Automatic Cake Decorator

ECE 445 Team #47
Muye Yuan, James Zhu, Rui Gong

4/29/2024

Overview

1. Objective
2. Project design and overview of the functional block's requirements
3. Project build and functional test results
4. Challenge and solution
5. Conclusion & Future work

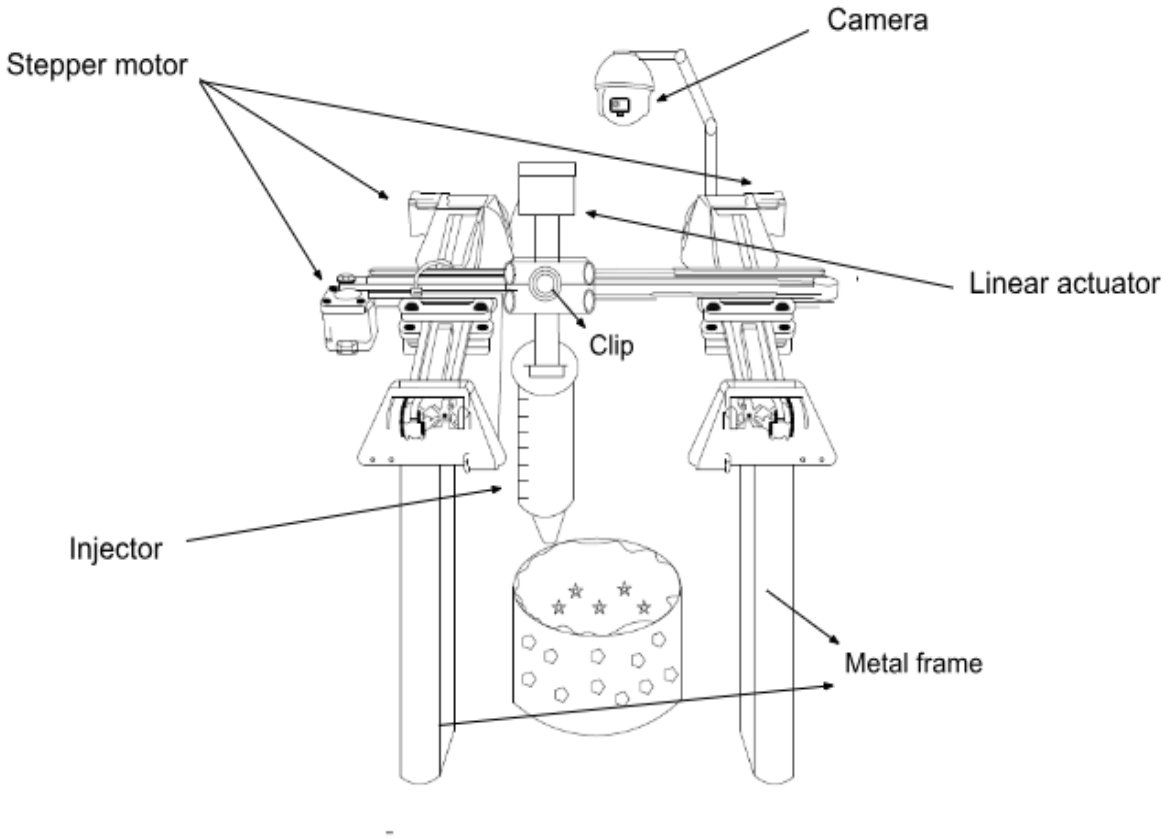
Objective

Objective

Problem we want to solve

Traditional cream decorating methods highly rely on manual application, which often results in variation in overall cake quality. Moreover, the labor cost during the process cannot be neglected.

Objective



Solution

- We designed an automatic cake decorator.
- Draw fancy shapes and curves on the top surface of the cake.
- Customize the shapes

Project design and overview of the functional block's requirements

Video of working machine

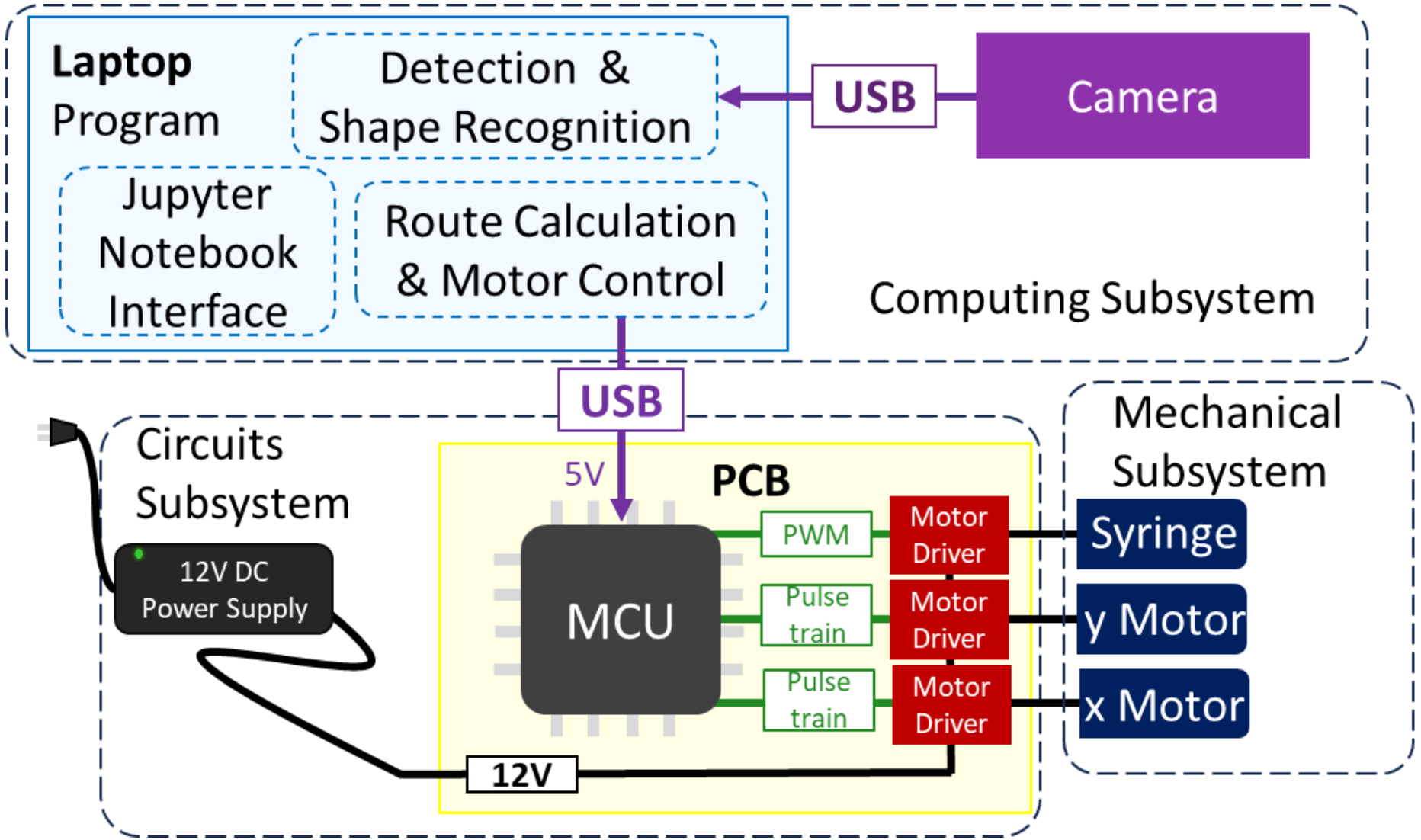
High Level Requirements

1. 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.

2. In the user program, at least **four decoration line** shapes or styles for the user to choose from.

3. 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.

