ECE 445 Design Document

TA: Akshatkumar Sanatbhai Sanghvi

Henry Guan (henryg2)

John Kim (jaehank2)

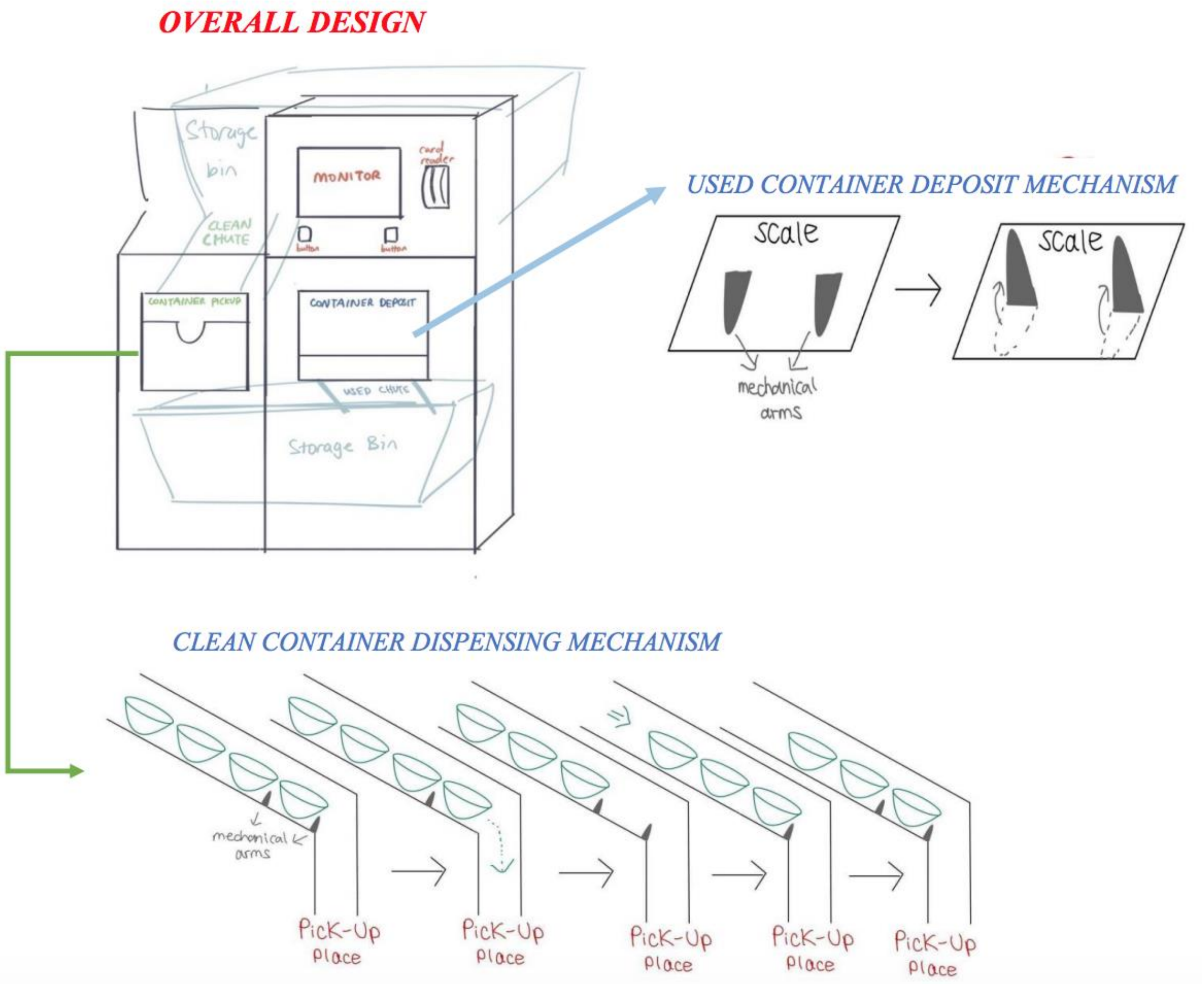
Ariocie Liang (arliang2)

