Smart Plastics Recycling System

Electrical & Computer Engineering

Team 7

Jennifer Chen, Smruthi Srinivasan, Jason Wright

12/4/2023

Agenda

1. Introduction
2. Objective
3. Design & Results
4. Successes & Challenges
5. Conclusion

Team Introduction

Jason Wright
Hardware
Electrical Engineering

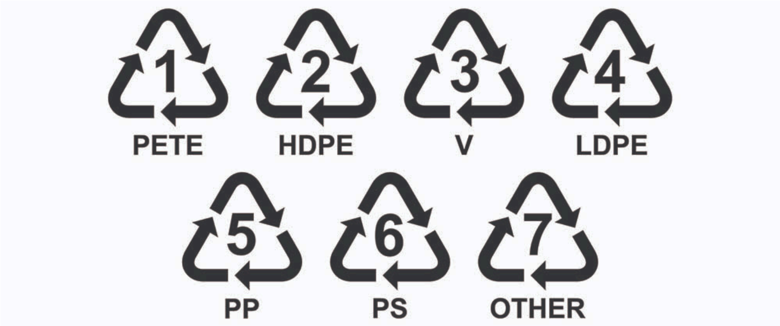
Jennifer Chen
Software
Computer Engineering

Smruthi Srinivasan
Software
Computer Engineering

Project Background

Problem

- Plastic materials are commonly recycled improperly, resulting in contamination of all other recyclables
- Not all plastics are recyclable – it depends on the material, which is denoted by a symbol, and the jurisdiction



Goal

- Design a system that takes in a camera input, learns what the symbol is, and determines if the item can be recycled in the user's area

Project Solution

- Imaging system captures the symbol denoted on a plastic container
- Machine learning model to read symbol on the container
- User interface determines if plastic can be recycled in the user's area and displays additional information
- Sorting actuator tips plastic into recycling or trash