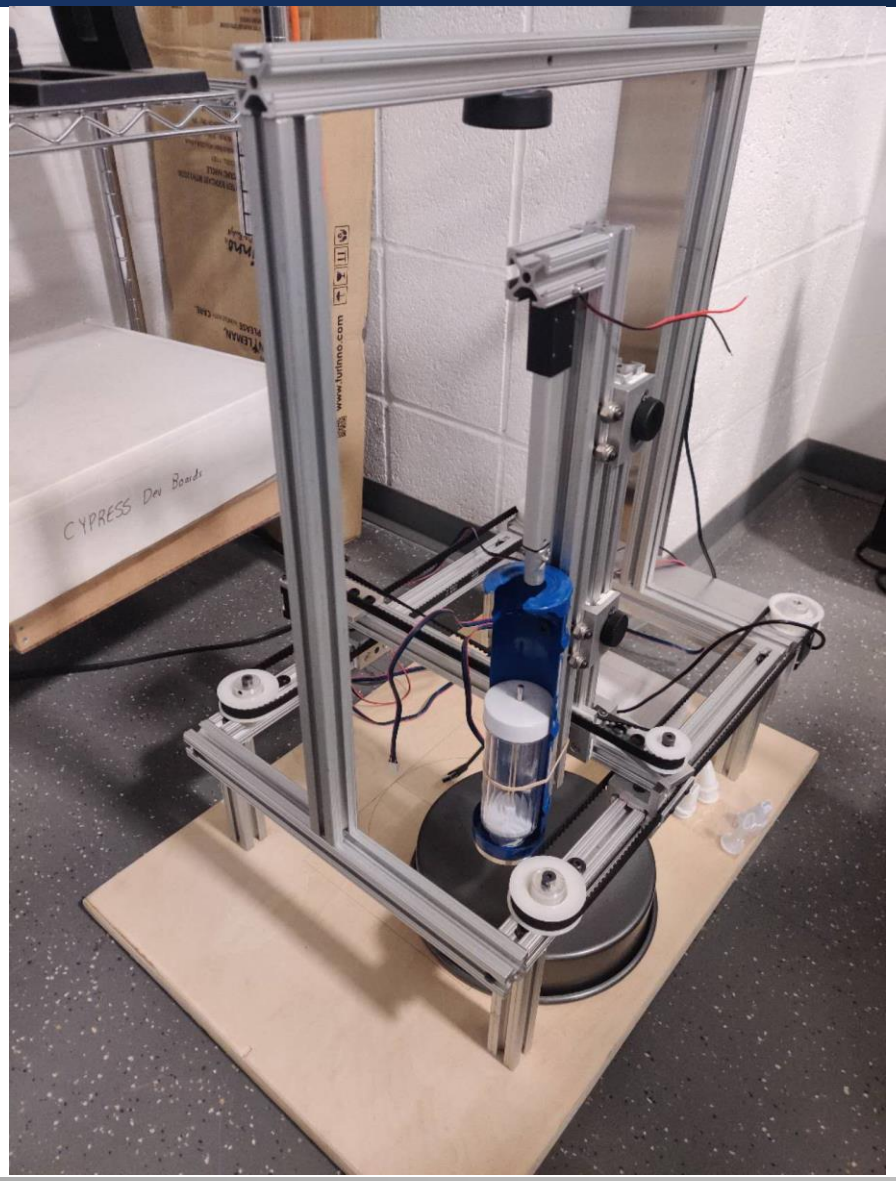
Block Diagram

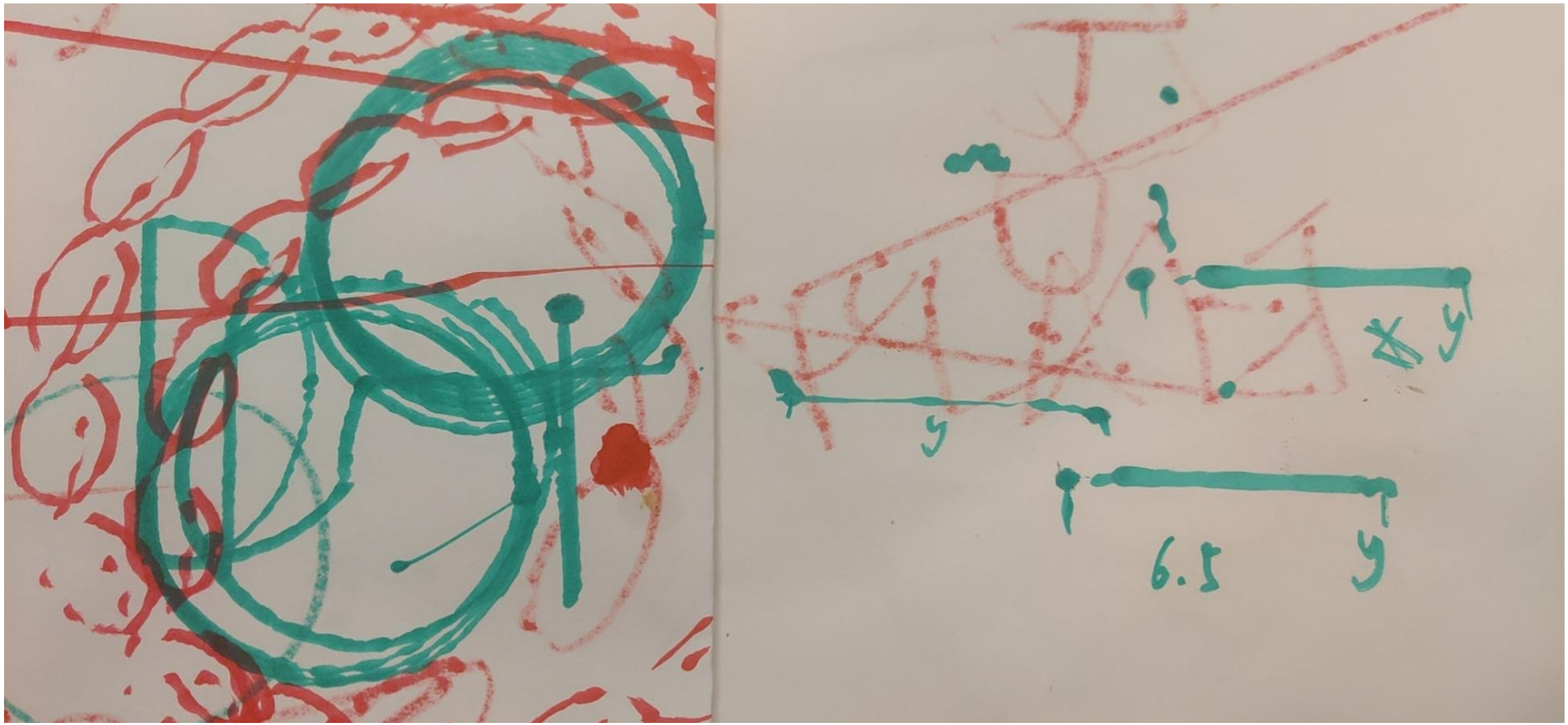


Project build and functional test results

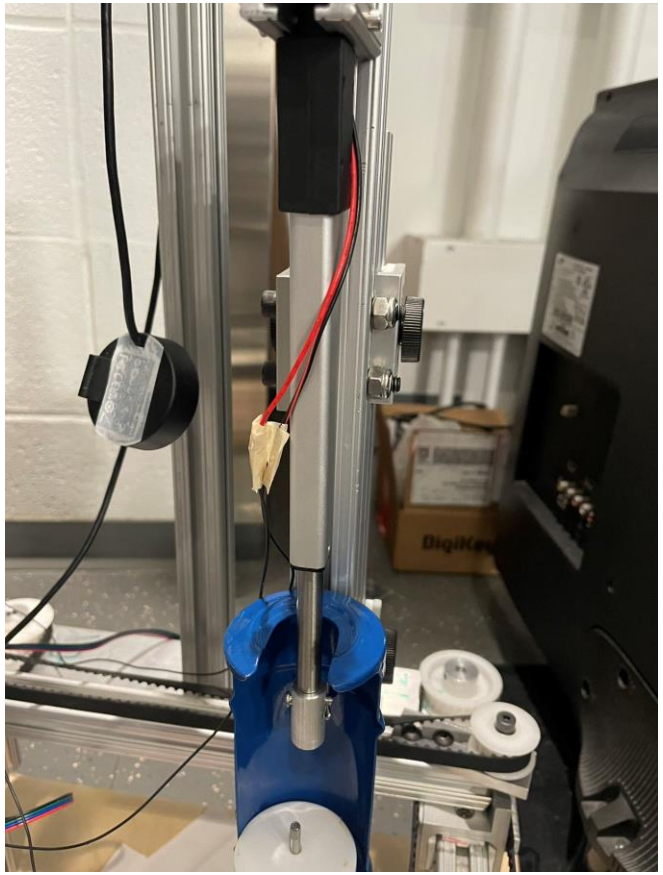
Requirements

- Mechanical parts strong enough to support the system under full load.
- Speed of the motor and extrusion of cream are consistent, not causing cream to pile up or not forming a continuous line.
- Motors powerful and accurate enough to drive the movement and injection of syringe under full load.