February 24, 2022

Introduction

We will build Good2Go Automated Token Exchanger (GATEr) for the people that want to take food out from the dining hall and eat it wherever they go. The people who take food out from the dining hall using the current system for Good2Go (G2G) face a big problem when it comes to the number of steps one has to take in order to obtain a brand new G2G container. Also, the current state of G2G is that a person wanting to get a G2G container will have to wait in the same line as other people who just want to eat in the dining hall. This results in longer time spent and having unnecessary human to human interactions. GATEr streamlines the process of obtaining a brand new G2G container by removing human interaction and supervision, as well as removing the need to deal with waiting in a line of students wanting to eat in the dining hall. Thus the product will cut down on the amount of steps and time it takes to obtain a new G2G container immensely and provide a more efficient solution to the current problem of G2G.

Visual Aid:



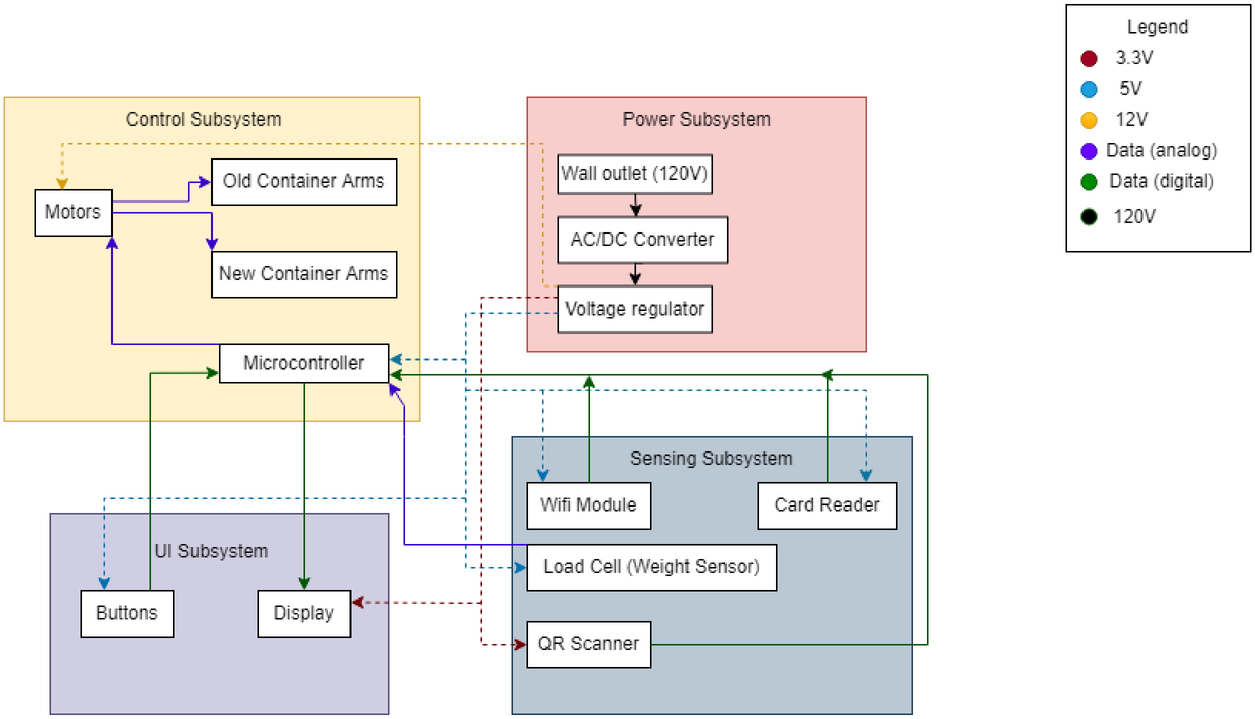
High-level Requirements List:

1. Deliver the entire process (from receiving container/token to dispensing a new one) in less than 15 seconds. This time is calculated assuming 0 delay in user's action. Thus, the time taken from receiving a container to dispensing a new one or updating token from the machine's side should be less than 15 seconds.

2. Dispense exactly one new container at a time indefinitely without any jamming. The system should never allow more than one container to be dispensed upon a single request.

3. Correctly detect invalid containers that are invalid G2G containers (un-identified/nonexistent QR code) or are overweight by more than 50 grams compared to the base weight so that containers holding too much food waste are rejected.