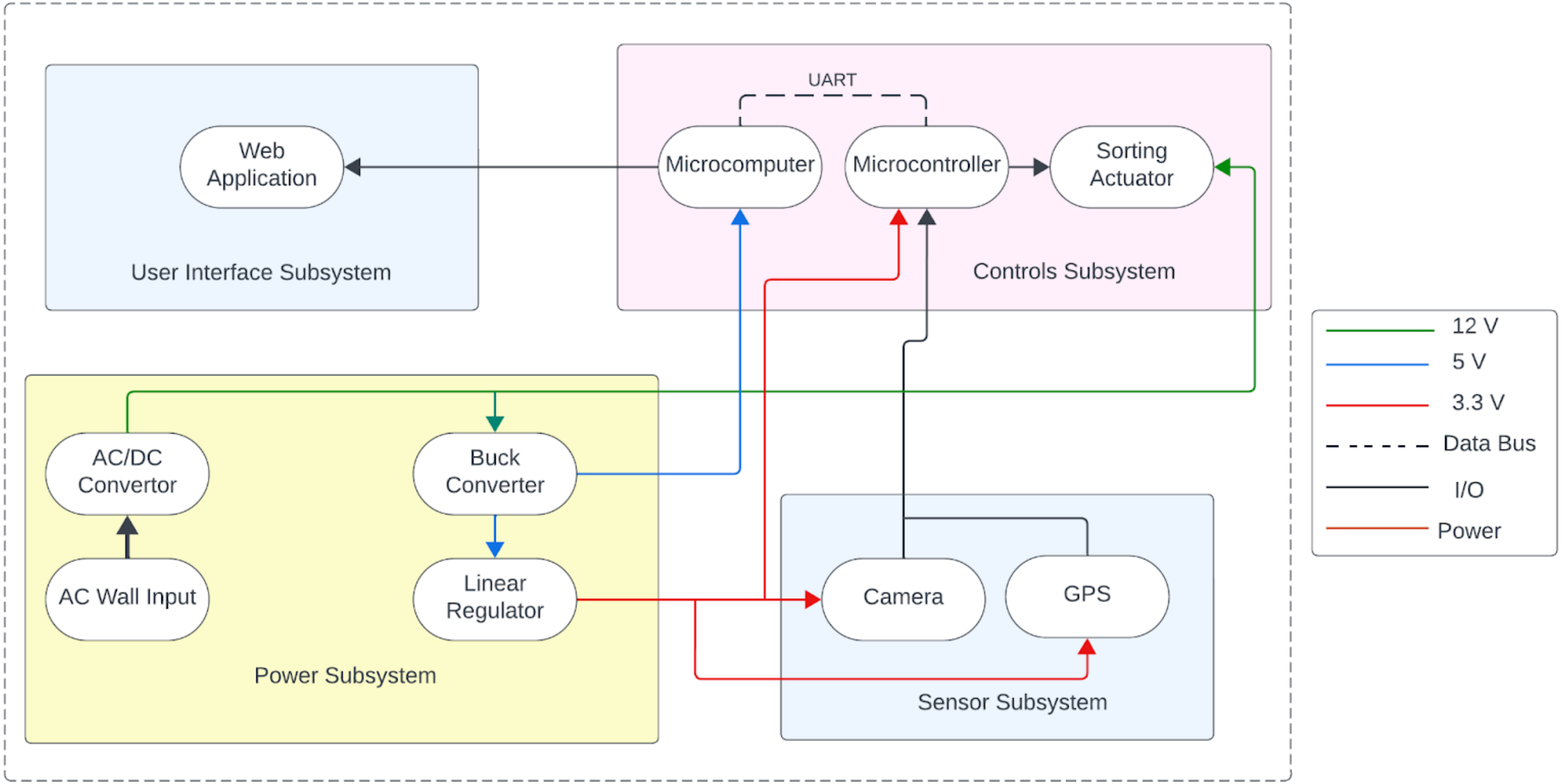
High Level Requirements

1. Camera detects plastic being positioned in front of it and system identifies symbol on the plastic

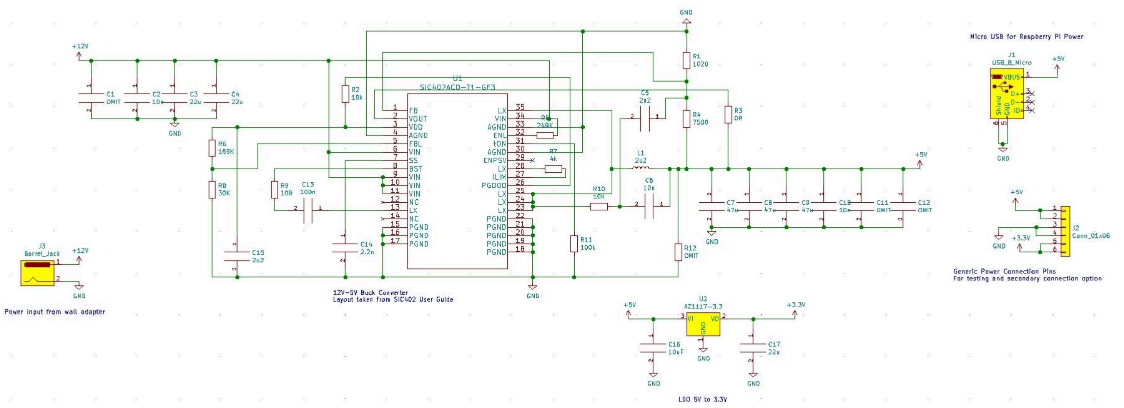
2. Determine the user’s location and pull recycling data in the area