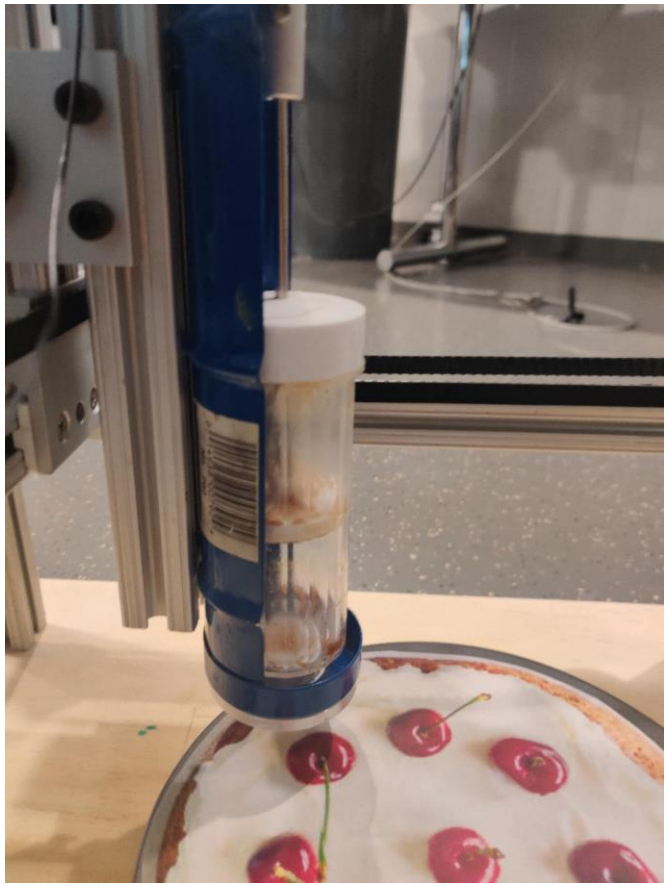


Mechanical Subsystem



Linear actuator

Stepper motor



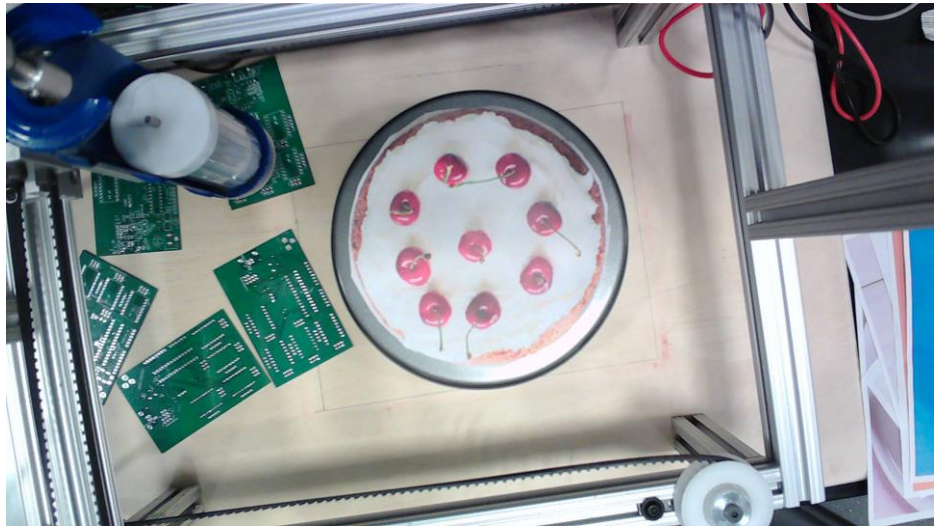
Cream Injector

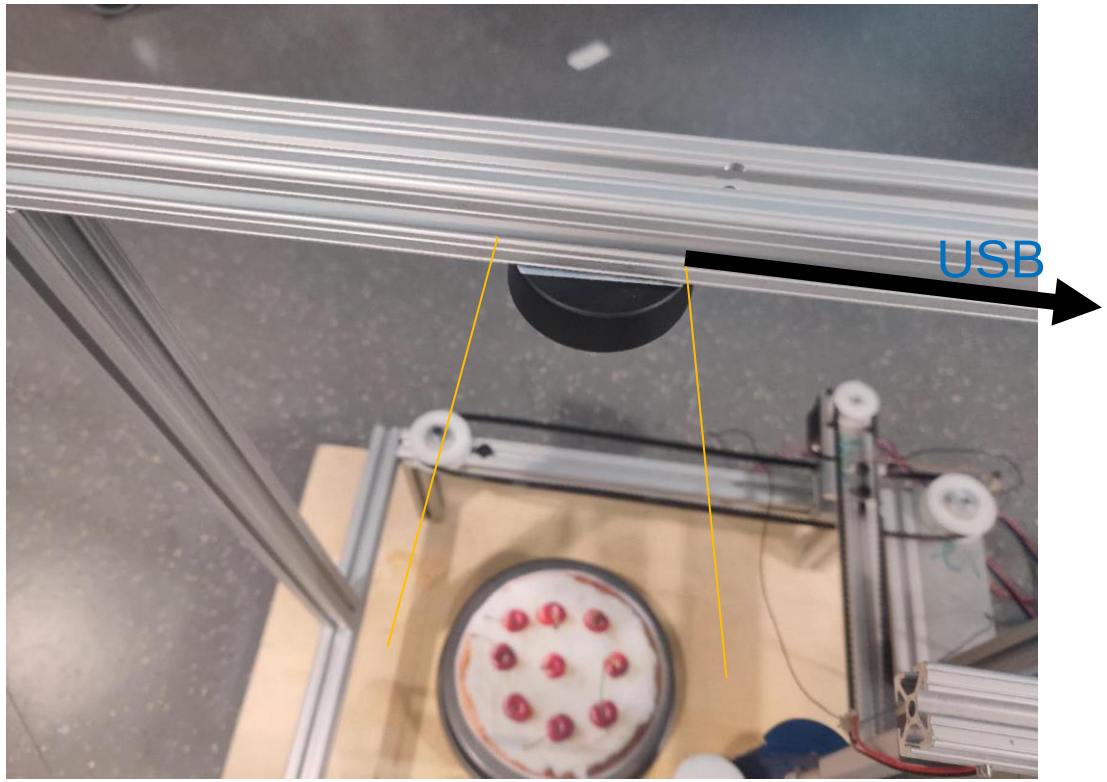
Requirements

- Optical resolution of the camera is better than 0.5cm in physical coordinate, when hanging at 0.6m looking down
- Mapping from camera pixel coordinate to physical coordinates results in a maximum deviation of 0.5cm
- DNN correctly recognize cake position and outline 90% of the time
- Robust algorithms projecting a trajectory onto movements on the x and y axis.

Optical resolution of the camera is better than 0.5cm

- 1080p webcam
- Viewing area: ~700mm x 1200mm
- ~0.7mm/pixel

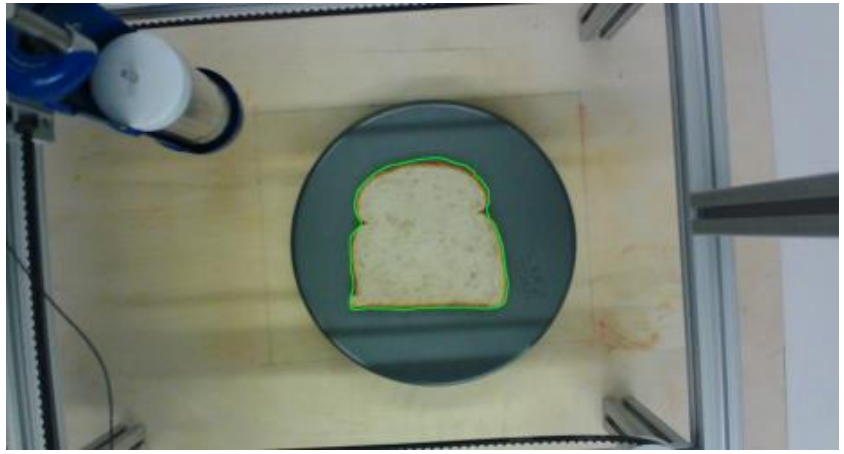




DNN correctly recognize cake position and outline 90% of the time

Algorithm: FoodSAM (based on Segment Everything)

