

Automatic cake decorator
ECE 445 Senior Design

3/20/2024

Team 47

Muye Yuan, Rui Gong, James Zhu

TA:Jason Zhang

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 Visual Aid

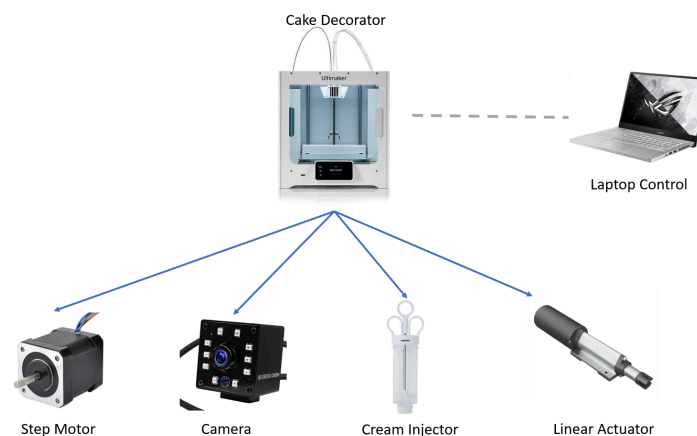


Fig. 1 Visual Aid

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

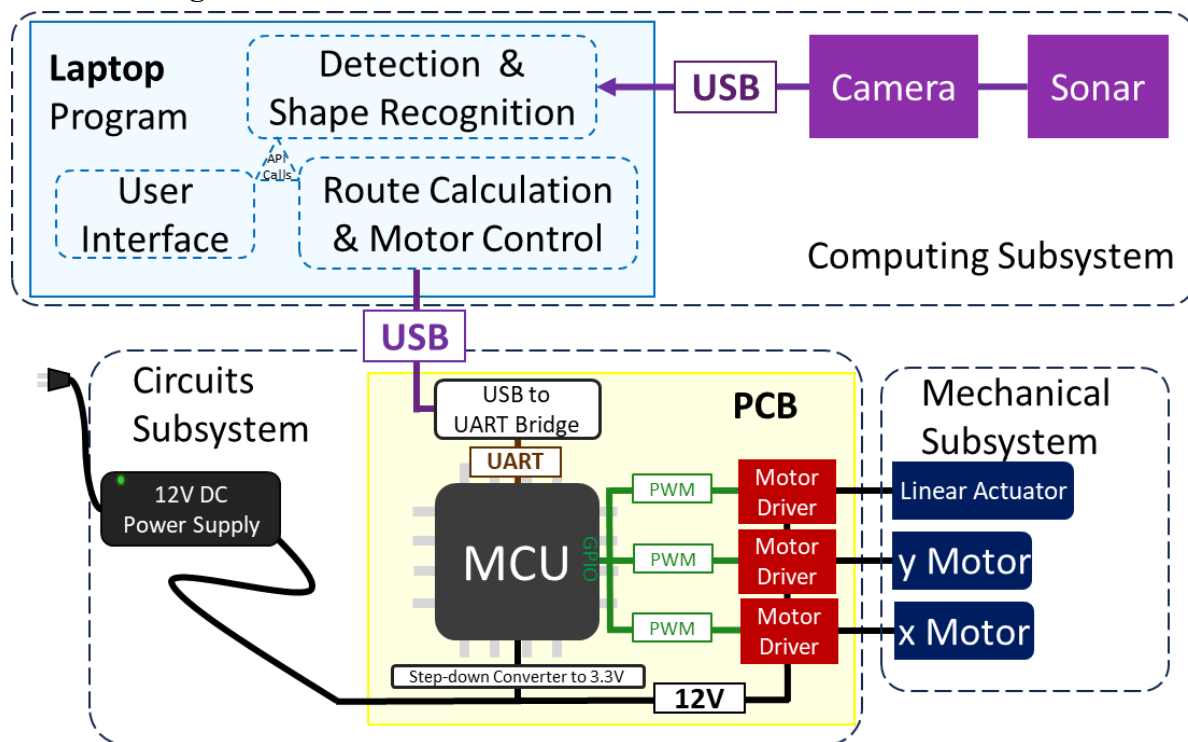


Fig. 2 Block Diagram

2.2 Physical Design

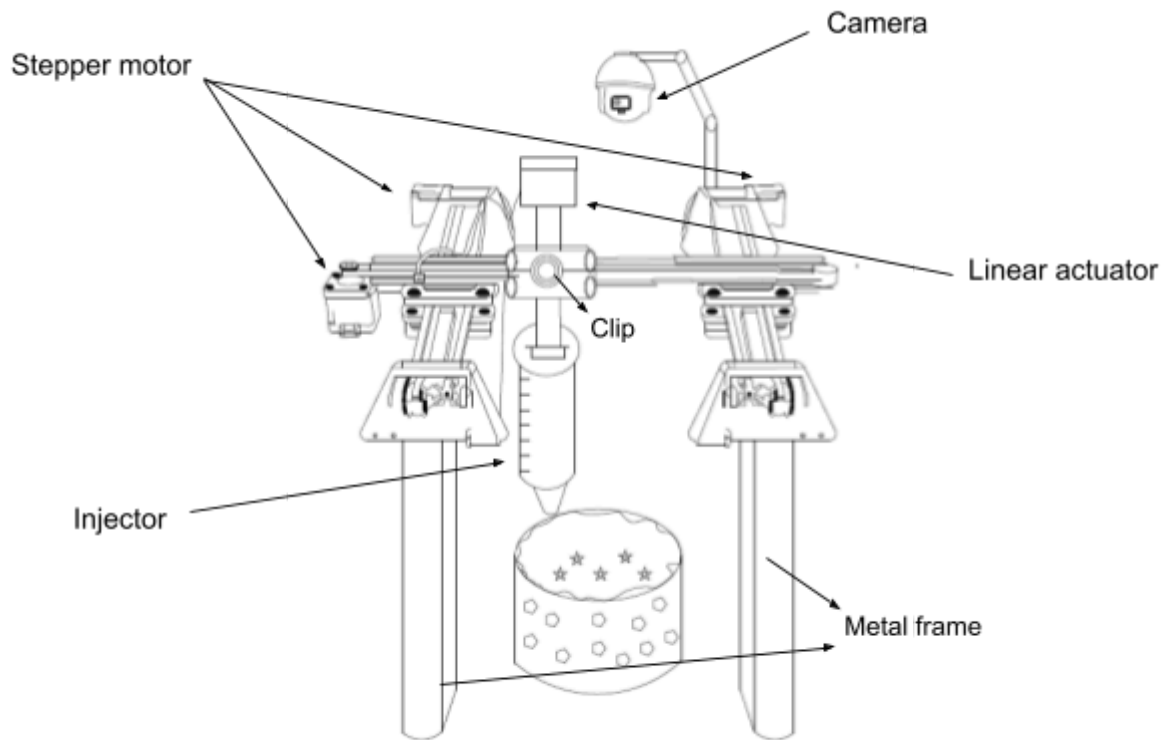


Fig. 3 Physical Design

As shown in the picture above, the structure includes two parallel rail guides, elevated from ground by metal frames. Sliders on them enable the travel along y axis, and should provide travel length of around 11". The third rail guide is mounted perpendicular to the two y-axis guides across on the two sliders, forming a H-shape looking from above. This provides the movement in x axis, with travel length around 10". All the sliders are powered by stepper motors, and force delivered by pulley & belt system as labeled in the detailed diagram below. The linear actuator and injector system is grabbed onto the x-axis slider by a clip, thus their height could be adjusted manually by unmounting them from the clip. The system is grabbed vertically, meaning injector in the bottom, pushing by the linear actuator on the top. The maximum travel of the "pushing" movement is around 10cm. The tip of the injector from ground should provide a maximum ground clearance of around 8", giving sufficient for extra tall cakes. The camera is hanged from top by a metal stand, looking down, and has height around 25" from ground to capture the cake without any distortion, or potential risk of hitting by the relatively tall linear actuator and injector system.

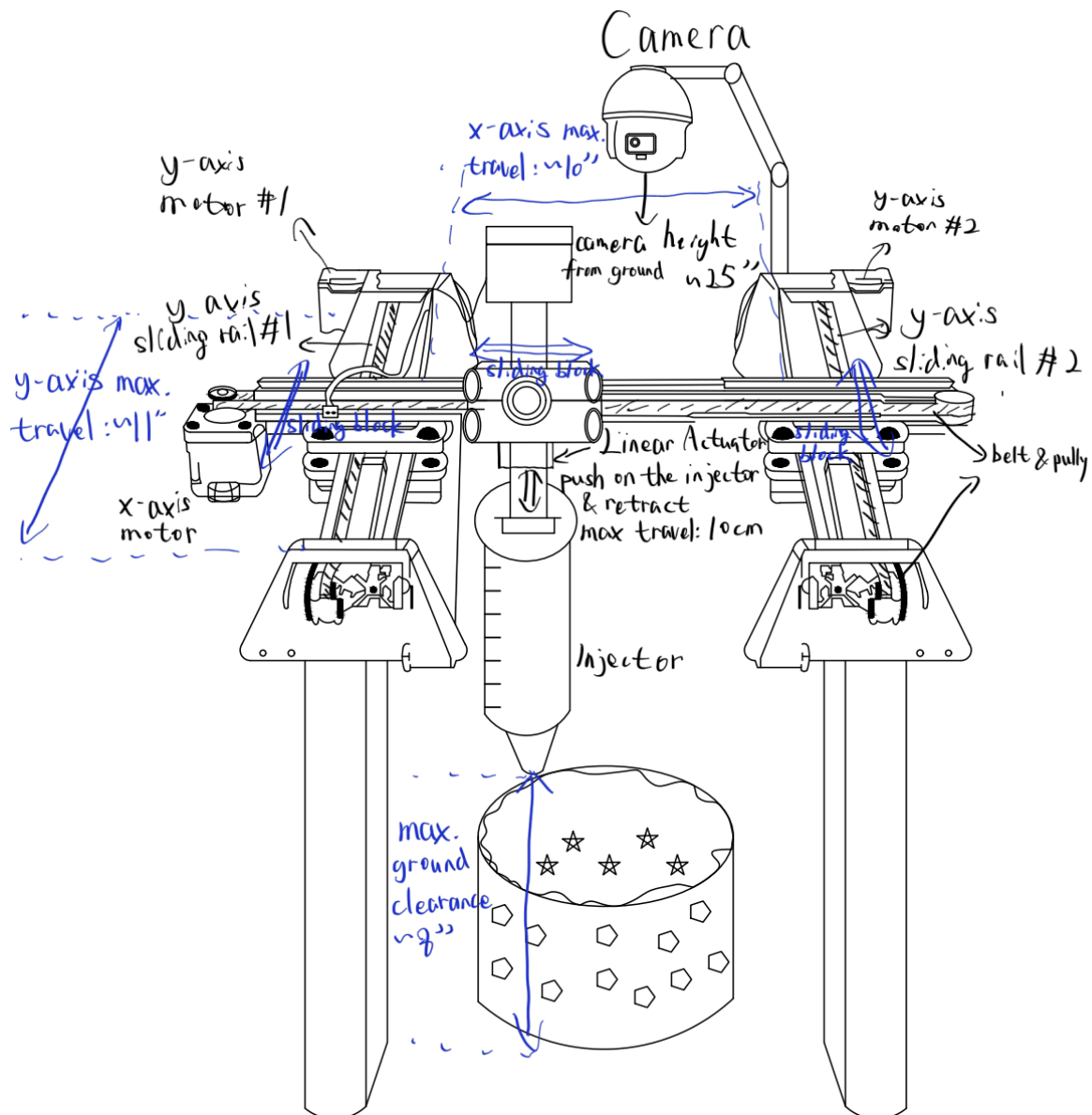


Fig. 4 Fully labeled physical diagram

2.3 Computing Subsystem

Parts:

1x Adjustable Metal Stand

1x 1080p USB Camera

1x Laptop

This subsystem includes a HD webcam held by a metal stand looking down to the top surface of the cake, connected to the laptop via USB. Our program is running in the laptop, and divided into several modules:

1. Computer vision part detect the position, shape and outline of the cake from the camera capture

2. The route calculation and motor control, as the name suggests, decide and give the signals for the motor to move. It's not directly connected to the motors, but to the MCU on the circuit's subsystem via USB. Once a key is pressed, it would send out the calculated target motors speeds calculated as described above to the MCU, allowing it to drive the mechanical parts.
3. The UI module handles the interaction between the modules above and users. It makes the cake detection result transparent to the user, and allows the user to choose the decoration style, or let it re-detect if the result is incorrect. This means the decoration may not necessarily be a simple line along the edge, but some zig-zag curves, with connected and unconnected parts, resulting in some fancy shapes and textures. Then it shows some operation buttons for the start and termination of the motor moving and decoration process.

All these modules will be written in C++, compiled and linked as a single program, so communication between them is standard C++ function call.

For the camera, for the system to meet the third high level requirements of 1cm of accuracy, the real-world resolution should be 0.5cm, allowing it to have an error of half a centimeter to the left and right. Say we the camera covers an area of $2m \times 1.125m$, the theoretical resolution should be around at least 400×225 pixels. Actual camera digital resolution on the datasheet should definitely be higher, as there are optical resolution limitations and noise.

For our program, it should recognize the edge of the cake with DNN based semantic segmentation algorithms. It could identify the position of the cake, and fit the edge into a set of waypoints for the cream extruder to follow. This is essential for the first high level requirement. The segmentation algorithm should be within the range of 10 pixels and success rate of more than 90%.

Although this subsystem is not connected to the actuators (motors in the mechanical subsystem) directly, the computation done in it is essential for an accurate decoration as the user previewed, and contribute to meet the third high-level requirement of accuracy of 1cm. There are two aspects:

1. Up Till now, the trajectory is calculated with respect to the pixel coordinates, and needs to be mapped into x-y coordinates on the decorating machine. The algorithm should be able to do unprojection from camera view to physical coordinates. It also needs automatic recalibration if the relative position of the camera and the machine is changed, by referring to two marks put at the boundaries of the machine. After the calibration, the mapping should not shift for more than 0.5 cm.
2. The program calculates the velocity movement of the motors necessary to push out the cream uniformly and move seamlessly along the designed trajectory. The trajectory must be projected onto the x and y axis, representing the positions of motors controlling the two axis. As the speed to move along the trajectory may not be constant, a dummy variable of s would be introduced. The position (expressed as P) is a 2×1 vector, with the two components x and y positions:

$$P = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix}$$

and s is a function of time t $s(t)$. Then the derivative of P wrt t :

$$\frac{dP}{dt} = \frac{dP}{ds} \cdot \frac{ds}{dt} = \begin{bmatrix} \frac{dx}{ds} \\ \frac{dy}{ds} \end{bmatrix} \cdot \frac{ds}{dt}$$

This is the time-varying speed the two motors should target when decorating the cake. Therefore, the sample rate of the route should be at least 60Hz, allowing us to have an accurate calculation of the derivative, that makes sure the speed does not have an error greater than 1 mm/s.

Requirements	Verification
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
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.
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.
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.

2.4 Circuit Subsystem

Parts:

- 1x ATmega328P MCU
- 3x A4988 stepper motor driver board
- 1x self designed PCB with other electrical components
- 1x 12V DC power supply

The core of this part is the ATmega328P MCU, along with its peripheral soldered on our self-designed MCU: step-down converter for powering it, USB to UART bridge chip handles its communication with laptop, A4988 drivers connecting it with the motor. The control from the laptop to the motor gets through this datapath.

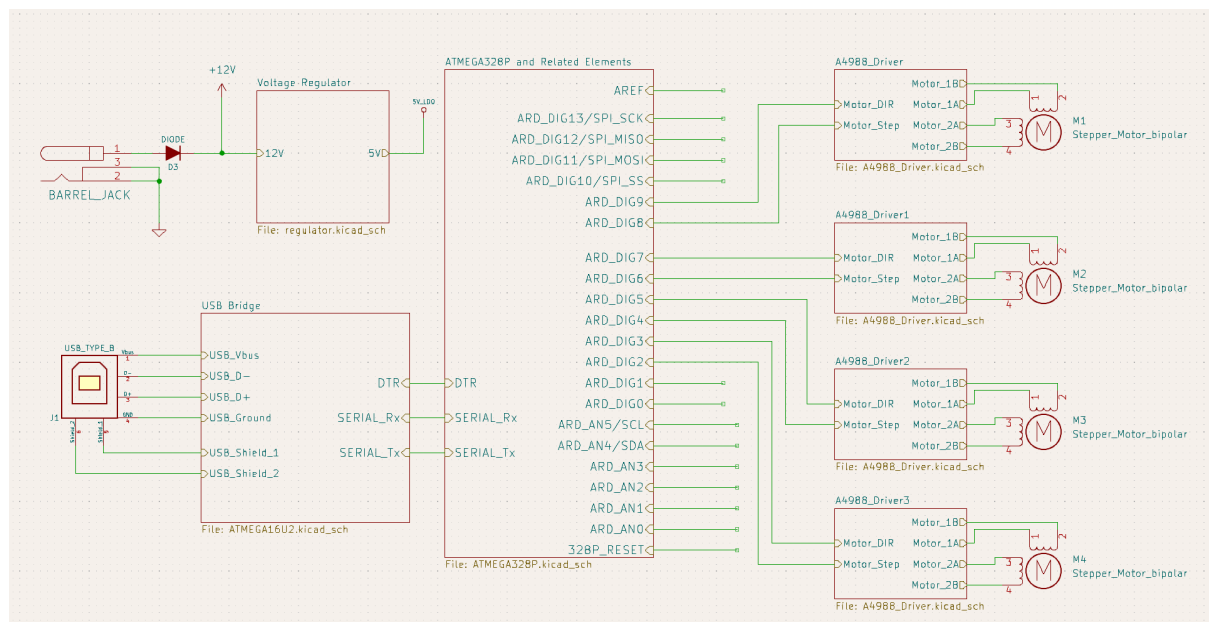


Fig. 5 High level schematic of the circuit subsystem.

The subsystem also includes an external 12V DC power supply powering all these electrical components, plugged into the PCB by the barrel jack, as well as the motor in the next subsystem.

The ATmega328P operates at a voltage between 2.7V to 5.5V [1]. Its standard bundled step-down converter could easily handle this range, dropping the voltage from 9-15V to 3.3 ± 0.1 V:

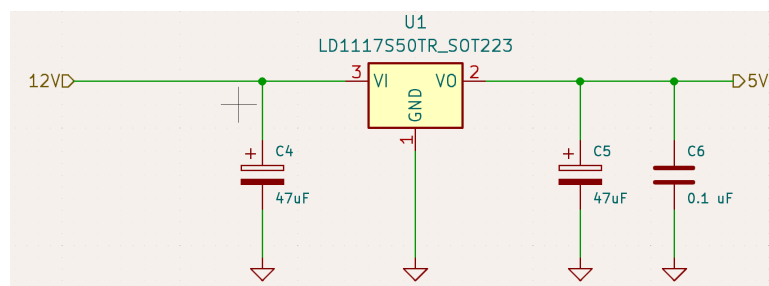


Fig. 6 Schematic for the regulator, receiving 12V from the power supply and providing an output of 5V.

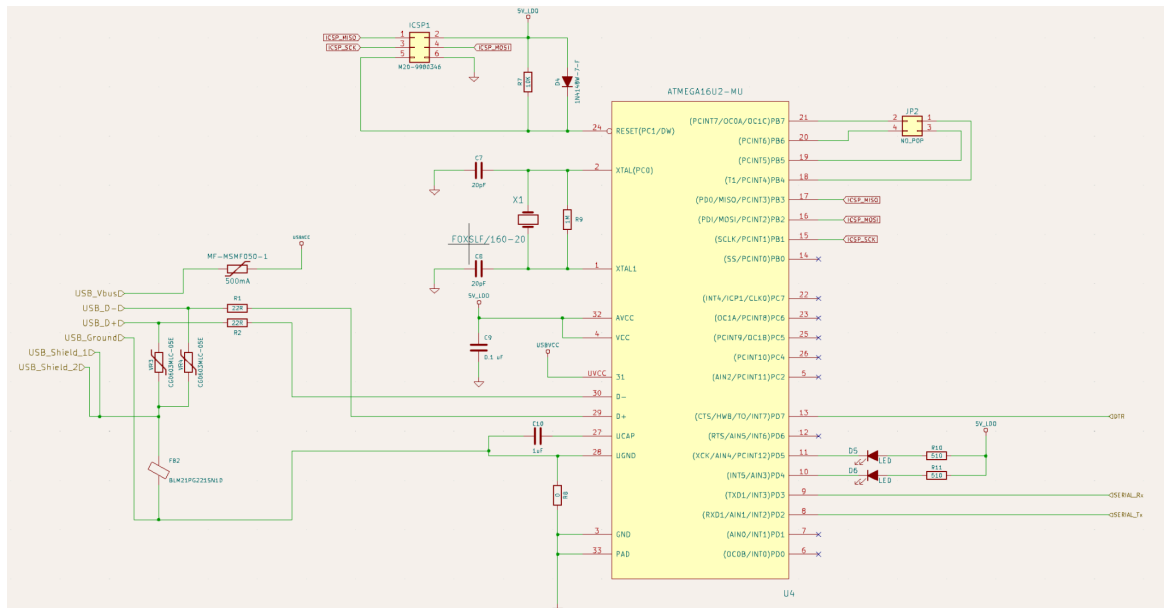


Fig. 7 Zoom-in schematic of the USB bridge, an ATMEGA16U2 MCU as USB to Serial converter. Unused pins are unconnected.

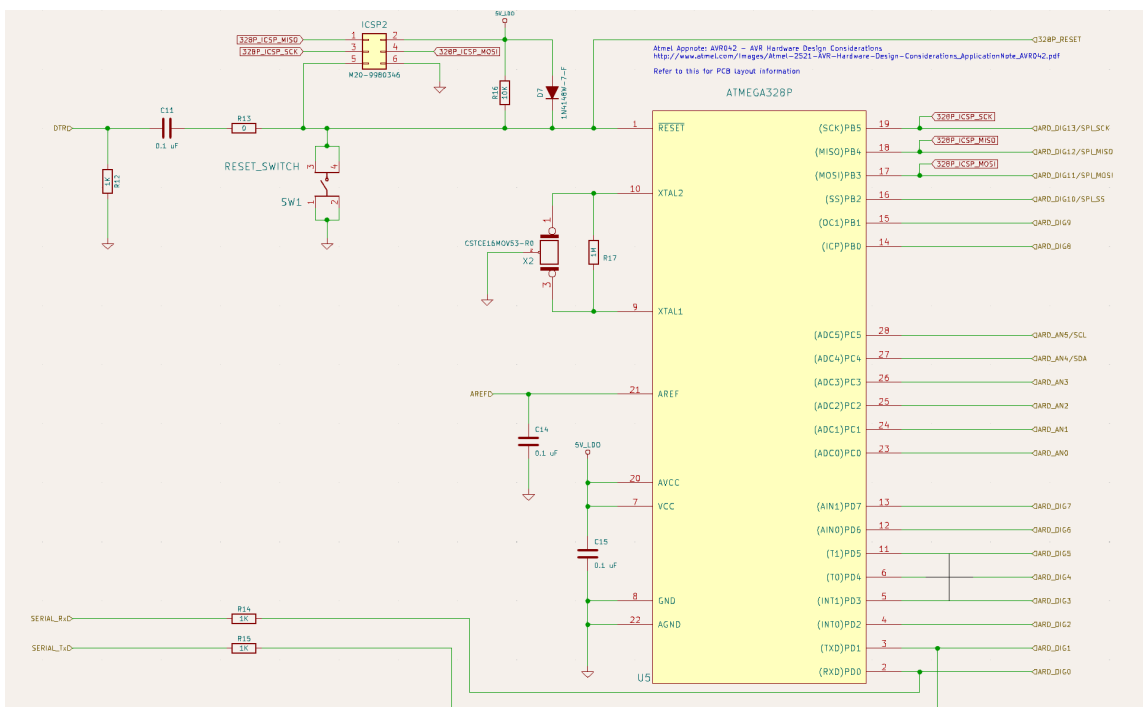


Fig. 8 Zoom-in schematic of the ATMEGA328P microcontroller. All the related components connected in standard configuration according to its datasheet [1].

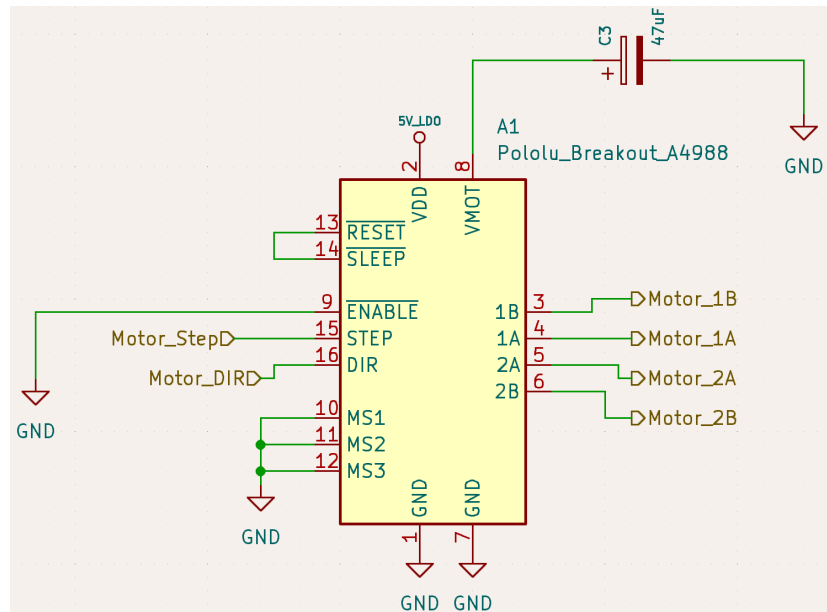


Fig. 9 Schematic of the A4988 motor driver.

The motors are rated at 1A maximum current, which is impossible to drive by the GPIO of the MCU which just has 18mA max current. That's why we need the A4988 motor driver to receive only control signals from the MCU and give power outputs to the motors.

The A4988 motor driver requires two power supplies:

- One for its logic circuit, requiring a voltage range of 3-5.5V. This can be powered by sharing with the MCU's step-down converter.
- Another for powering the motor where the current is transmitted through it. This requires a range of 8-35V, which can be drawn from our 12V DC power supply directly.

The total power of the electrical components used is listed as follows:

- The motors operates at max 12V and 1.2A, resulting in a maximum power of 15W, and there are three of them
- The elements on the PCB: MCU, motor drivers, and the peripherals including resistors and capacitors are all designed for low-power computing applications by the manufacturer, and would not exceed a total power of 15W.

Therefore, the external DC power supply at least $15W \times 3 + 15W = 60W$, and should supply a stable voltage within the range of 9-12V.

Requirements	Verification
Power supply can maintain a maximum total output of 60W to 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.
Voltage regulators sustain a stable 5V for the logical components (MCU, USB bridge, drivers etc.) under max load.	Keep sending and feedback the same message between computer and MCU in a loop; give control signals to the motor drivers meanwhile. Test if the feedback is normal, and probe on the output of the voltage regulator, observe if the voltage is above 4.6V on the oscilloscope.
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.

2.5 Mechanical Subsystem

Parts:

- 1x Large cake pan
- 3x Linear rail guide and slider
- 3x 42 stepper motor 1.5A 1.8 Degrees
- 1x linear actuator
- 1x Metal frames and parts
- 1x Extra Large Dessert Decorating Syringe Set

The structure of the machine resembles that of 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.

Before decorating, the injector should move to the corner of the machine to avoid blocking the camera. Based on the detection of CV, the motor can adjust speed on the x and y axis to make it move along the right track. If the linear motor reaches the bottom of the syringe, it will stop pushing.

According to 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. So our motor should be able to drive the system within this speed range.

Mechanical systems should be reliable enough. It needs to finish the decoration of the cake in more than 95% of the trials, without errors such as syringe getting stuck in by the cream, or motors and mechanical slides stuck somewhere in the middle.

Requirements	Verification
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.
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.
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).

2.5 Risk Analysis

In our design, we plan to incorporate a camera positioned atop the machine to accurately detect the portion and edges of the cake. However, due to inherent properties of the camera, distortion may occur at its periphery, leading to potential inaccuracies in detection. It is very critical to find the right balance – placing the camera too far from the cake may compromise accuracy, while positioning it too close may amplify distortion effects. To address this challenge, we aim to identify an optimal camera placement that minimizes both sources of error, ensuring precise and reliable detection.

2.6 Risk Tolerance

Requirements on motor step accuracy:

In the high level requirements, we want the discrepancy to be below 0.5cm. As the machine is driven by a step motor without close loop control, the error would build up throughout the decoration process. Therefore, the biggest error would appear by the end of the decoration line. If the accumulated error at this point is below 0.5cm, then the requirement is met.

As the decoration goes along the edge, it is related to the circumference of the cake. And as there are fancy curves, we give a multiplier of 4 times to the length of the decoration line. It could be calculated by:

$$\text{Diameter}_{\text{cake}} = 8 \text{ inch} = 20.32 \text{ cm}$$

$$\text{Length}_{\text{decoration}} = \text{Diameter}_{\text{cake}} * \pi * 4 = 255.35 \text{ cm}$$

As the line is on a 2D plane, and the two axes are moved by the motor separately, we assume the movement is projected evenly on the x and y axis.

$$\text{Movement}_{\text{motor}} = \text{Length}_{\text{decoration}} / \sqrt{2} = 180.56 \text{ cm}$$

To calculate how many steps the motor needs to take, we should find out how much movement one step of the motor would make. This is related to the circumference of pulley:

$$\text{Circumference}_{\text{pulley}} = \text{Diameter}_{\text{pulley}} * \pi = 9.42 \text{ cm}$$

And one step would turn the pulley by 1.8 degree, so one step will make it move by:

$$\text{step degree} = 1.8^\circ / \text{step}$$

$$\text{Circumference}_{\text{pulley}} * \text{step degree} / 360^\circ = 0.047 \text{ cm/step}$$

And moving the motor by $\text{Movement}_{\text{motor}}$, it needs:

$$180.56 / 0.047 = 3842 \text{ steps}$$

And having the error of 0.5cm on the 2d plane, we should make sure on any of the axes, the error should be less than:

$$\frac{0.5}{\sqrt{2}} = 0.354 \text{ cm}$$

And this would need:

$$0.354 / 0.047 = 8 \text{ steps}$$

So less than 8 steps could be loss after running 3842 steps, then step loss rate should below:
 $8/3842 = 0.21\%$

Tolerance of the voltage regulator:

In our project, we will use a voltage regulator that can convert 12v to 5v to guarantee voltage supply is constant and voltage supply can satisfy each different component. For the regulator, we will plan to use the LD1117S50TR 3-Terminal Adjustable Regulator. To mitigate risks, we prioritize operational stability, demanding the regulator to maintain a steady 5V output. It's imperative that the regulator maintains precise voltage accuracy to adhere to component specifications.

$$V_{in} - V_{out} > V_{dropout}$$

$$12 - 5 > V_{dropout}$$

We use 12V for the input voltage to account for maximum expected ripple. This tells us to choose a dropout voltage less than 7 V.

If too much power is dissipated inside the regulator, it will overheat. In this case, we must determine the power dissipated in the device, P_D .

$$P_D = i_{out} * V_{ce}$$

$$P_D = i_{out} * (V_{in} - V_{out})$$

According to the parameters given by the device and the V_{in} and V_{out} we want to achieve, we can calculate the junction temperature, T_j . If the junction temperature exceeds the maximum T_j , there will be thermal problems in the circuit.

$$T_j = i_{out} * (V_{in} - V_{out}) * (\theta_{jc} + \theta_{ca}) + T_a$$

Current

Parts	Current	Comment
Arduino uno r3	25mA	The Arduino Uno itself draws about 25 mA
4 * A4988	4 * 8mA	According to the datasheet for the A4988 the logic supply should draw a maximum of 8mA

LD1117S50TR

Variable	Value	Comment
i_{out}	57mA	According to the calculation above
V_{in}	12V	Voltage from power supply
V_{out}	5V	The target voltage after regulator
θ_{ja}	110°C/W	Thermal resistance junction-ambient
T_a	25 °C	Ambient Temperature

$$T_j = 0.057 * (12 - 5) * 110 + 25 = 68.89^\circ\text{C}$$

According to the datasheet of LD1117S50TR, the maximum operating junction temperature is 125°C. The calculated temperature is significantly lower than the maximum operating junction temperature. Based on the result, the device will not overheat.

3. Cost

3.1 Cost Analysis

3.1.1 Labor Cost

We are assuming a reasonable salary for students who graduate from ECE at Illinois is about \$25 per hour. For the next 10 weeks, we assume we will work for 10 hours per person each week. Therefore, the cost for each person is $25 * 2.5 * 100 = \$6250$, and for the entire group is \$18750.

3.1.2 Parts Cost

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. Schedule

- Week 6 (2/19):
 - Design document: Everyone
 - Communicate with the machine shop: Everyone
 - Start design of PCB: Everyone
 - Try Programming the motor control algorithm on Arduino: Everyone
- Week 7 (2/26):
 - Attend design review: Everyone
 - Design PCB: Muye Yuan, Rui Gong
 - Order electric parts: Muye Yuan
 - Continue programming the motor control algorithm on Arduino: James Zhu
- Week 8 (3/4):

- Order PCB: Everyone
 - Receive and examine mechanical parts from machine shop: Everyone
 - Solder PCB: Rui Gong
 - Test PCB: Muye Yuan
- Week 9 (3/11)
 - Spring Break: Everyone
 - Catch up if our progress is lagging the schedule: Everyone
- Week 10 (3/18):
 - Continue on program the motor control algorithm in microcontroller: James Zhu
 - Integrate and connect circuit with mechanical subsystem: Muye Yuan and Rui Gong
 - Develop UI for user program: Muye Yuan
- Week 11 (3/25):
 - Continue develop UI for user program: Muye Yuan
 - Develop CV and communication part: James Zhu and Rui Gong
- Week 12 (4/1):
 - Continue develop UI for user program
 - Continue develop CV and communication part
 - System start running as a whole
- Week 13 (4/8):
 - Preparation for Mock demo: Everyone
 - Testing on the general system
- Week 14(4/15):
 - Mock demo: Everyone
 - Final review for the project: Everyone
 - Final report: Everyone
- Week 15 (4/22):
 - Final demo: Everyone
 - Final report: Everyone
- Week 16 (4/29):
 - Final presentation: Everyone
 - Final report: Everyone

5. Safety and Ethics

5.1 Food sanitation safety concerns

According to IEEE Code of Ethics I.1 [2], The health of the users is essential. Our product is designed to deal with food, creating potential hazards for food safety, and we are well aware of that. In the machine, the syringe holding the cream and the food tray holding the cake are components that have direct contact with the food. They are parts bought by us instead of made by us, which are manufactured for food use, adhere to FDA regulations for Packaging & Food Contact Substances (FCS). [3] Other mechanical parts such as the sliding rails, and motors, though no physical contact with our food, can potentially put dust on the cake. So we make sure to clean them with food level wipes to make sure they are clean and won't pollute the cake.

5.2 Ethical considerations for food waste

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 should reduce the amount as much as possible this includes:

- Except for the final demonstration, we will be replacing real cake with fake ones made from cupboards. And we will try to give the cake to everyone after the demonstration.
- 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. And we might be able to find some out-of-date cake for the experiments.

6. References

[1] "8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash", *Atmel*, Jan.2015.S

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[2] "7.8 IEEE Code of Ethics", *ieee.org*, 2020.

<https://www.ieee.org/about/corporate/governance/p7-8.html>

[3] "Packaging & Food Contact Substances (FCS)", *fda.gov*, Oct. 2023.

<https://www.fda.gov/food/food-ingredients-packaging/packaging-food-contact-substances-fcs>