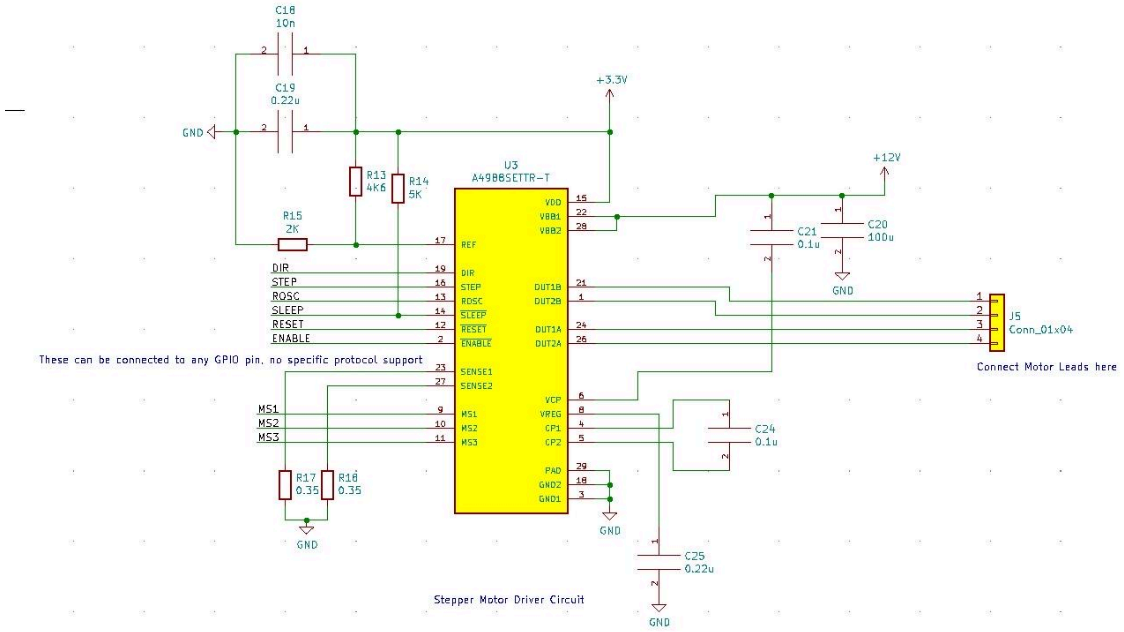
3. System determines if container is recyclable, places into proper bin, and UI shows information about plastic and centers that accept it

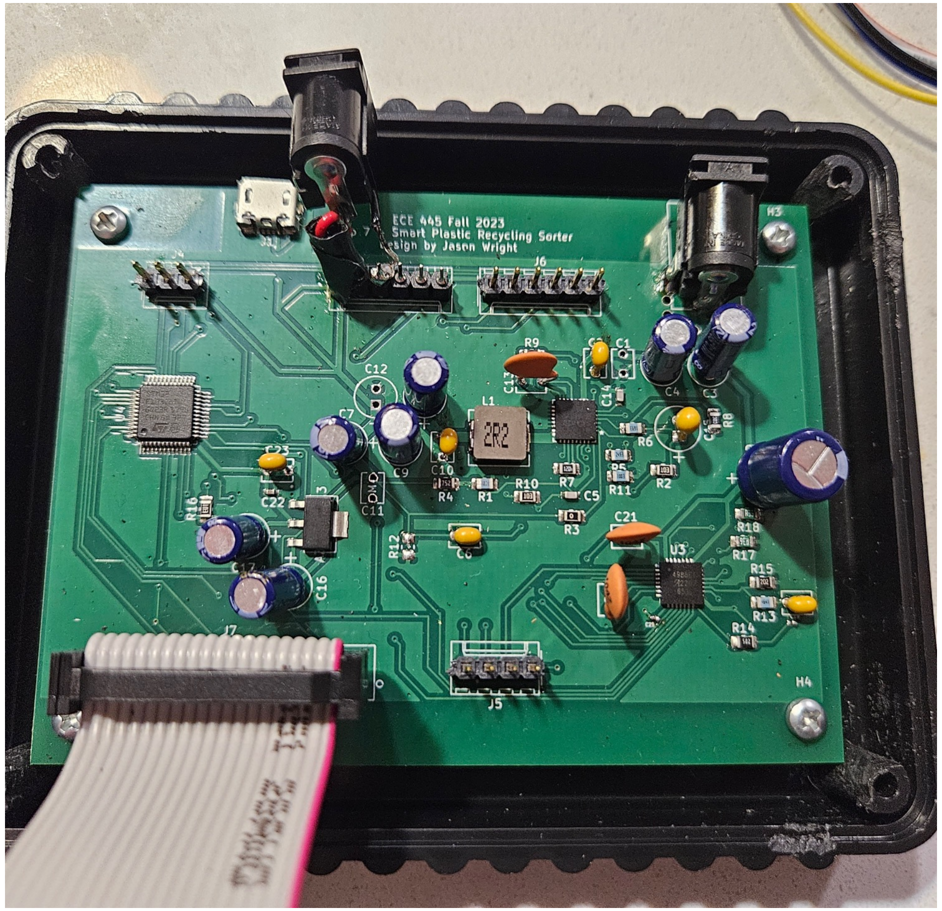
Block Diagram



PCB Design


Micro USB for Raspberry Pi Power
U1
SIC402ACD-T1-GE3
Power input from wall adapter
12V-5V Buck Converter
Layout taken from SIC402 User Guide
LDO 5V to 3.3V
Generic Power Connection Pins
For testing and secondary connection option


C18
10n
C19
0.22u
GND
+3.3V
R13
4K6
R14
5K
R15
2K
U3
A4988SETTR-T
+12V
C21
0.1u
C20
100u
DIR
STEP
ROSC
SLEEP
RESET
ENABLE
These can be connected to any GPIO pin, no specific protocol support
J5
Conn_01x04
Connect Motor Leads here
MS1
MS2
MS3
C24
0.1u
R17
0.35
R16
0.35
C25
0.22u
Stepper Motor Driver Circuit


ECE 445 Fall 2023
Smart Plastic Recycling Sorter
design by Jason Wright

Mechanical Design

Proximity Sensor

Camera Sensor

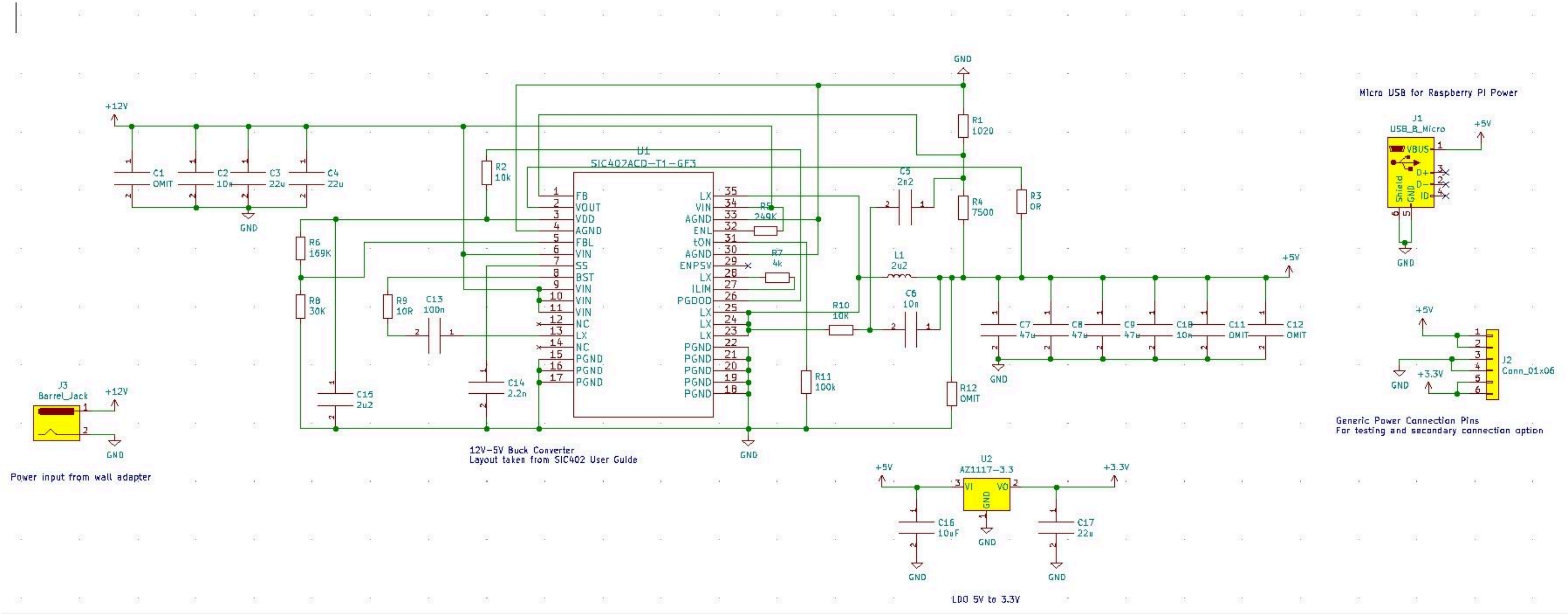
Motor Controlled
Tilting Platform

Trash

Recyclable

Power Subsystem

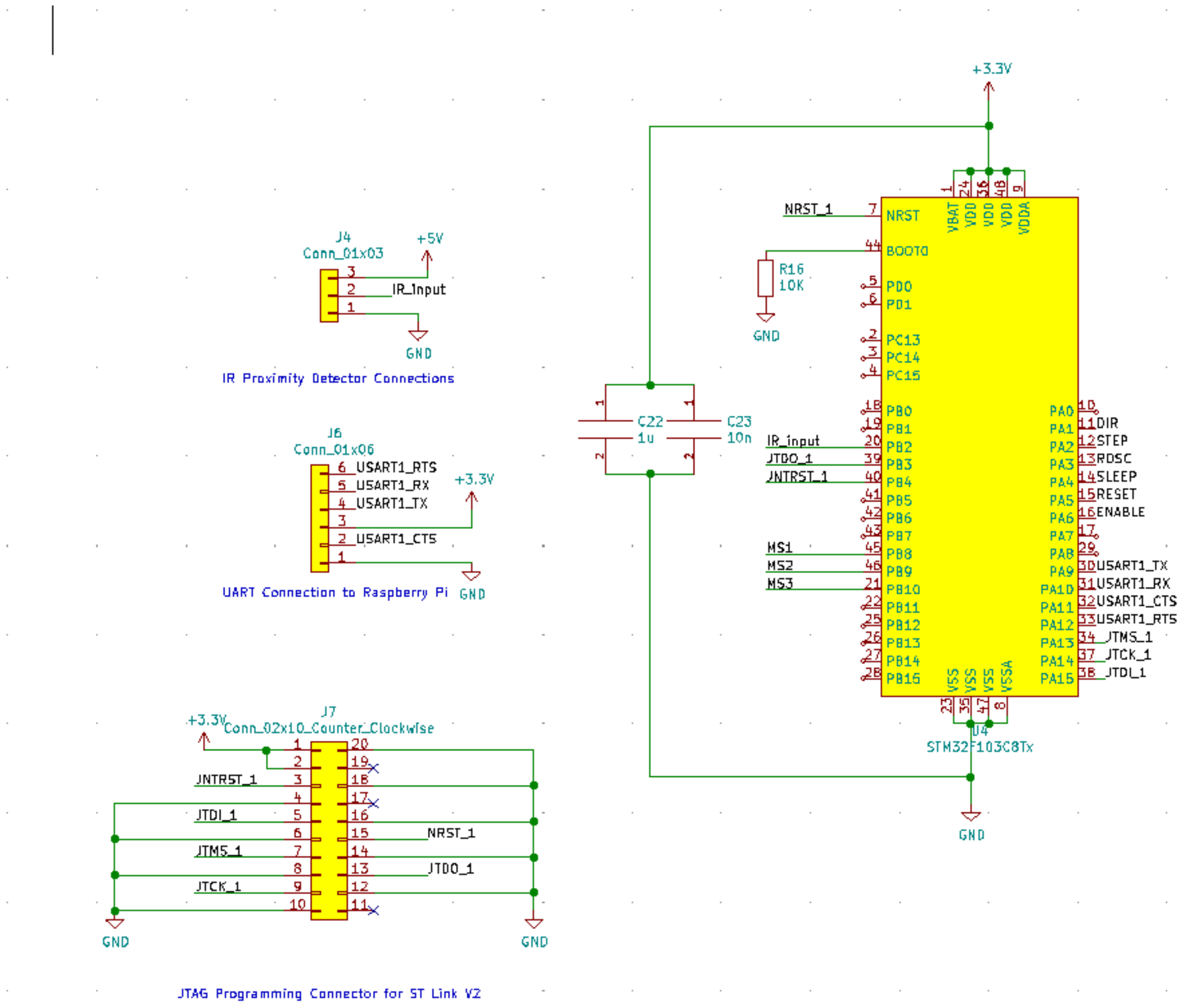
- Required Voltage levels of 12, 5, and 3.3 V
- Design utilizes a Buck Converter to go from 12 to 5 Volts
- Linear Regulator Drops 5 V down to 3.3 V
- Power source is a 12V wall converter



Power Subsystem

Requirement	Verification
• Ripple on the 5V system stays under 250 mV, as needed by the raspberry pi and can support expected current draw by the LDO and raspberry pi (1.5 A)	• Ripple can be measured with an oscilloscope, current support is available on datasheets
• Ripple on the 3.3V system is kept under 300 mV for MCU operation, current must support all 3.3V components	• Ripple can be measured with an oscilloscope, current support is available on datasheets

Control Subsystem

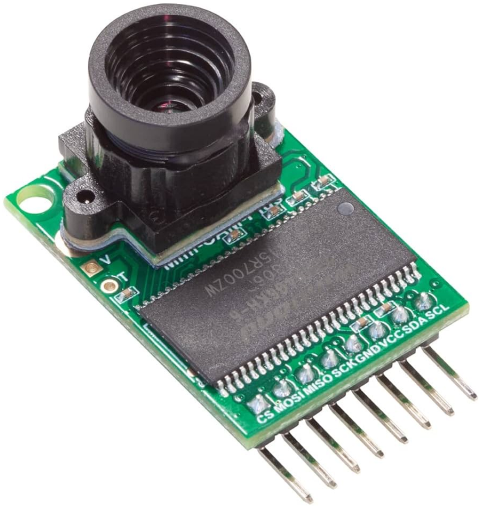


- Stepper motor moves item based on information received and returns to level position
- STM32 communicates with Raspberry Pi
- Raspberry Pi houses VGG16 model

Control Subsystem – Requirements & Verification

Requirement	Verification
•Stepper motor capable of rotating the amount required to move item and return to level position ± 1.8° (1 step)	•Test sorting motion on objects of different weights and shapes to find necessary angle difference (estimated 45°)
•MCU can communicate with other components and collect data for analysis in an organized way with use of one of several supported protocols	•MCU receives data input from camera and GPS modules with minimal losses (>98% success rate) •MCU communicates back and forth with microcomputer, can be verified through raspberry pi output
•Microcomputer interfaces with the internet and retrieves data in a reasonable time (<5 seconds)	•Run a script that times the data retrieval and outputs the time

Sensor Subsystem



- Original design includes a camera and GPS that both communicated with the microcontroller
- Final design moved the camera to the Raspberry Pi, removed the GPS, and added a proximity sensor
- Proximity sensor detects when a container is placed on the platform and signals the camera to capture an image

Sensor Subsystem – Requirements & Verification

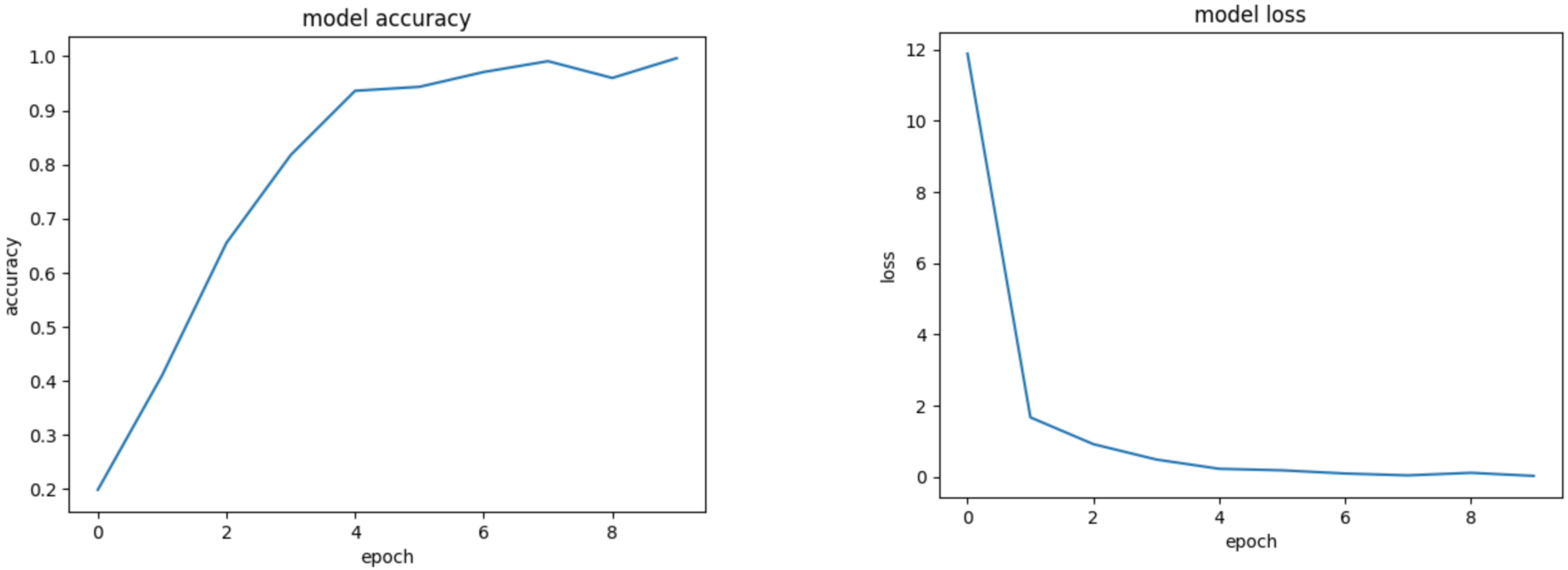
Requirement	Verification
• The image the camera produces must be at least 244 x 244 to ensure that the symbol can be read properly by the VGG model.	• Place a plastic container in front of the camera sensor • Verify that the pixel count is at least 244 x 244 pixels using OpenCV
• The GPS must provide coordinates within a 10 meter radius given that the device is in a location free of large obstacles.	• Display the GPS coordinates sent by the device on the web page (for verification purposes only) • Calculate the distance between the true location and GPS coordinates • Verify the calculated distance is within a 10 meter radius

Software Design

VGG16

- Deep convolutional neural network consisting of 16 layers – convolutional, max pooling, and dense layers
- Implemented using a PyTorch model pre-trained on ImageNet
- Finetuned model on Plastic Identification Symbol data set and our own dataset consisting of ~60 images
- Achieved 97.5% accuracy