- Semantic segmentation: finding boundary & position all-in-one



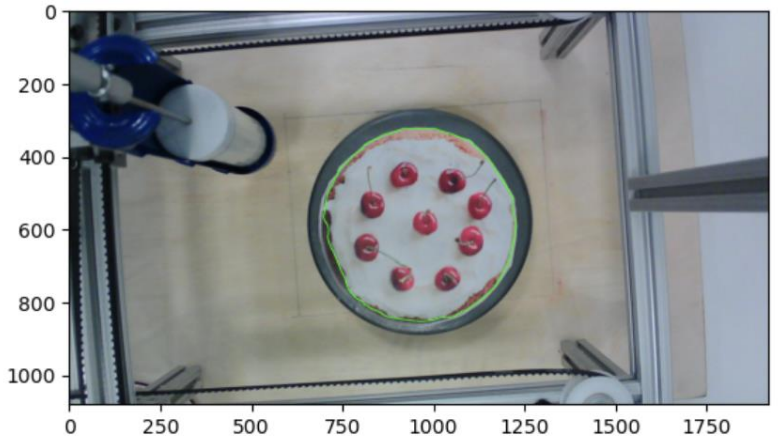
Get the recognized outline and visualize

In [88]:

```
mask = get_mask(0)
# mask = get_mask_enhanced(0)

cake_img = cv2.imread('tests/input/0.jpg')
cake_img = cv2.cvtColor(cake_img, cv2.COLOR_BGR2RGB)

contour = get_cake_countour(mask)

img_copy = cake_img.copy()
cv2.drawContours(img_copy, [contour], -1, (0, 255, 0), thickness=2)

plt.imshow(img_copy)
```

Out [88]:

```
<matplotlib.image.AxesImage at 0x1943dcd0f90>
```

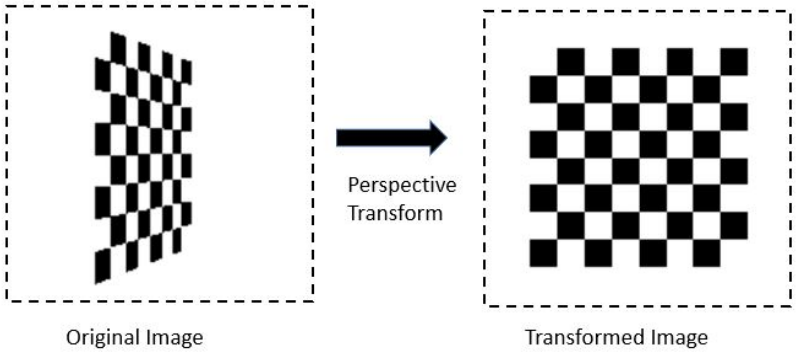


Mapping from camera pixel coordinate to physical coordinates results with a deviation less than 0.5cm

Calibrated by chessboard image:

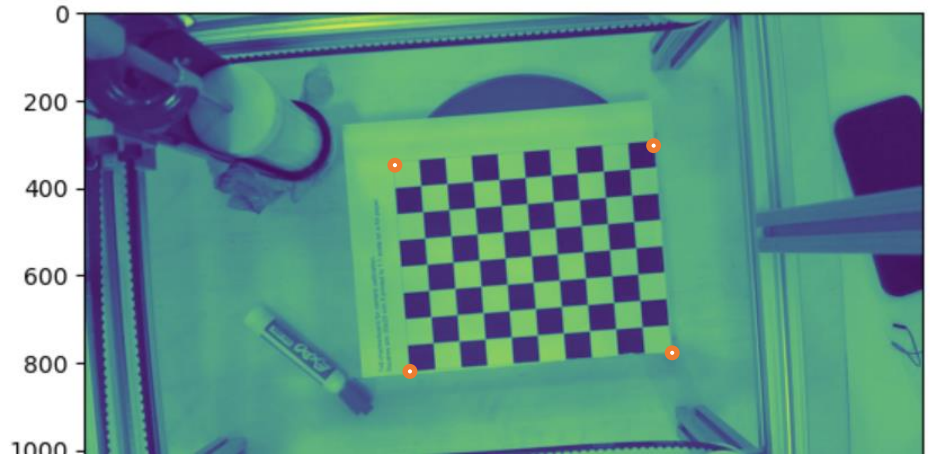
Critical points:

- We know their physical coordinates
- Computer recognize them in pixels
- Allows to do a perspective transform



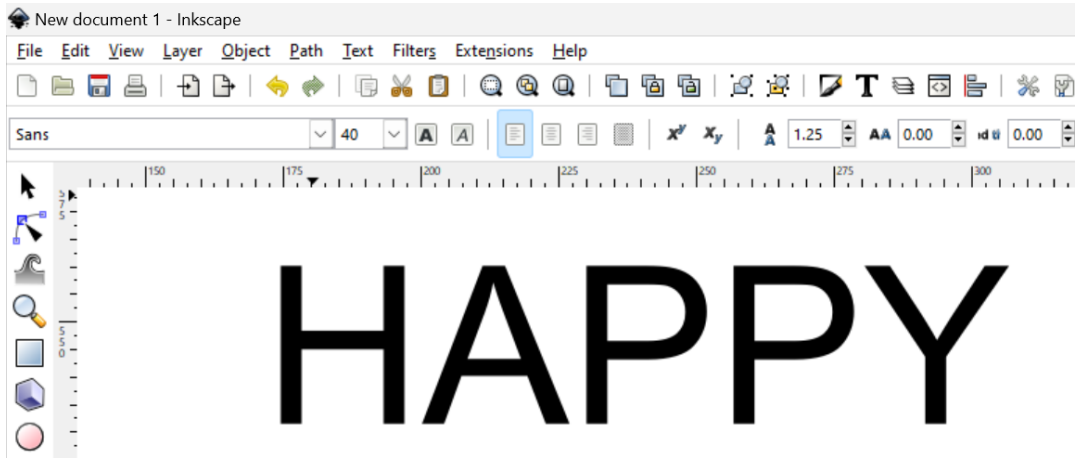
Read the chessboard image and get the calibration

```
cb_img = cv2.imread('tests/cb0.jpg')
cb_img = cv2.cvtColor(cb_img, cv2.COLOR_BGR2GRAY)
plt.imshow(cb_img)
```



Load arbitrary drawings

Vector arts

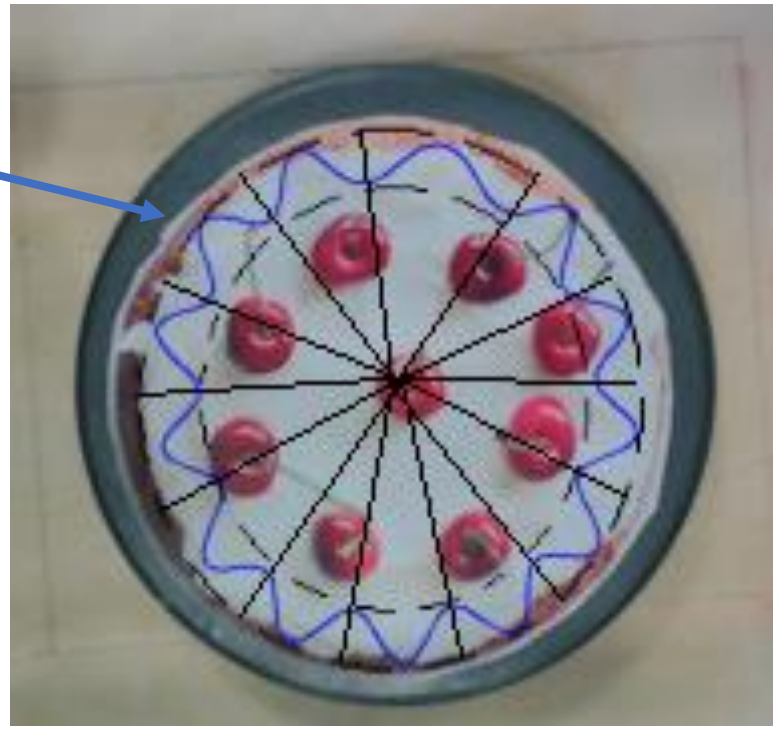


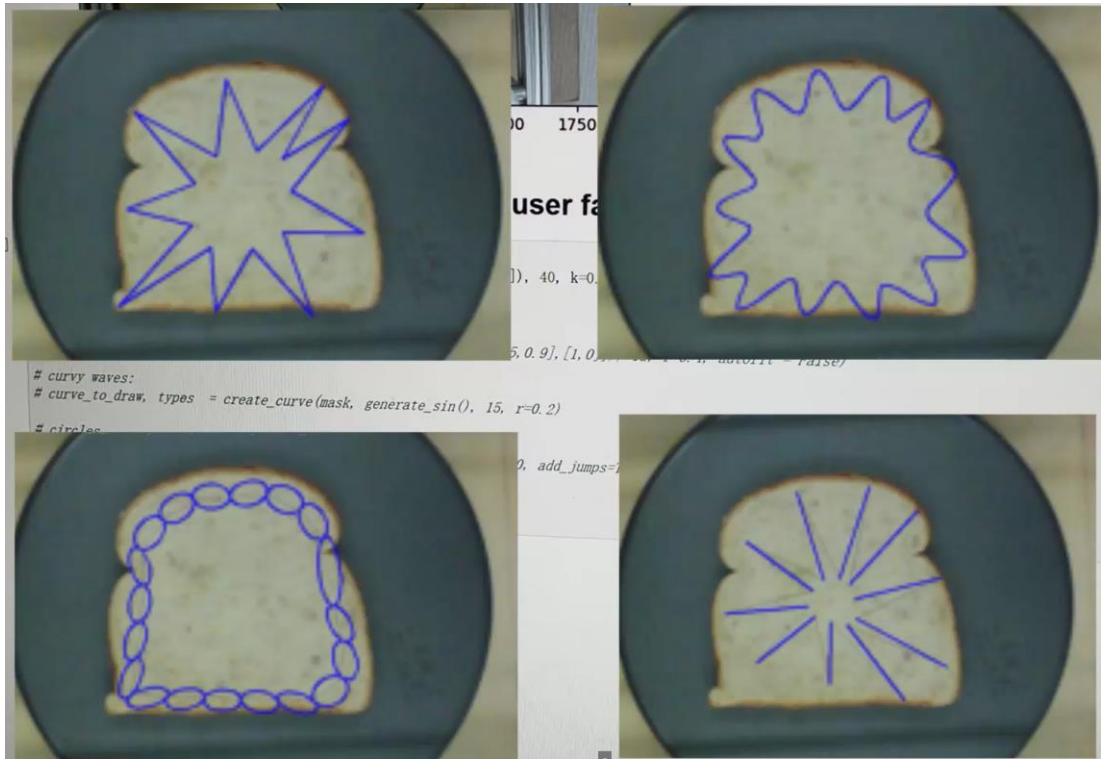
Autofit

Generate decoration styles according to the outline

Blueprint:

Map and copy





Sample points on curves, encoding movement and transmit

```
mm_per_sample = 2
curve_to_draw_phy = resample_pts(curve, mm_per_sample,  types)

x_mm_per_step = 0.35
y_mm_per_step = 0.40
stepper_moves= generate_stepper_move(curve_to_draw_phy, x_mm_per_step, y_mm_per_step)
```

Enoding:

XXXX XXXX

x-move y-move
(4-bit 2's compliment)

Special commands:
1xxx 1000: jump according to next two numbers

Codes that won't go into movement buffer:
0010 1000: retract linear actuator
0011 1000: extend linear actuator
0100 1000: stop linear actuator
0101 1000: start/resume
0110 1000: reset

Requirements

- Power supply can maintain a maximum total output that is enough to motivate all the electrical components (especially motors).
- ~~Voltage regulators sustain a stable 5V for the logical components (MCU, USB bridge, drivers etc.) under max load.~~
- The control signal to the motor is accurate enough that it deviates from the desired part within 1 cm.

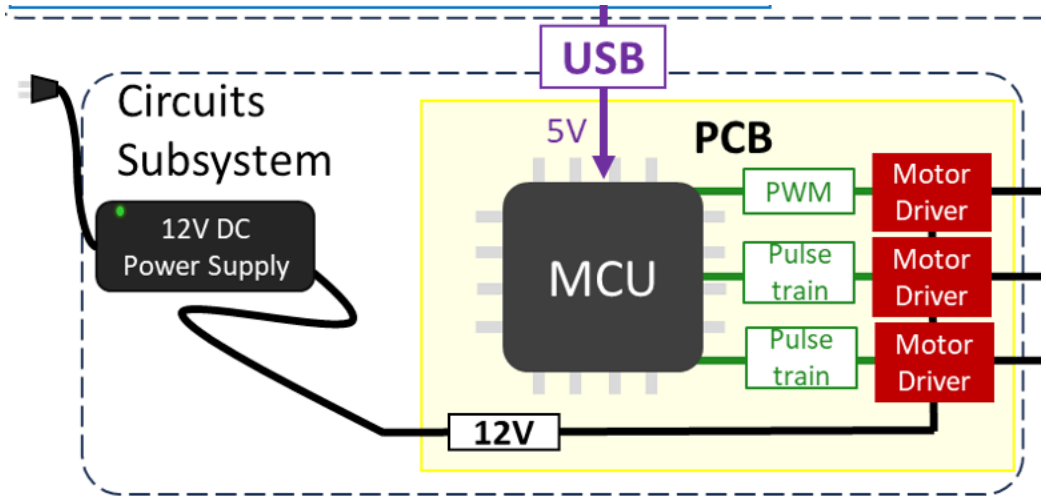
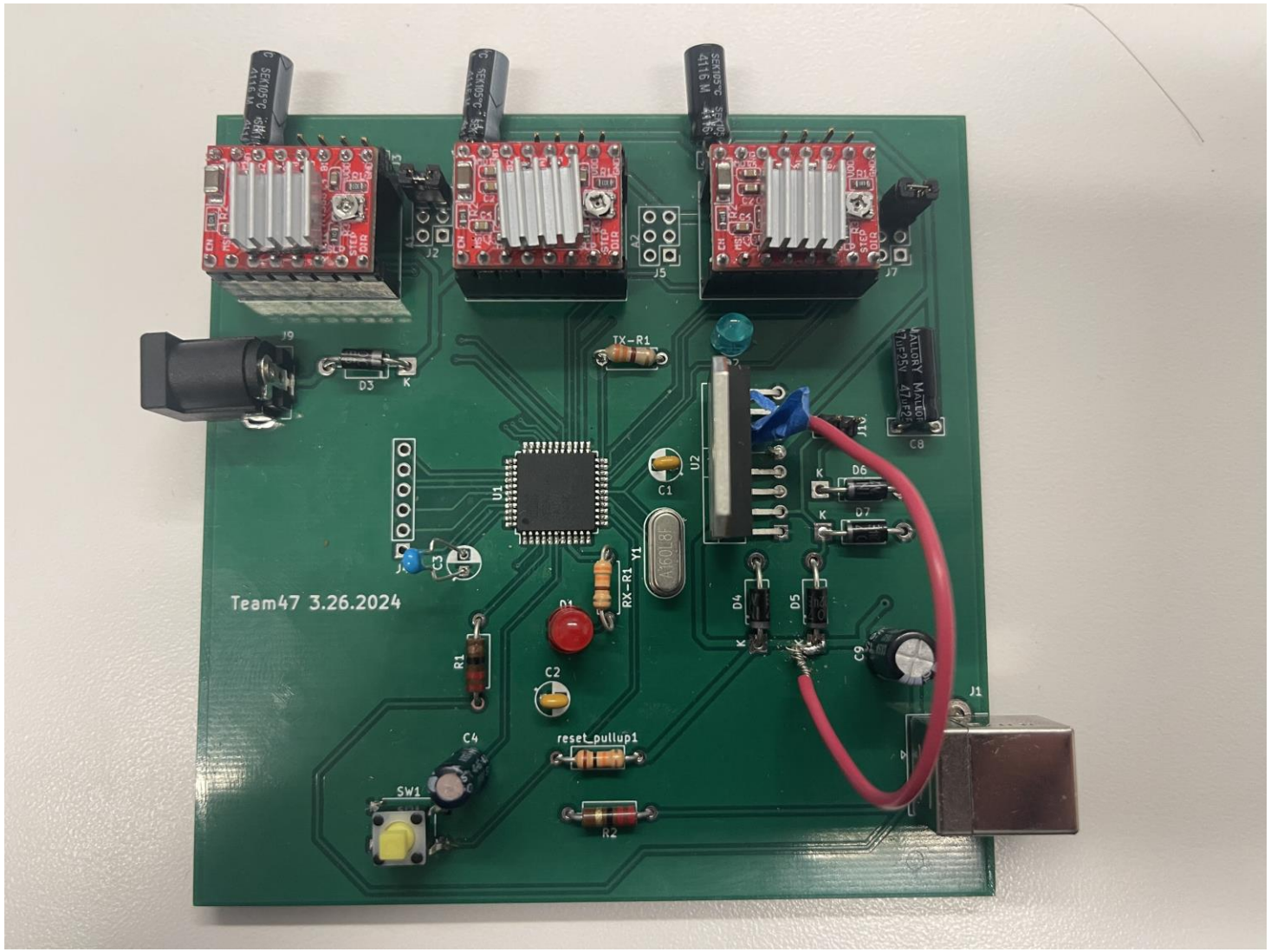


Photo of PCB

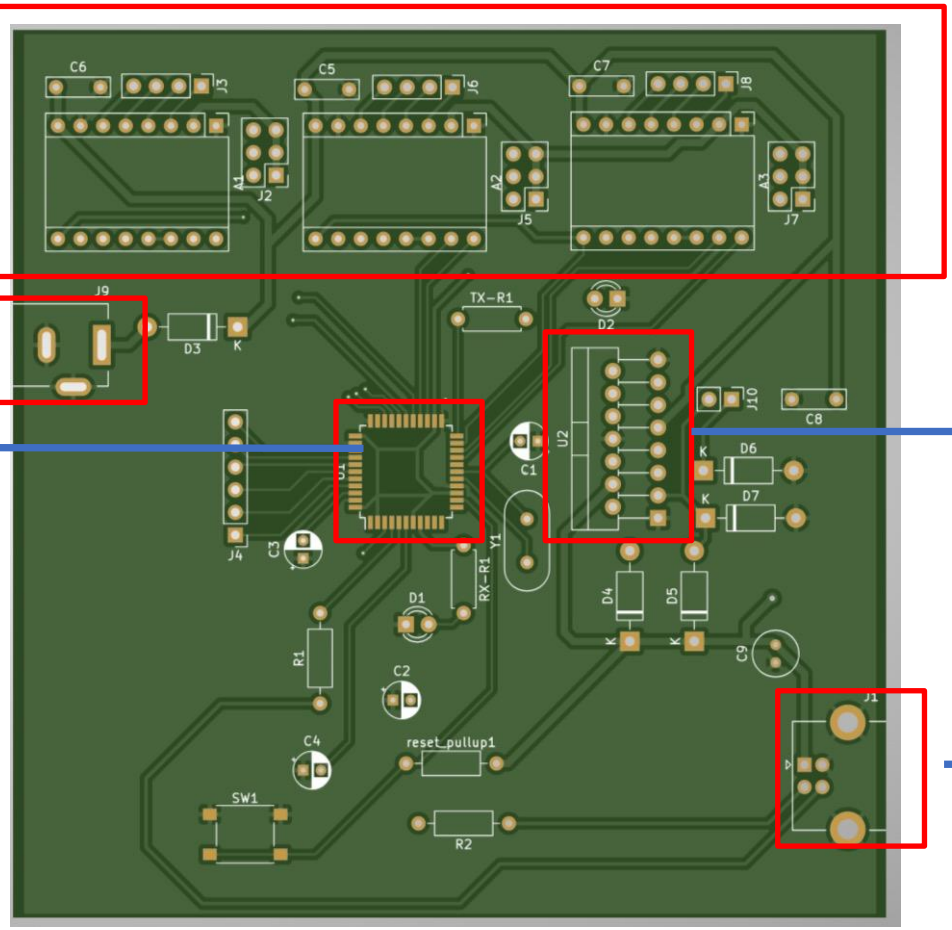

Team47 3.26.2024
TX-R1
RX-R1
reset_pullup1
D1
D3
D4
D5
D6
D7
U1
U2
C1
C2
C4
C8
C9
R1
R2
Y1
SW1
J1
J2
J5
J7
J9
STEP
DIR

PCB Overview

Stepper Motor driver (A4988)

12V Power

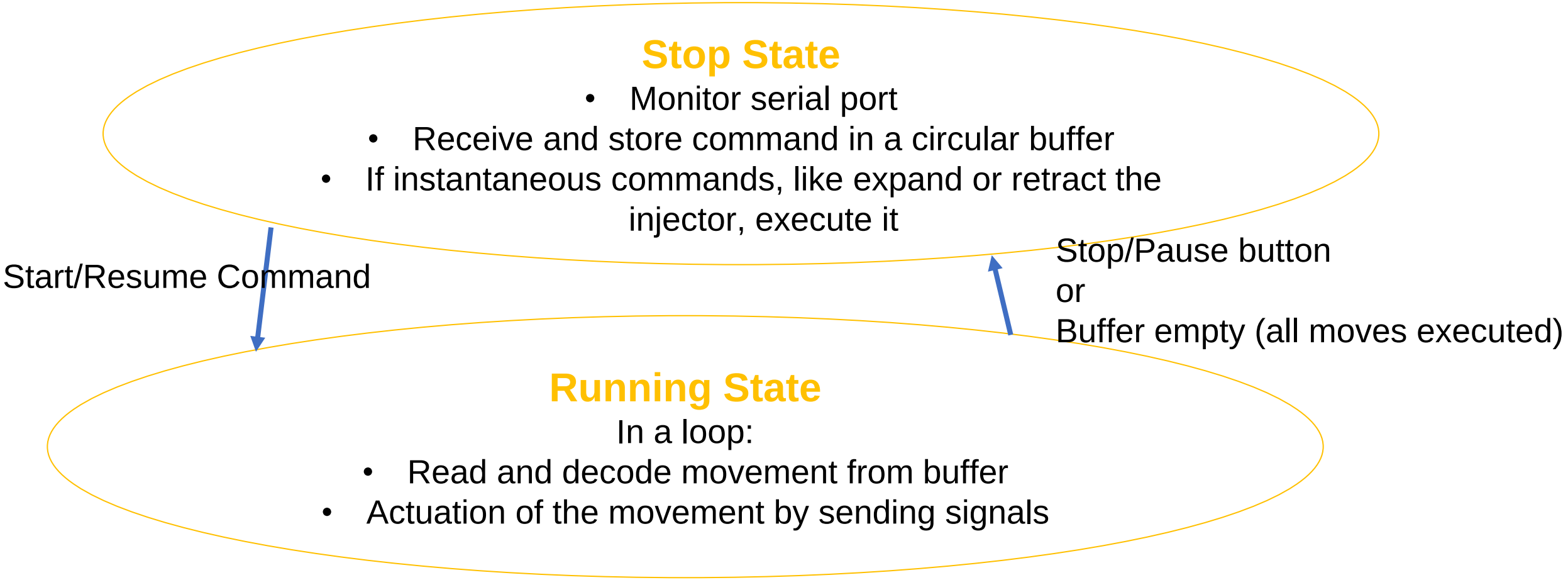
MCU (ATMega32U4)



DC Motor Driver (L298N)

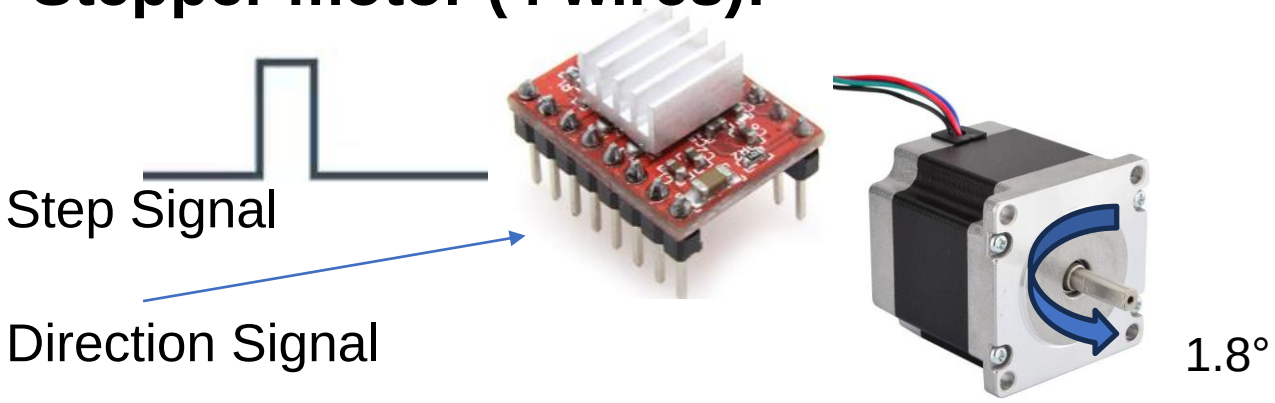
USB/5V Power supply

Embedded system: Program on microcontroller



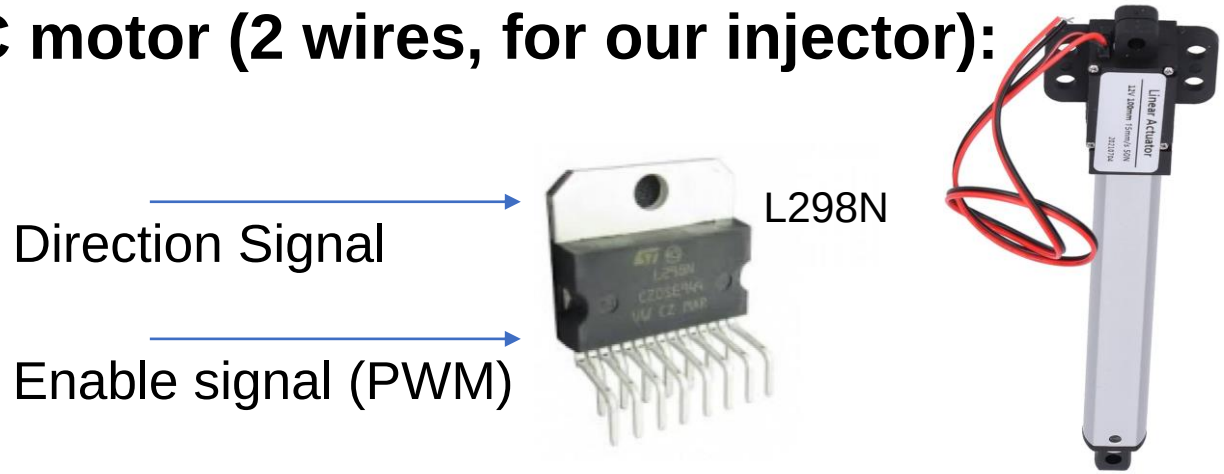
Embedded system: Control signals for motors

Stepper motor (4 wires):



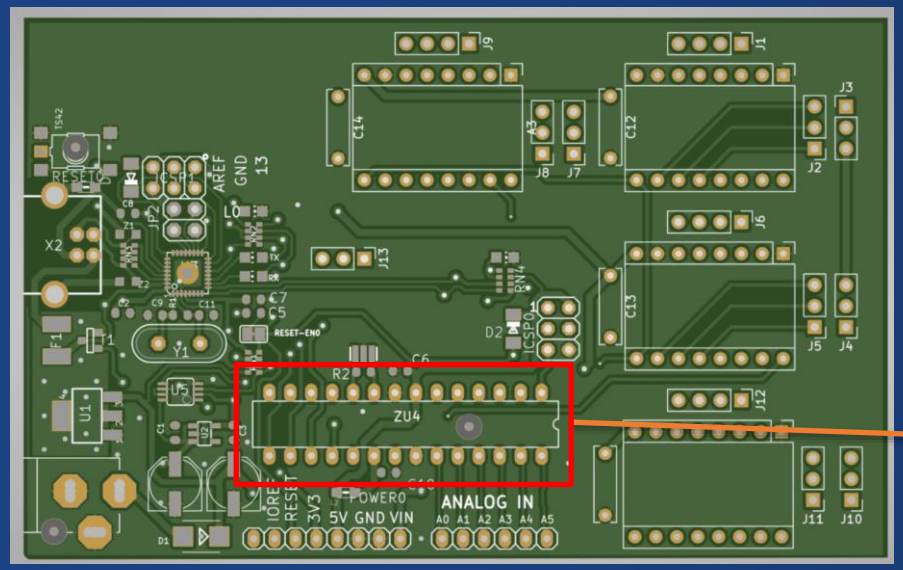
Optimal for open-loop control
Generated by accelStepper library

DC motor (2 wires, for our injector):



Challenge and solution

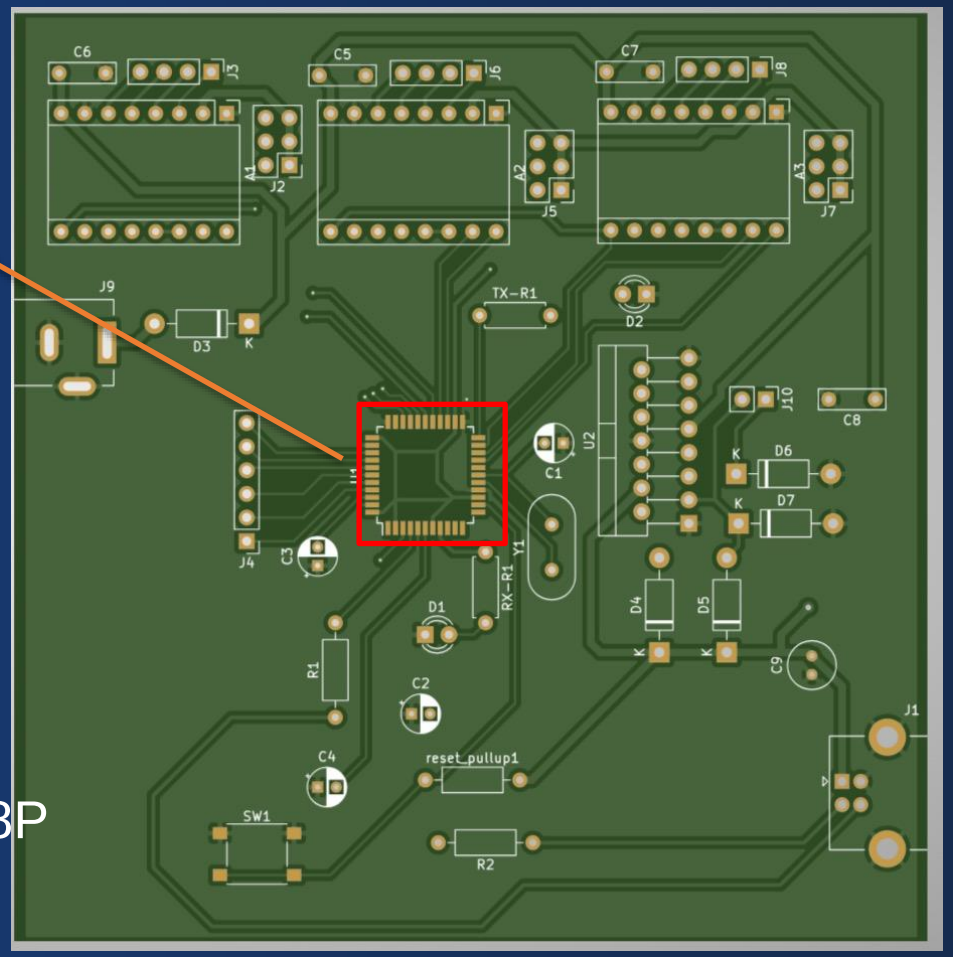
Challenge and solution



Old version

ATMega32U4

ATMega328P

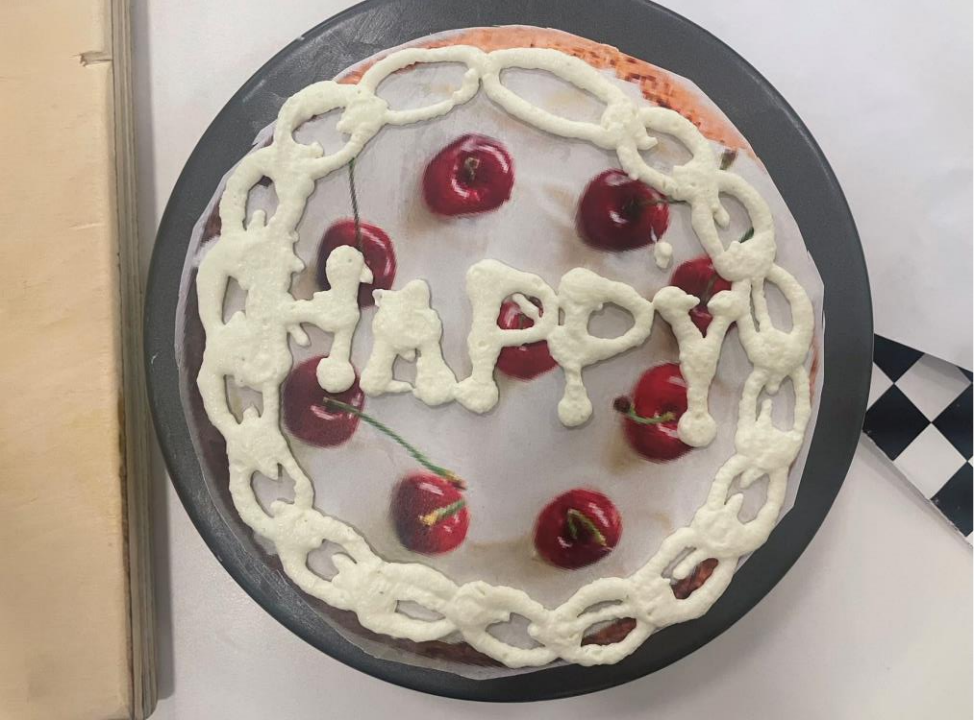


Updated version

Conclusion & Future

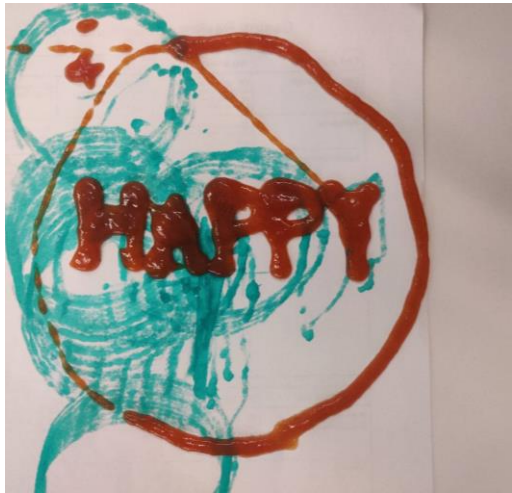
Conclusion

- Applied knowledge related to the circuit design, embedded system, robotics, and computer vision.
- Facilitated team cooperation
- Practiced time management
- Had funs and enjoyed the achievements

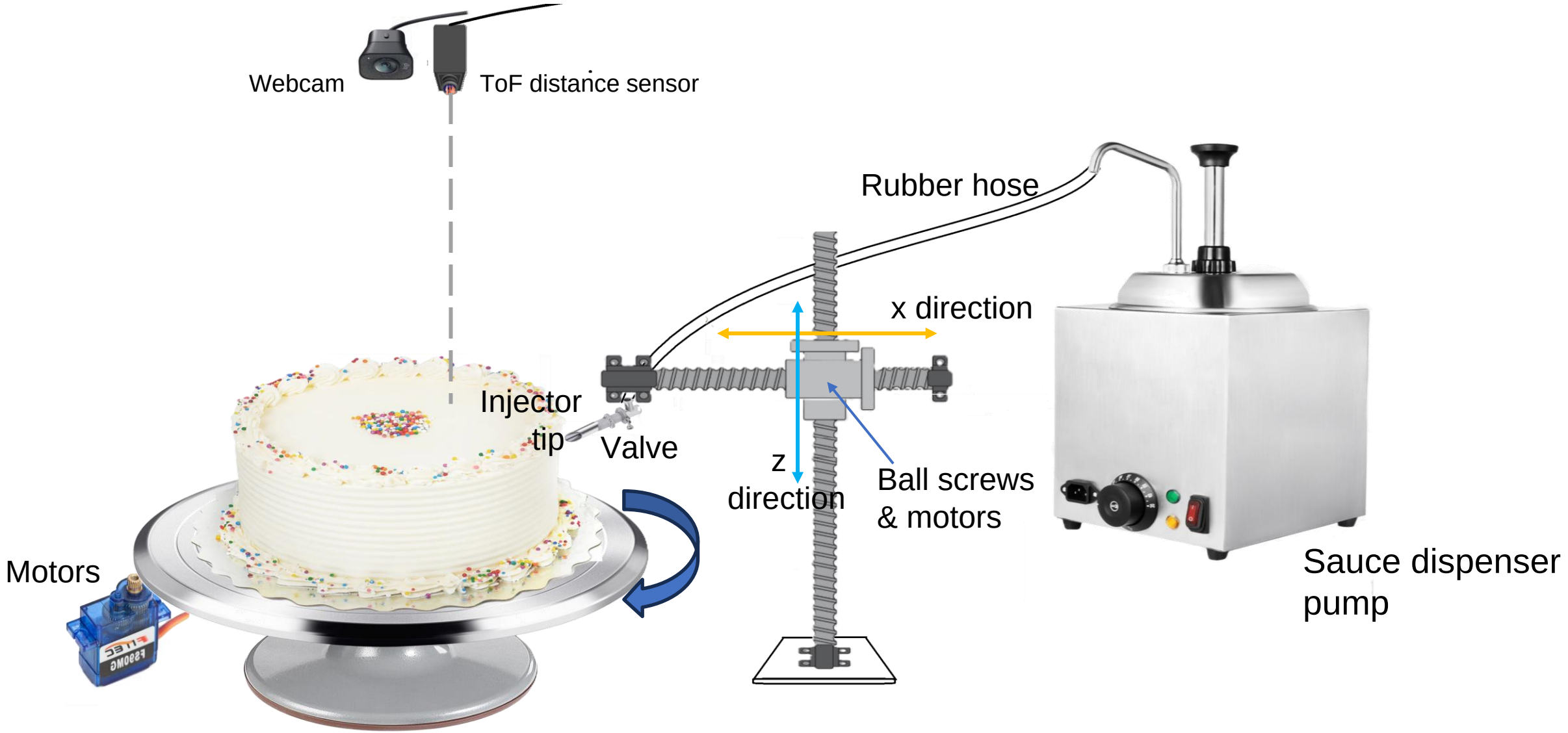

HAPPY


HAPPY


HAPPY

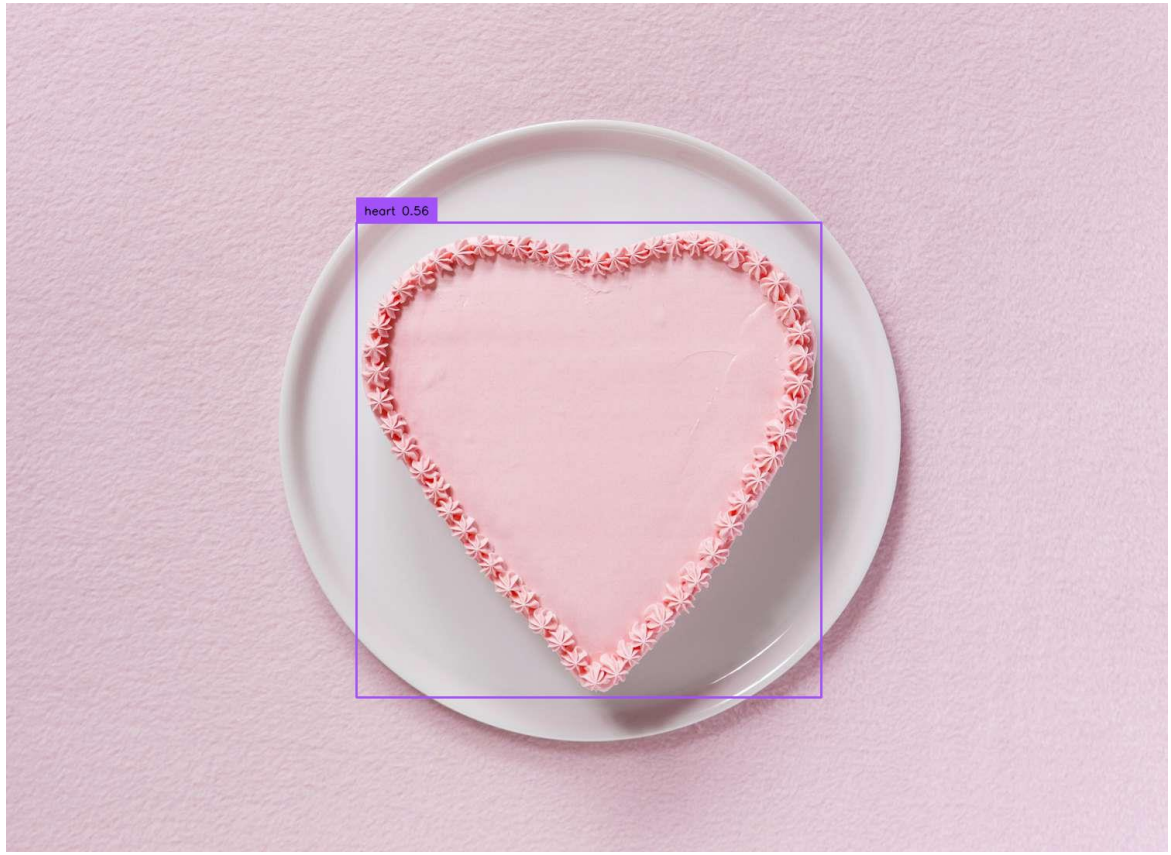

HAPPY

Future work: Next gen cake decorator



Future work: Better detection algorithm

- LLM Prompt-based object detection & segmentation:



https://github.com/AILab-CVC/YOLO-World

Thank you!

Questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN