pixel to physical

mapping using perspective transform. This should only be runned once if the relative position between the camera and the cake decoration area is changed.

2. Food recognizer:
   This part starts with taking a picture of the cake. Then a machine-learning based algorithm, namely foodSAM[1], is applied. It is shape recognition + position detection all-in-one, where the exact outline of the shape could be found directly. This algorithm also understands the semantic meaning of the object, meaning that it won't confuse with anything else other than food.

3. Generating decoration styles:
   The program offers two modes for creation of decoration curves on top of the area of the cake. First, an algorithm is built from scratch, to segment the recognized food shape into grids, and a customizable blueprint could be copied and mapped into each grid, creating a continuous decoration curve. The second mode allows the user to load a gcode vector file created by InkScape[2] + MIGrbl extension[3]. It then gets automatically fitted to the center of the shape just recognized. Users could customize the size of it. The program also provides an interface to connect various curves generated.

4. Sampling the curve, transforming to physical coordinates, and generating stepper movement according to the step distance of the motors.

5. Encoding movement and transmit to the machine
   We designed a dedicated protocol to encode the movements, allowing us to package it efficiently, and send it to the controller board via serial port. The embedded software on the controller board would decode the movements and execute them.

2.3 Mechanical Subsystem

The structure of the machine, as you can see from Fig. 3, resembles a cartesian robot, or a 3D printer, which is two perpendicular sliding rails (powered by motors) connected to each other, able to move its tips to arbitrary x-y positions. A large syringe with cream inside is mounted at the tip, extruding the cream uniformly when pushed by a motor.

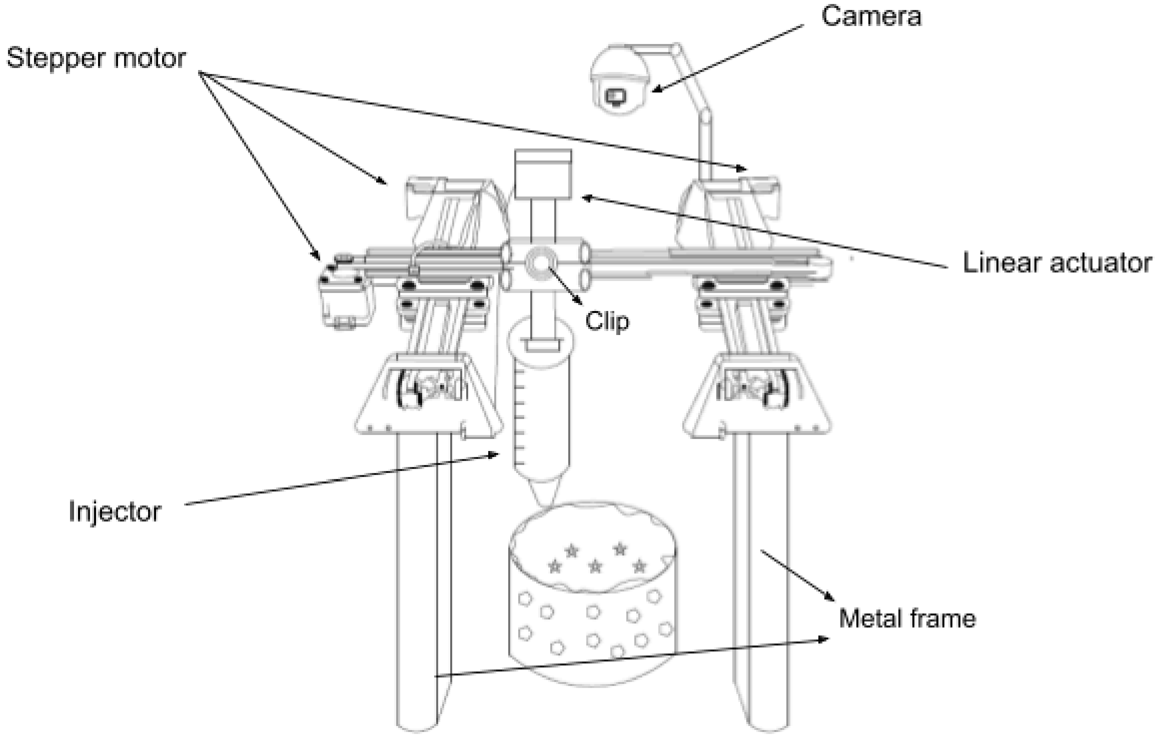

Fig. 3 Mechanical Structure of the Machine

Before decorating, we need to manually move the injector to the corner of the machine to avoid blocking the camera. According to the detection from the computing subsystem, the motors can adjust their speed on the x and y axis to allow the injector to move along the right track.

Based on the observation on human decoration on cakes, the speed of moving the syringe is around 0.1 - 1 m/s. If too fast, cream would be thrown away and would not stick to the cream; if too slow, the cream would stack up on the cake causing ugly patterns. To deal with this problem, we gave the input of the linear actuator a constant 5 V voltage to make the speed of the linear actuator unchanged. It takes about 60s to push all the cream out of the syringe.

We also met a problem with our machine during the process of the project. When we first got the machine from the machine shop, we tried to move the rails by hand to see how it worked. It turned out that the belt between the pulleys is very tight, resulting in large friction between the axis and making it very difficult to move. After setting up the Arduino program, we let the machine move forth and back. From the observation, we found that it is very difficult for the stepper motors to overcome the friction and the vibration of each metal frame made some noisy sound that is definitely not appropriate for this kind of machine. We also marked the starting and ending position of each move. Ideally, every movement should have the same positions, but due to the friction the result was not that accurate.

After finding the problem, we communicate with the machine shop for improvement. According to our need, the machine shop swapped pulleys with larger ones (increased the torque ratio), reducing the torque needed on the motor. The mounting of the belt is modified so that less force is pulled to the side, generating less friction. Also some lubricate is added on the rail to reduce friction. After that, the problem was solved.

2.4 Circuit Subsystem

We developed two versions of the circuit subsystem. The earlier version centers around the ATmega328P MCU, which is complemented by peripherals on our custom-designed MCU board: a step-down converter powers the system, a USB to UART bridge chip facilitates communication with a laptop, and A4988 drivers link it to the motor. The pathway for controlling the motor from the laptop operates through this configuration.

Additionally, the subsystem incorporates an external 12V DC power supply that energizes all the electrical components and the motor in the subsequent subsystem. This power supply connects to the PCB via a barrel jack. At the same time, we use the regulator to convert 12V to 5V. In this way, we can provide 5V to activate the logic circuit.

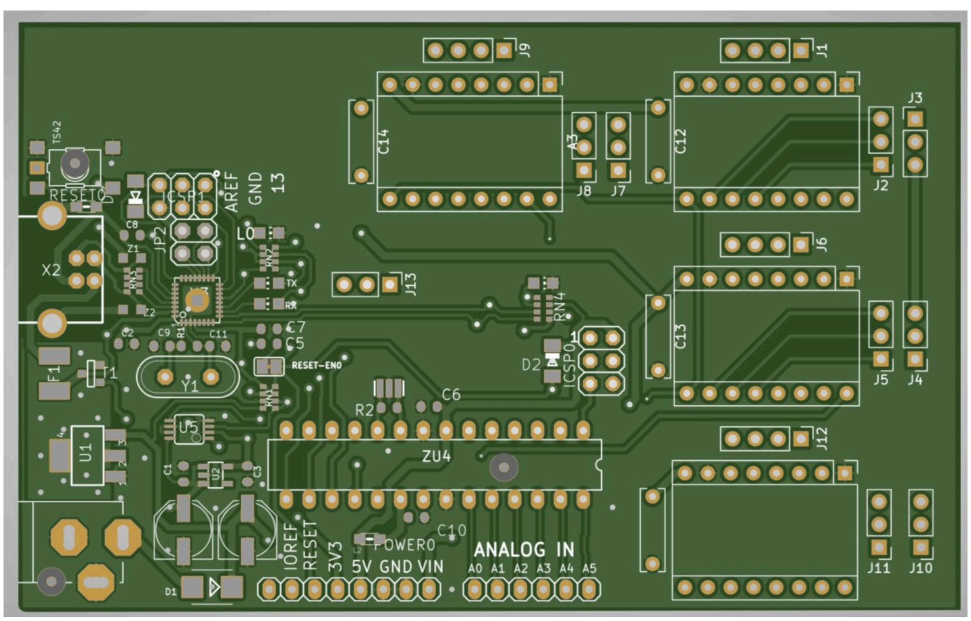

Fig. 4 old version PCB.

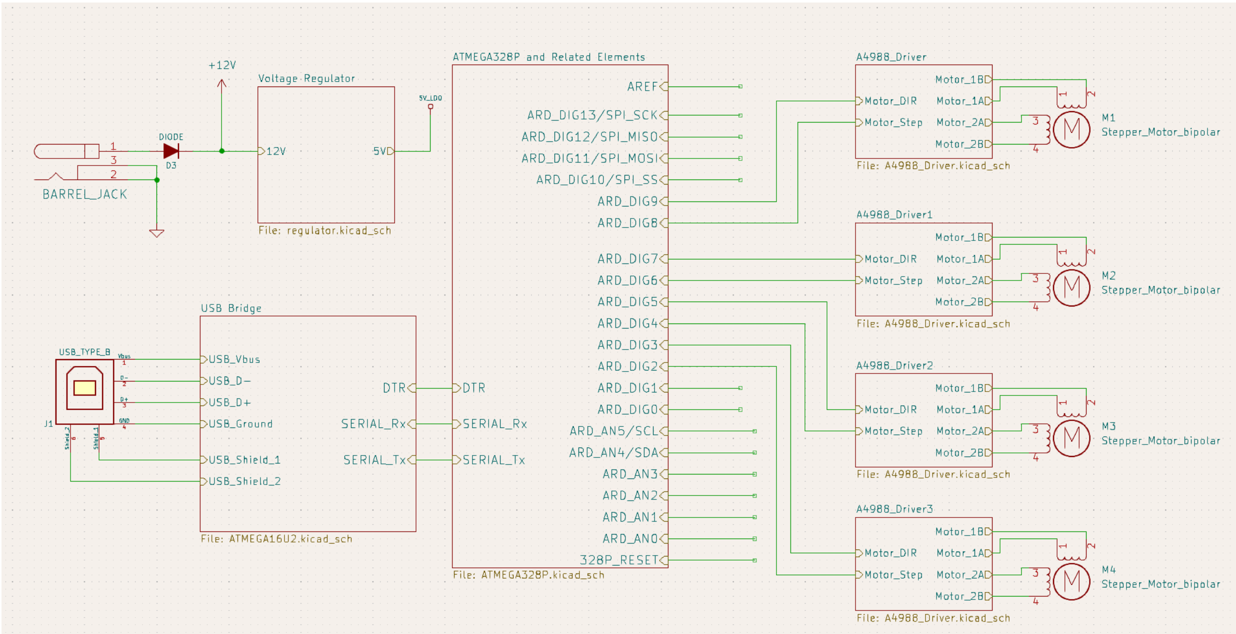

Fig. 5 High level schematic of the early version circuit subsystem.

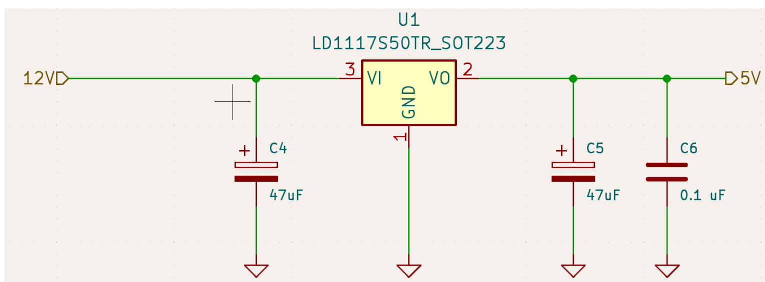

Fig. 6 Schematic for the regulator, receiving 12V from the power supply and providing an output of 5V.

However, for the early version, we found that we use many redundant components and use the unsuitable footprint for some components. Moreover, we need to use one more MCU to establish the USB bridge. All these issues lead to the result that it is hard for us to solder the components to the PCB. Thus, to solve the issues, we design the second version of PCB.

For the updated version, we replace ATMega328P with ATMega32U4 since ATMega32U4 has the USB bridge and is compatible with Arduino Integrated Development Environment. Moreover, we delete the design of the voltage regulator. Since we only need 12V to drive the motor and 5V to activate the logic circuit, 12V is provided externally, and the 5V is provided by USB. For the motor drivers, we keep the design from the old version. We still use A4988 to drive stepper motors and L298N to drive the linear actuator.

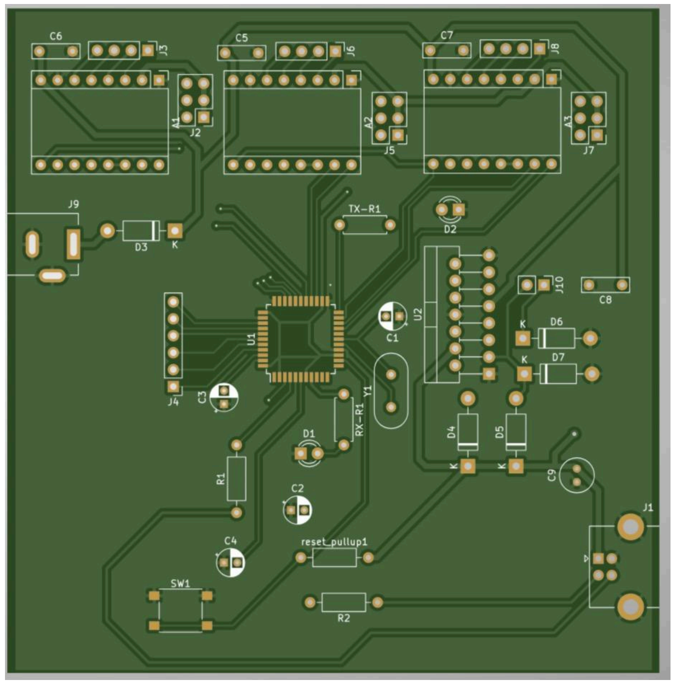

Fig. 7 updated version PCB.

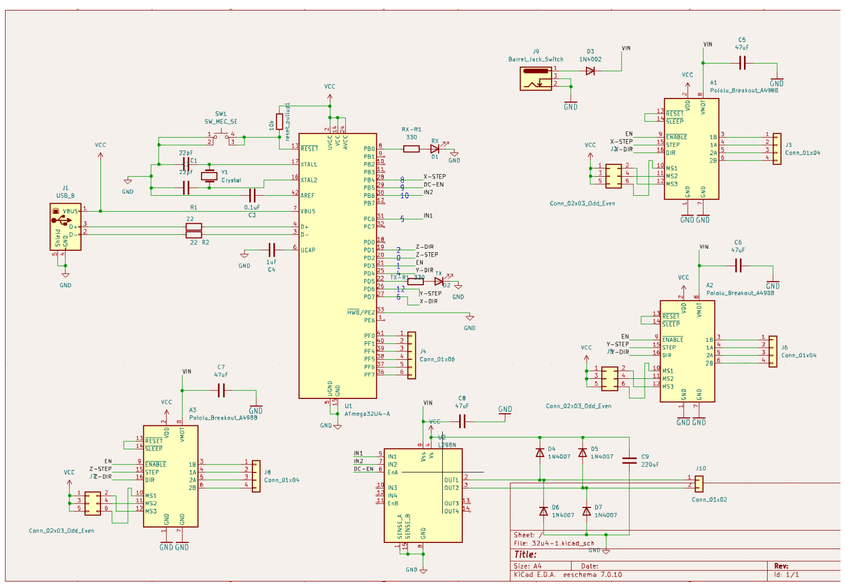

Fig. 8 High level schematic of the updated version circuit subsystem.

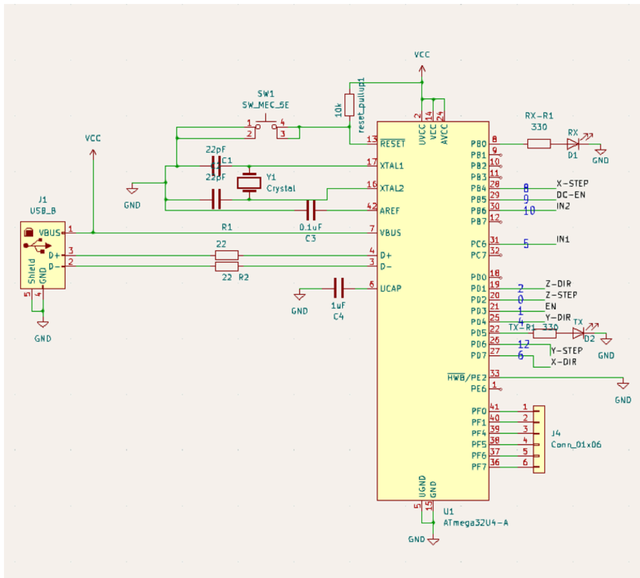

Fig. 9 schematic of ATMega32U4.

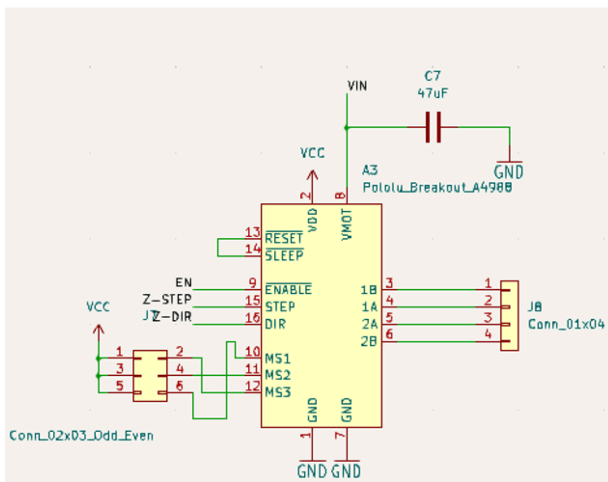

Fig. 10 schematic of Stepper Motor Driver A4988.

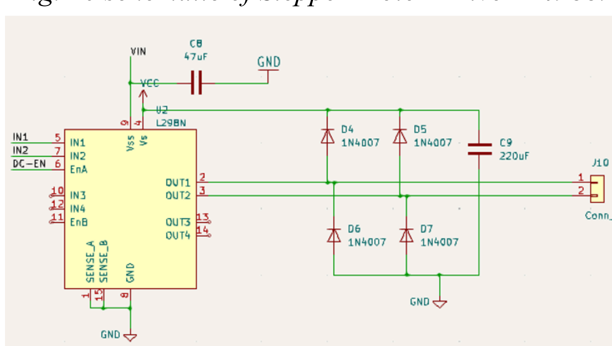

Fig. 11 schematic of DC Motor Driver L298N.

3. Design Verification

3.1 Computing subsystem

Requirement: Optical resolution of the camera is better than 0.5cm

This is dependent on the product we bought and the place we mount it. To verify that, we take a picture of the area it covers:

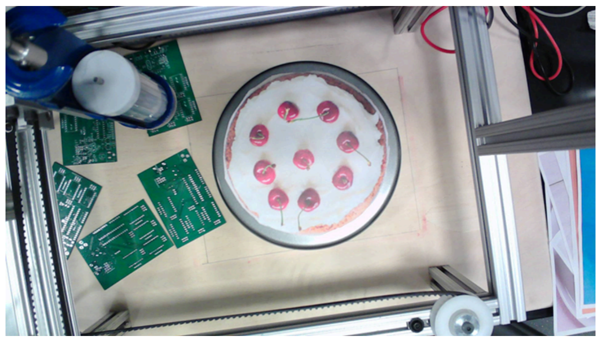

Fig.12 Sample picture taken by our camera

Which is about 1200mm x 700mm in size.
The digital resolution is 1920 x 1080.
From this, we calculate that the digital resolution is:
1200/1920 = 0.625 mm/pixel.
The optical resolution could be lower, but is still way above the requirement of 0.5cm/pixel.

Requirement: Correctly recognize cake position and outline 90% of the time

To verify this, we searched for 10 images of the top view of foods, printed them out, placed them under the camera and found the success rate of a recognition.

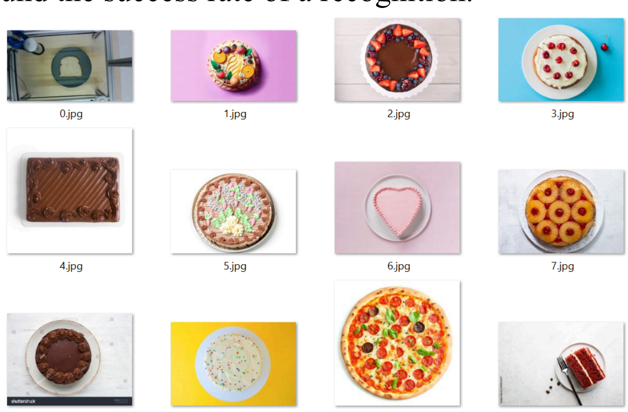

Fig.13 test images for recognition

Out of the 12 images, only one encountered a constant inaccurate recognition on the edge, so the accuracy is:
11/12 = 91.7%
The requirements are met

Some other essential components in this subsystem are program codes. So their verification process is similar to debugging in software development. We tested each standalone function using edge cases as input, and fixed the code until all the bugs were gone.

3.2 Mechanical subsystem

According to our design document, to achieve the high level requirement of within 0.5cm range of decoration accuracy, we need the step loss rate (that is the accuracy of movement) to be less than 0.21% for the motors on each axes. This error could be measured by letting the axes run by a distance and comparing to the actual distance run. The test process follow:

1. Mark the starting and ending positions of the move, and measure it by a ruler.

2. Calculate the distance it is supposed to move can be calculated from the number of steps being sent, multiplied by the parameter of the pulley.

3. Repeat several trials with various movement lengths are performed, and for both axes.

The results and error calculations are shown in the table below:

axis	steps driven	expected travel length (mm)	actual travel length (mm)	error (%)	average error
x	100	31.416	31.5	0.26738	0.212206095
	150	47.124	47	0.263136	
	200	62.832	63	0.26738	
	250	78.54	78.5	0.050929	
y	100	47.124	47	0.263136	0.237670826
	150	70.686	70.5	0.263136	
	200	94.248	94	0.263136	
	250	117.81	118	0.161277	

As shown in the table, the error of the movement is mostly around the maximum step loss rate. Some samples may have a little bit above it, but this is mostly due to the inaccuracies of our measurement by a ruler. So it meets our expectations for achieving the high-level requirements.

In the future, we will be starting to test for the machine to draw specific shapes, like circles and squares. We would replace the syringe with a pen so that we could keep its track, and compare and see how much distance it derives from the initial trajectory after multiple runs.

3.3 Circuit subsystem

Requirement: Power supply can maintain a maximum total output that is enough to motivate all the electrical components (especially motors).

To verify this requirement, we operated all the motors and the linear actuator at maximum speed for two minutes, paused for one minute, and then resumed the test for another two minutes straight. Following this procedure, we observed that all the motors functioned steadily.

Requirement: The control signal to the motor is accurate enough that it deviates from the desired part within 1 cm.

To verify this requirement, we compared 5 shapes drawn by the machine with the shapes predefined by the program. We pick 10 points from each shape and record the average deviation of all the points for each shape.

SHAPE	Average Deviation(cm)
Circles	0.29
Smiling face	0.34
Star	0.42
Spikes	0.38
“HAPPY” text	0.72

Fig.14 shape drawn by the machine(left one) shape predefined by program(right one)

4. Costs

4.1 Parts

Quantity	Part	Part Number	Cost
1	Camera	EMEET 1080P Webcam	$31.78
1	Linear actuator	NORJIN Mini Electric Linear Actuator 12V	$27.99
3	stepper motor	SIMAX3D Nema17 Stepper Motor	$29.99
1	Cupcake Injector	Norpro Cupcake Injector/Decorating Icing Set	$5.98
4	Motor Driver	A4988 Stepper Motor Driver	$10.19
1	Development Board for test	Arduino Uno R3	$27.60
1	Circuit Board for test	Arduino CNC Shield Board	$3.99

4.2 Labor

Name	**Rate**	**Hours**	**Total(R*H*2.5)**
James Zhu	$35	240h	$21000
Muye Yuan	$35	240h	$21000
Rui Gong	$35	240h	$21000
			$63000

Grand Total: $63000

5. Conclusion

5.1 Accomplishments

The topics involved in this project are circuit design, embedded system, robotics, and computer vision. Upon completion, we improved a lot on these technical skills. Throughout the semester-long endeavor, effective collaboration and time management skills were also essential. As a result, we successfully delivered a functionally complete project, meeting all high-level requirements, and impressed the users by printing fancy shapes with a variety of material and foods.

5.3 Ethics

The best way to set-up, test and showcase the functionality of our machine to simulate real-life circumstances is to use real ingredients: cream and cake. So some food will inevitably be wasted. We reduce the amount as much as possible this includes:

- Except for the final demonstration, we replaced real cakes with fake ones made from cupboards. We also tested with toasts, which are still edible after the decoration, so we consumed all of them as snacks.
- In the experiment, since we are not eating them, we would try to recycle the cream by getting them off from the fake cake and put back into the syringe.

5.4 Future work

For future work, we want to add an additional ToF distance sensor beside the camera to get the height of the cake so that the injector can adjust its height individually. We will also improve the project by adding a rotational cake holder. In this case, the tip of the injector can move to the side of the cake and put some cream on the side.

One of the problems we meet for now is that there is some delay between pushing the syringe and cream coming out of the injector tip. In future improvements, instead of using a simple cream injector, we will use a sauce dispenser pump which contains constant pressure and we can precisely control the timing of the cake.

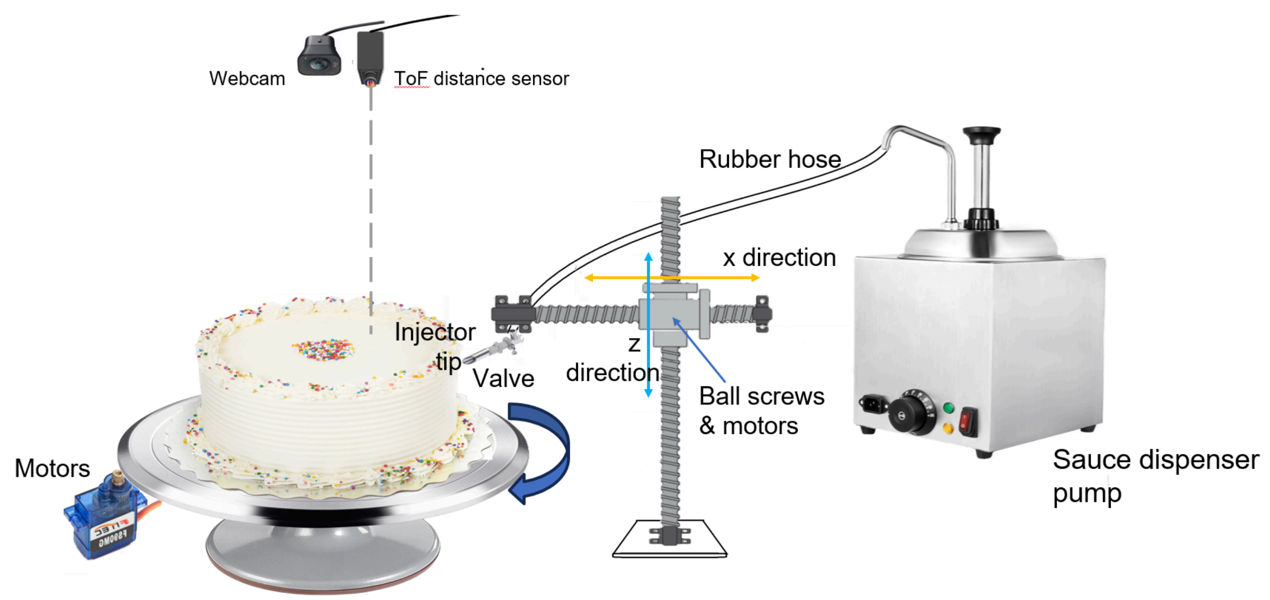

Fig.15 Future work for the project

References

[1]"FoodSAM: Any Food Segmentation", *Lan, Xing, Lyu, Jiayi, Jiang, Hanyu and Dong, Kun and Niu, Zehai and Zhang, Yi and Xue, Jian,* IEEE Transactions on Multimedia, Aug. 2023. http://dx.doi.org/10.1109/TMM.2023.3330047

[2] "INKSCAPE", https://inkscape.org/

[3] "MI-GRBL-Z-AXIS-Servo-Controller", *ikae*, n.d. https://github.com/ikae/MI-GRBL-Z-AXIS-Servo-Controller

Appendix A Requirement and Verification Table

Circuit Subsystem RV Table

Requirements	Verification	Results
Power supply can maintain a maximum total output that is enough to motivate all the electrical components (especially motors).	Spin the all motors and the linear actuator to max speed for 2 minutes, rest for 1 minute , and perform the test for another 2 minutes staright, see if it runs stably.	Success
The control signal to the motor is accurate enough that it deviates from the desired part within 1 cm.	Run the test code to navigate the syringe along a circle and a square repeatedly at full speed; mark the designed trail, see if deviate from it. Can take a video and replay to check more closely.	Success

Mechanical Subsystem RV Table

Requirements	Verification	Results
Mechanical parts strong enough to support the system under full load.	Fill the syringe with cake. Operate the motor to move it around. Check if it causes any physical deformation that hinders the movement.	Success
Motors powerful enough to drive the movement and injection of syringe under full load.	Perform a cake decoration with minimum cream in the syringe, then perform the same decoration with the same route with a fully loaded syringe. See if the speed or quality of finishing the job changes drastically.	Success
Speed of the motor and extrusion of cream are consistent, not causing cream to pile up or not forming a continuous line.	Test program producing shapes of a rectangle and circles, check if the result is acceptable (no issues described).	Success

Computing Subsystem RV Table

Requirements	Verification	Results
Optical resolution of the camera is better than 0.5cm in physical coordinate, when hanging at 0.6m looking down	Put a ruler with marks, make sure two marks 0.5cm apart are clearly resolved in two pixels in the image taken by the camera	Success
DNN correctly recognize cake position and outline 90% of the time	Print 20 pictures of the top surface of cakes on papers with various shapes and looks, scan them under the camera. Put some common distraction, like a cake pans beneath them, or a plate on the side. See if recognized correctly for at least 18 of them.	Success
Mapping from camera pixel coordinate to physical coordinates results in a maximum deviation of 0.5cm	Mark several points on the base of the machine. Measure its physical coordinate with a ruler. In the camera view, locate their pixel coordinates, and use the algorithm to map them back into physical coordinates. Check if the measured results and calculated results are within the euclidean distance of 0.5cm.	Success
Algorithms projecting a trajectory onto movements on the x and y axis are correct.	Add visual to the debugging: plot the calculated movement trajectory on an openCV window; test the part by moving along a rectangle and a circle shape; check in the window if the desired path is followed.	Success