Block Diagram



Requirements & Verification Tables

Sensing Subsystem

Requirements	Verification
Can scan QR code on transparent G2G containers	- Check if the white sticker backening ensures correct QR scan by having it placed on a transparent glass with transmittance greater than 70%
Execute action (whether dispensing or receiving) only for valid G2G containers	- Run different test cases for both valid and invalid containers. Invalid container runs will include {valid G2G containers with too much waste (> 50 grams)} and {containers with no QR code}. The machine should not execute any action for at least 5 seconds or until a valid container is placed. For valid containers, the machine should carry out an action such as dispensing or receiving in less than 5 seconds.

Control Subsystem

Requirements	Verification
Container should receive old container in less than 9 seconds and dispense a new container in less than 6 seconds	- Conduct 10 repetitive tests, timing the receiving starting when the user places an old container. The timing of dispensing is taken right after the receiving to ensure total time taken is less than 15 seconds.
Dispense exactly one new container upon a single request without any mechanical faults such as jamming	- Conduct 10 repetitive tests and compare the distance of the arms to their original position, ensuring it is less than 0.5mm to show it can be done indefinitely

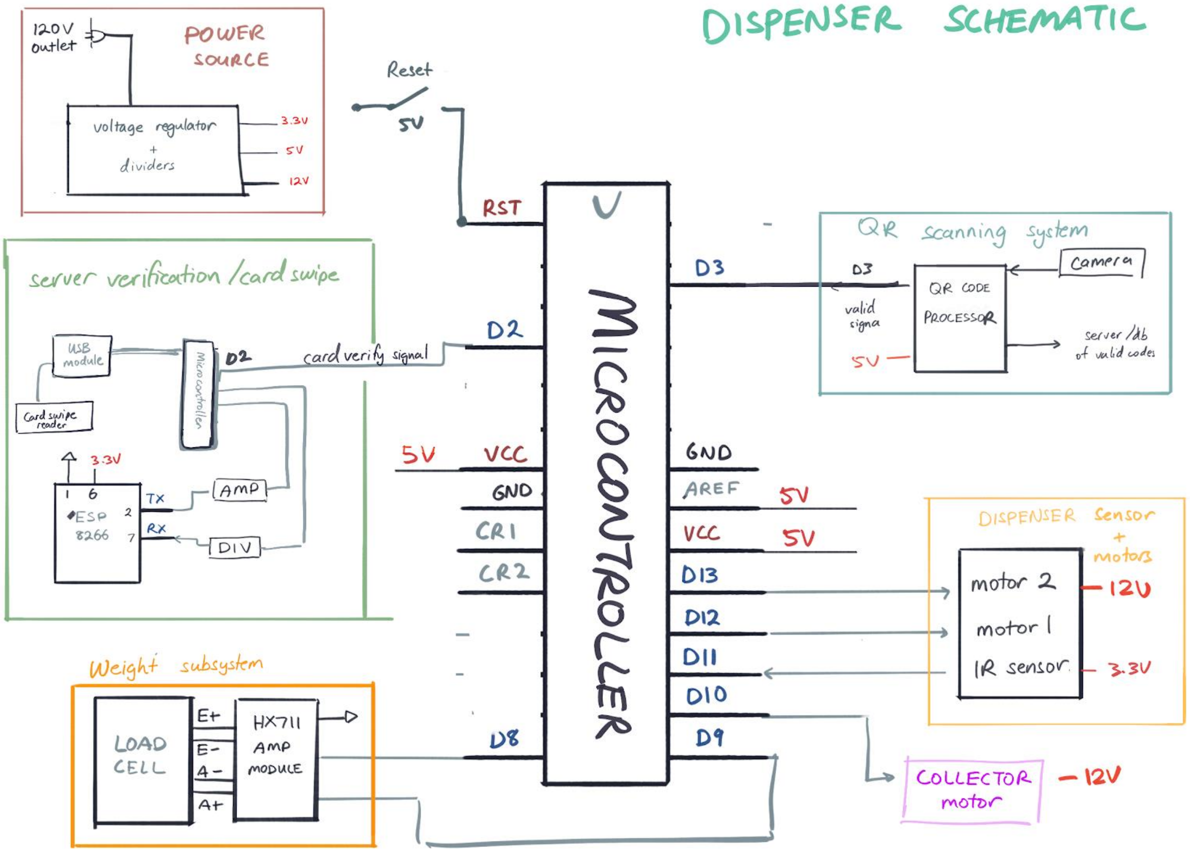
UI Subsystem

Requirements	Verification
Correctly keep track of the number of tokens per user.	- Insert 4 old containers and convert them to tokens. Assume 4 tokens are correctly stored. Then using the same user card, convert the tokens to new containers until no tokens are left. Ensure the number of new containers received equals the number of old containers put in.

Power Subsystem

Requirements	Verification
Power all the components with constant voltage in accordance with the voltage ratings in the block diagram	- Test the voltage running through each component with an oscilloscope and observe if the voltage supplied exceeds a 5% variance

Circuit Schematics



Tolerance Analysis:

The mechanical aspect of the dispensing and retrieval system is one of the most critical components of our project. The main functionality of the machine highly depends on accurate and correct retrieval of old containers to be thrown into the storage bin. This action is executed by a pair of 5cm arms powered by a stepper motor with a 1.8 degree step angle with 200 steps per revolution. Since the placement of the G2G container on the scale may vary in position, the arms need to provide enough slant and force to retrieve the container without having the container collide with any side of the walls.

To ensure such performance, it will be tested with different angles for the arm ranging from 45 to 90 degrees with a step size of 5 degrees. The placement of the pair of arms will be spaced to maintain a distance of 15cm to ensure no containers can slip through. The speed of the motor will be calculated as follows to enable the retrieval mechanism:

$$\text{Motor rotation } (^\circ) = \text{step angle } (^\circ/\text{step}) \times \text{number of pulses}$$

$$\text{Motor speed } (\text{r/min}) = \text{step angle } (^\circ/\text{step}) \div 360\ (^\circ) \times \text{pulse rate } (\text{Hz}) \times 60$$

If the constant velocity motion profile for the stepper motor does not generate enough force to throw back the old containers, the acceleration profile can be used with acceleration torque and a constant rate of pulses. For the dispensing system, we will incorporate a 1 second delay between when both actuate so that the first arm can allow a container to be dispensed, which will then allow for the next container to preload itself via the second arm dropping. The upper bound on the dispensing is 9 seconds with motors for each mechanical arm having 1 second motion for the up and down movements. Thus the 1 second delay is calculated using the bound below where ‘a’ is the first mechanical arm and ‘b’ is the second:

$$\text{a down (1s)} + \text{wait (2s)} + \text{a up (1s)} + \text{delay (1s)} + \text{b down (1s)} + \text{wait (2s)} + \text{b up (1s)} \leq 9\text{s}$$

Ethics and Safety

This project has potential pitfalls with ethics and safety issues that can conflict with the IEEE Code of Ethics and the ACM Code of Ethics, with safety and privacy of the user being two standout issues (ACM, IEEE, 2022). Namely, the user could get their finger stuck in the motor area with the chutes for dispensing and accepting containers, which is a safety concern. Additionally, with our database of user info associated with each RFID card, the user could be prone to a data leak, which is a big concern for privacy, especially in current times where data is extremely valuable (ACM, 2022).

It is our utmost priority to ensure that people’s privacy is protected and they are aware of safety concerns (ACM, 2022). As noted in the IEEE and ACM Code of Ethics, we will treat all persons fairly and not discriminate against people based on race, age, disability, gender, national origin, gender identity, or gender expression (ACM, IEEE, 2022). We will also not engage in harassment of any form, including sexual harassment or bullying behavior (ACM, IEEE, 2022). Additionally, we will also accept honest criticism of our project as well as properly credit contributions by others (IEEE, 2022).

References:

[1] *IEEE Code of Ethics*. IEEE. (n.d.). Retrieved February 20, 2022, from https://www.ieee.org/about/corporate/governance/p7-8.html

[2] *ACM Code of Ethics and Professional Conduct*. Code of Ethics. (n.d.). Retrieved February 20, 2022, from https://www.acm.org/code-of-ethics