Project Results – Model Accuracy



Model Accuracy and Losses

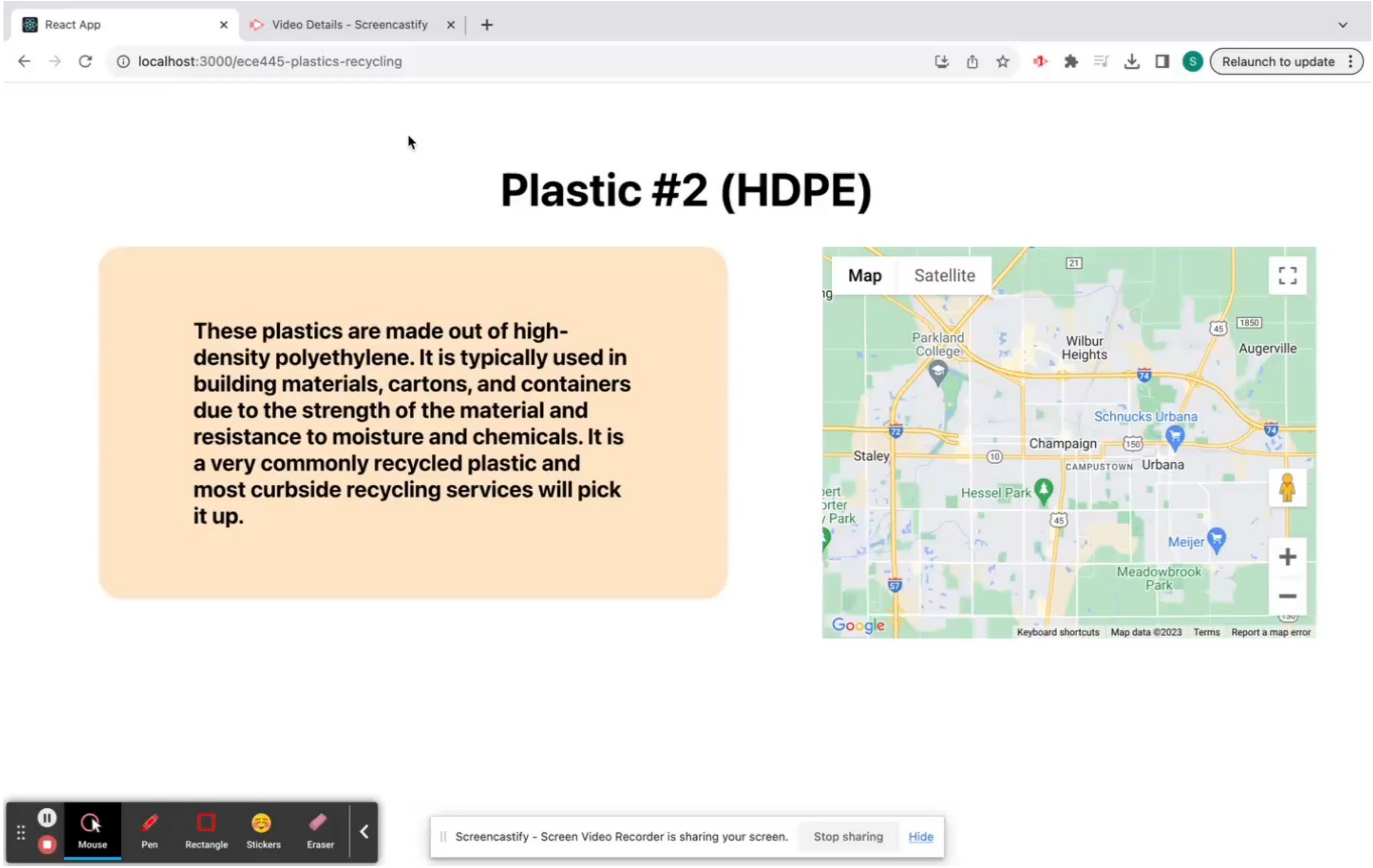
User Interface Subsystem

- Receive model classification and display information about the plastic

- Pinpoint nearby recycling locations depending on a user’s location

- Built using React, Django, Geoapify API (originally RecycleNation API), Google Maps API

User Interface



Successes and Challenges

Successes

- Gained hands on experience in machine learning, image processing, PCB design, microcontroller programming, and UI design
- Majority of the individual components in our system worked

Challenges

- Training the VGG16 model
- Broken components
- Integrating the system

Conclusion

What We Learned

- Experience with a variety of concepts
- Fail early and learn to adjust
- Continue iterating to improve design
- Technical writing and presentation skills

Conclusion

What We Would Change

- Load model onto electronics earlier
- Order parts earlier and have backups
- Build a schedule based on research and have a Plan B

Future Work

- Fully integrate the subsystems
- Streamline design by removing the Raspberry Pi
- Extend the functionality of the UI to a mobile application
- Adjust system to include other recyclables

Thank you!

Special thanks to our TA, Jeff Chang.

Contact Information:

Jennifer Chen – jc46@illinois.edu

Jason Wright – jasonlw2@illinois.edu

Smruthi Srinivasan – smruthi2@illinois.edu

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN