Automatic Spice Dispenser

GROUP 10

Motivation

- To create an efficient way to maintain spices organized and available in a home kitchen.
- Eliminating the need to manually search even an organized spice rack to get a single spice.
- Experimental new gadgets are appealing to us.

Goals and Objectives

- Creating an automatic turntable which can be rotated via voice commands and hand gestures.
- Creating a mobile app which can interface with the new device and control the majority of its functionality, chiefly spice selection.
- Creating a specialized dispensing/grinding mechanism for the serving of spices selected by the user.

Specifications

- Chassis must respond to hand gestures in under 5 seconds.
- Shall dispense a tablespoon of spices in under 15 seconds.
- Chassis shall hold 6 spices at a time.
- Chassis shall weigh under 25 pounds.
- Chassis rotation speed must be at 8 seconds per rotation.
- User will be able to request spices though a mobile application.
- System must detect hand gestures within 2 meters of camera.

Three Mechanical Subsystems

Rotational Subsystem

•This system will rotate the spice rack to the desired spice for the user.

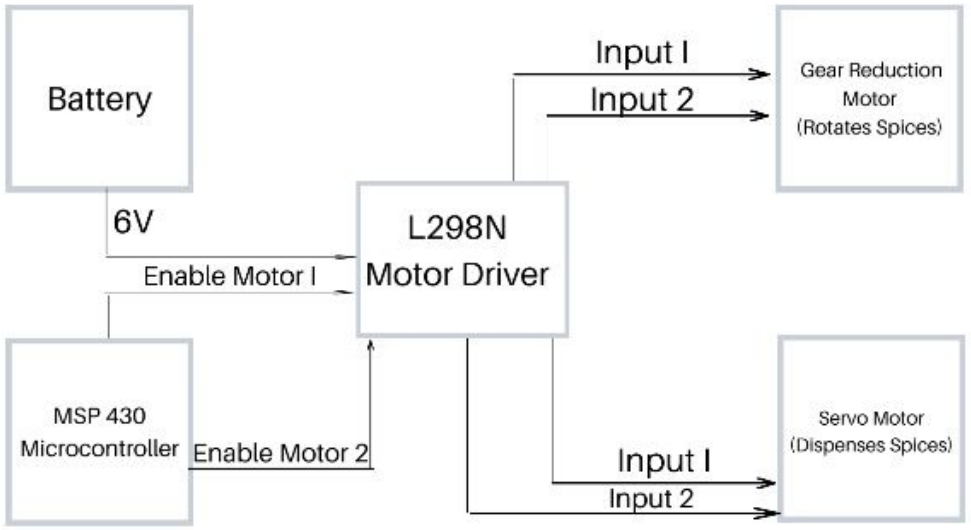
Dispensing Subsystem

•This system will dispense the amount of spice the user inputs into the App or how much they manually dispense by using a switch.

Crushing Subsystem

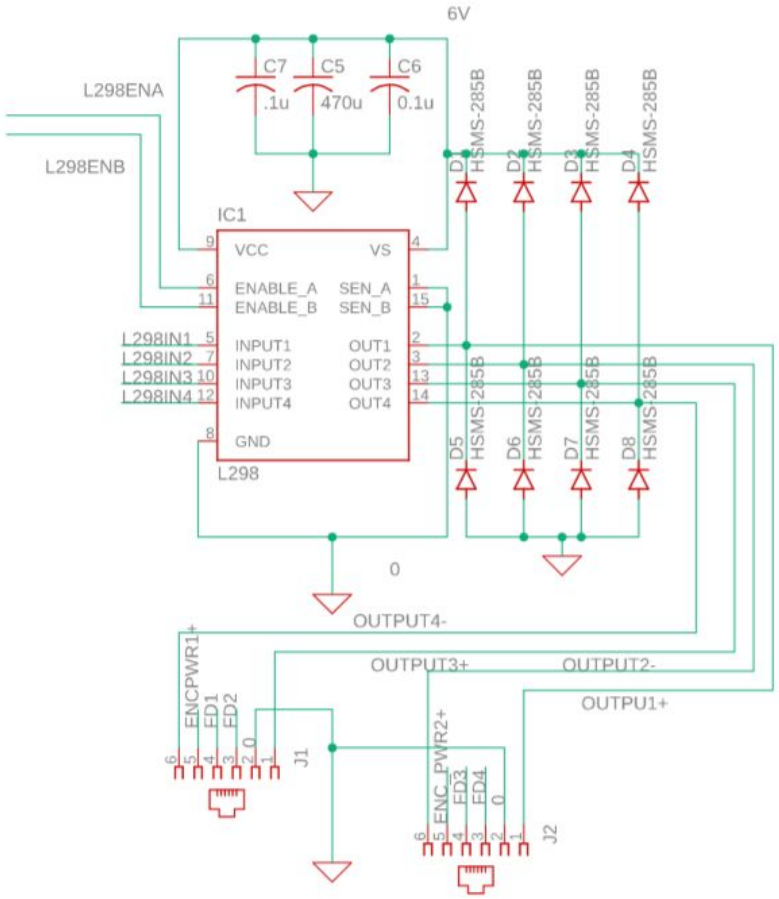
•This system will crush fresh spices the amount of fresh spices for the user through the app or by the manual switch.

Rotational Subsystem

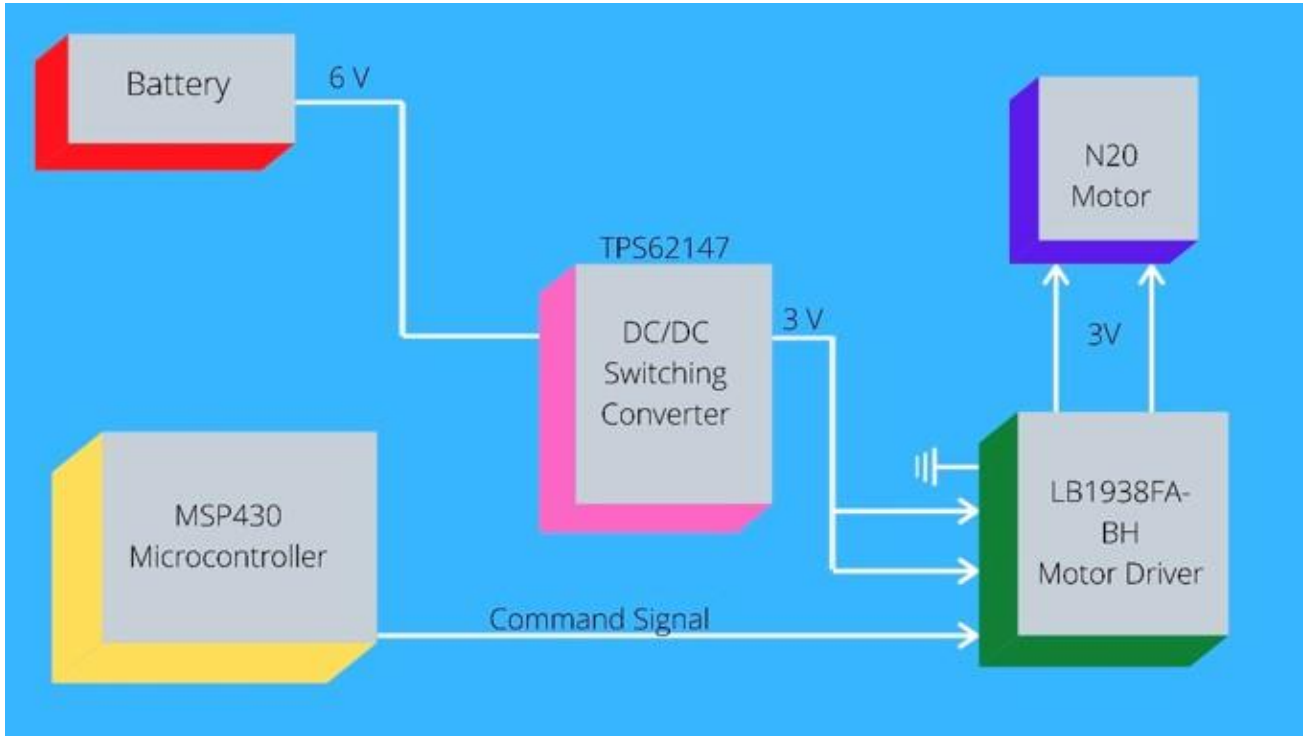


- The 6V gear reduction motor will spin at 12 rpms, and consume 3 Watts of power (0.5A) during use.
- L298N motor driver allows for a 4A output, which will give us plenty of room for adjustments. Since the motors may consume more power to rotate the spice rack.
- The Hall-Effect Encoder on the motors will communicate with the MSP430 to verify the position of the spice rack.

Rotational Schematic



Dispensing Subsystem



- The dispensing motor will be a speed reduction motor that will spin at 15 rpms and only consume 0.15 Watts (50mA). The low power consumption is very beneficial because dispensing will be the most used feature in our design.
- Like the Crushing subsystem, the motor driver will allow the MSP430 to turn the motor on and command the motor to spin in different directions.

External Electronics

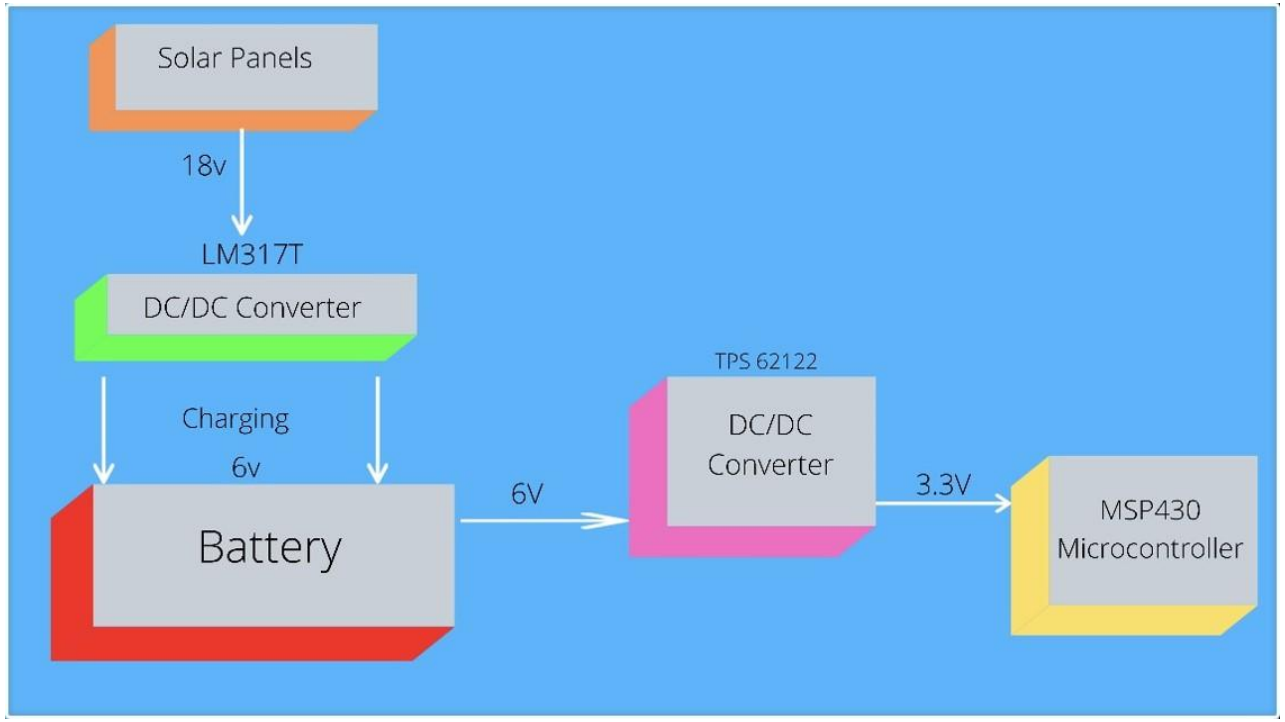
PV Panel

- The PV panel will be used to charge the 6V battery when the user has no use for the spice rack.

Position LEDs

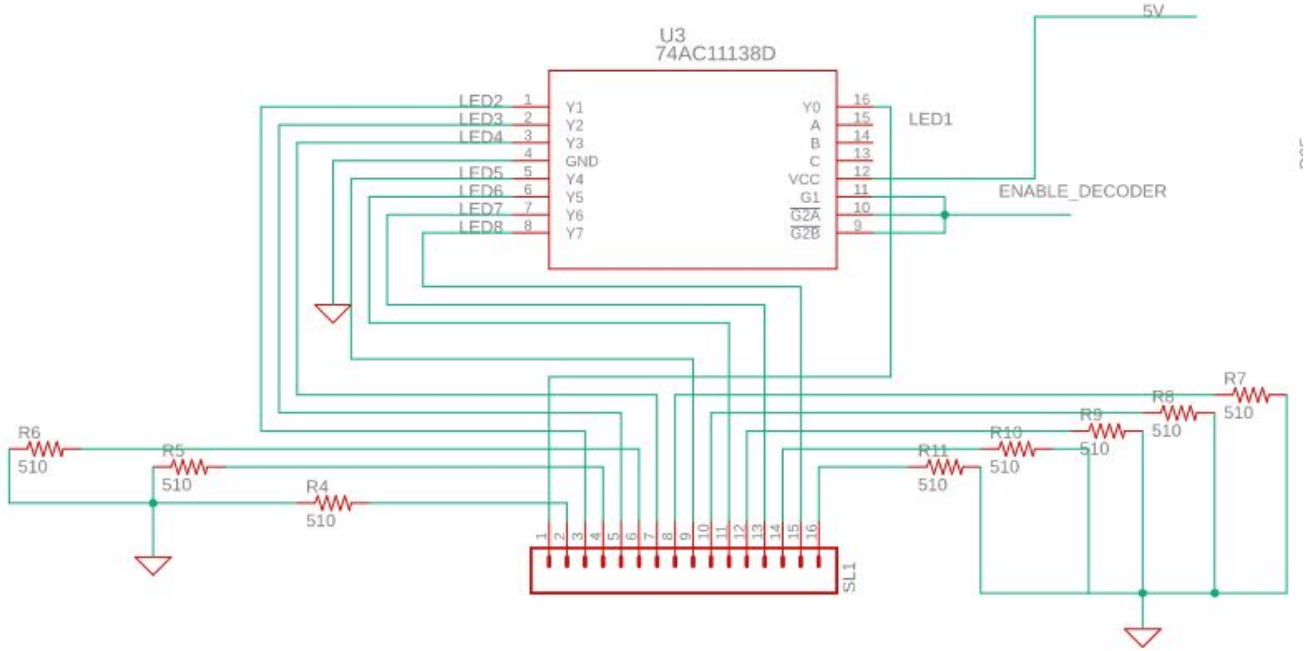
- These will indicate which spice the user has selected and where it is on the spice rack.

PV Panel Charging



- The solar panel will output 4.2 Watts at 18V. This will give us a recharge time of 15 hours. This may seem like a lot of time, but the user will only need the spice rack around 3 times a day. So there is no need to fully recharge the battery.
- The solar panel is 200mm by 130 mm (about 8in. by 5in.) and will fit perfectly on the top of the spice rack.

Position LEDs



- We will be using a decoder to get a 3 bit input from the MSP430 and an 8 bit output for our 8 different LED positions.
- The connector will have 16 pins because the LEDs will have a power and return, which will need a 510 Ohm resistor to function properly.

CPU Controls

- All instructions for the spice dispenser will go through an MSP430 at one point or another.
- MSP430 CPU will interface with a Raspberry Pi, an ESP-32, and motor drivers for output.
- This will allow all components of the device to connect to the MSP430 through their associated peripherals

Raspberry Pi - MSP430

- The Raspberry Pi will be responsible for the gesture control functionality of the device.
- A Machine Learning Model will be pretrained inside of the Raspberry Pi, which will process data from a camera and send appropriate commands to the MSP430 via I2C Communication.
- 12 Bytes of Data can be transmitted at a time, but through the use of numerical codes, that should allow us to send a large number of instructions (~2^{12} unique Instructions)

ESP-32 – MSP430

- Similar to the Rapsberry Pi, the ESP-32 will communicate with the MSP430 via I2C communication, also sending codified instructions.
- This is possible since the MSP430 has two sets of I2C pins which allows both devices to send data simultaneously (although their instructions will have to be differentiated).
- The ESP-32 will be responsible to translate information received from the Mobile App to MSP430 instructions.

Raspberry Pi Computer Vision Module

- The Raspberry Pi will be responsible for processing a video input stream
- Using OpenCV, it will filter out human skin from an image, and then process the result using a Keras ML model to see if a hand gesture is detected.
- If one gesture is a match, then the Raspberry Pi will trigger the appropriate command via I2C.

ML Model

- The Machine Learning Model is trained on a set of preprocessed images.
- This pre-processing is done by using an HSL color filter to attempt to remove anything that is not within the range of human skin colors.
- Then, contours are drawn on the remains, with the largest found contour set being selected. All remaining contour sets are masked and set to black. This leaves only the hand closest to the camera in the image.
- This same preprocessing is done on the camera recorded frames, allowing for the same conditions in both training and actual use of the ML Model.

Mobile Application

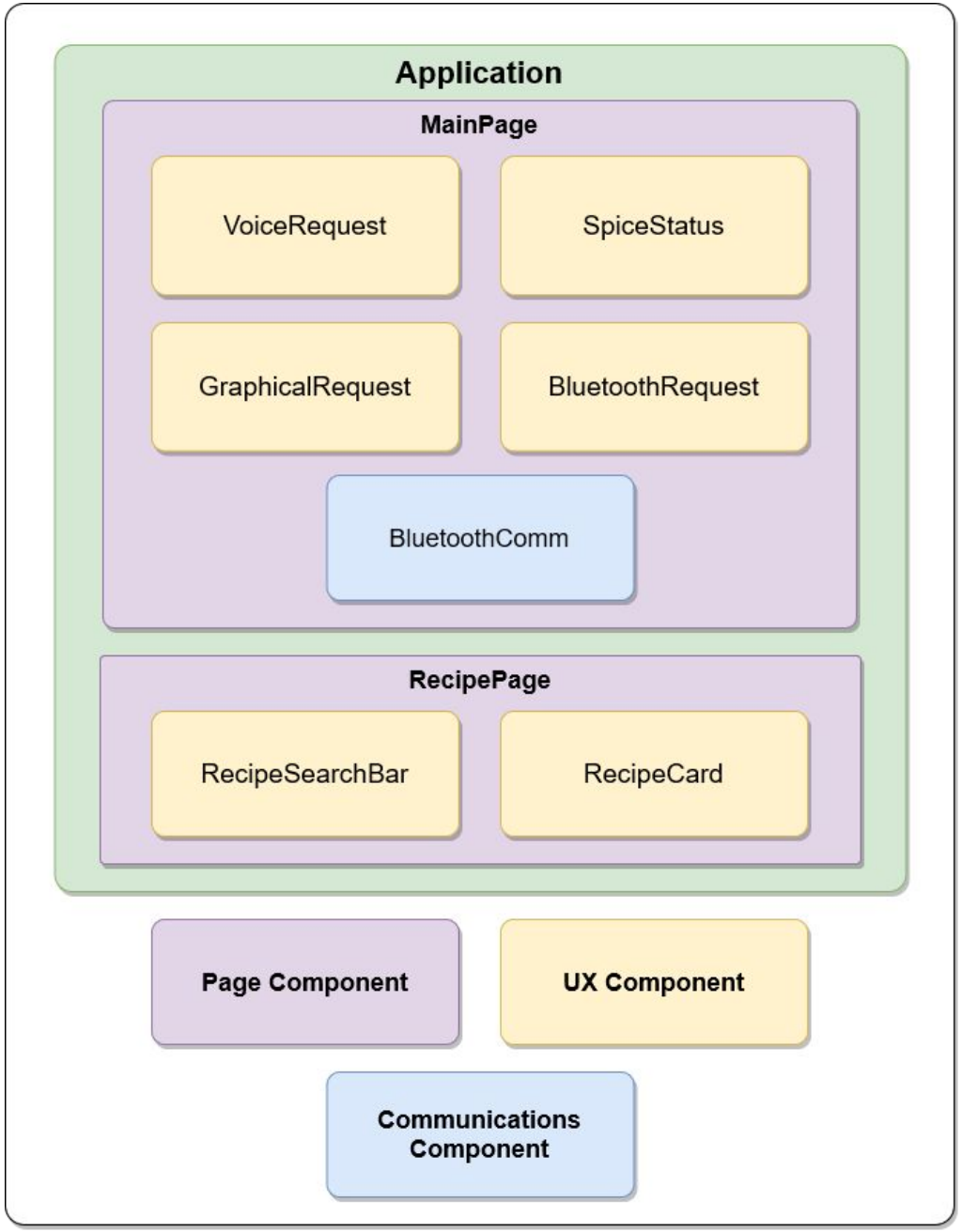
- We will create a mobile application to give the user more ways to interact with the spice dispenser.
- The user will be able to request a specific spice stored in the dispenser in a specified amount through UI options, voice request or through recipe lookup.
- The user will also be able to change the names of spice within each capsule in the app in the event that spices are switched out in the dispenser.

React Native

- The application will be built using React Native, a JavaScript library for creating cross-platform mobile applications.
- Using this library will allow us to quickly create a single code base for both Android and iOS.
- This library is also allows for integrating different packages to extend an application's functionality. We will be using some of these packages for this project.

Application Design

- The page components will server to house and organize the buttons and other interactive (UX) components for the user.
- UX components will capture and process the input from the user.
- The communication component will send and receive information from the dispenser based on user input.



Recipe Lookup

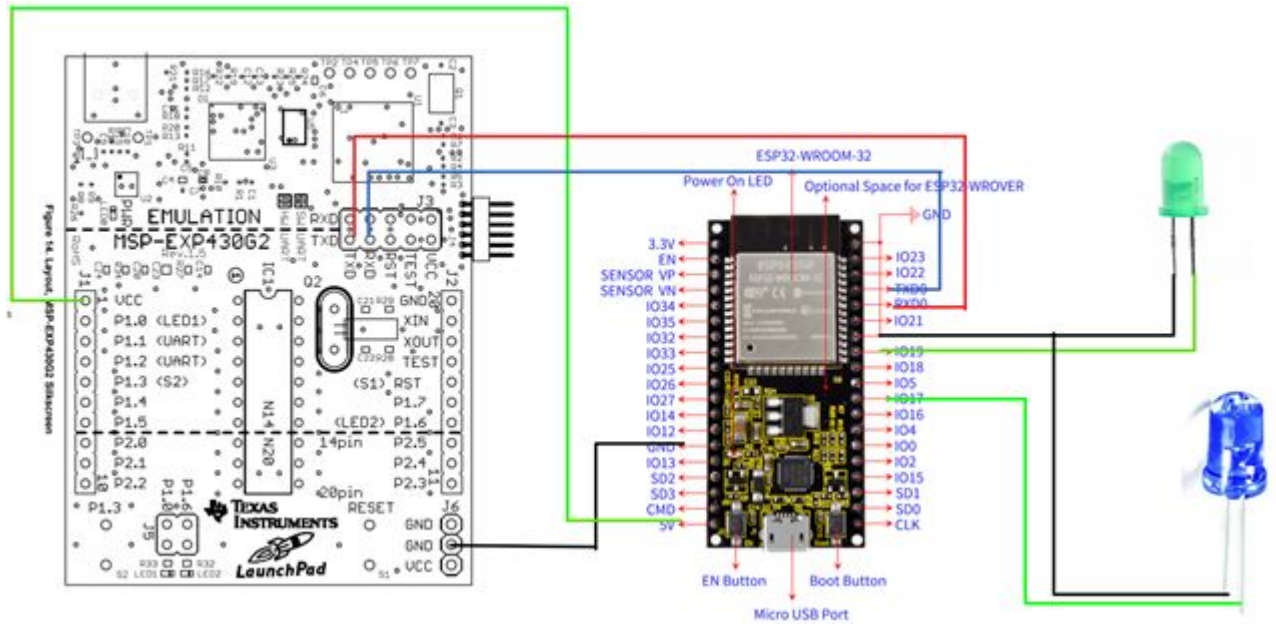
- We will implement a convenient way for users to request spices from an online recipe.
- We will use the Edamam's Recipe Search API, which allows different applications access to information of millions of recipes online.
- The user will input a search query for recipes and some filter options. This information will then be used to send an API request for a list of matching recipes.
- We will process the results to highlight spices that match those which are stored in the dispenser.

Bluetooth

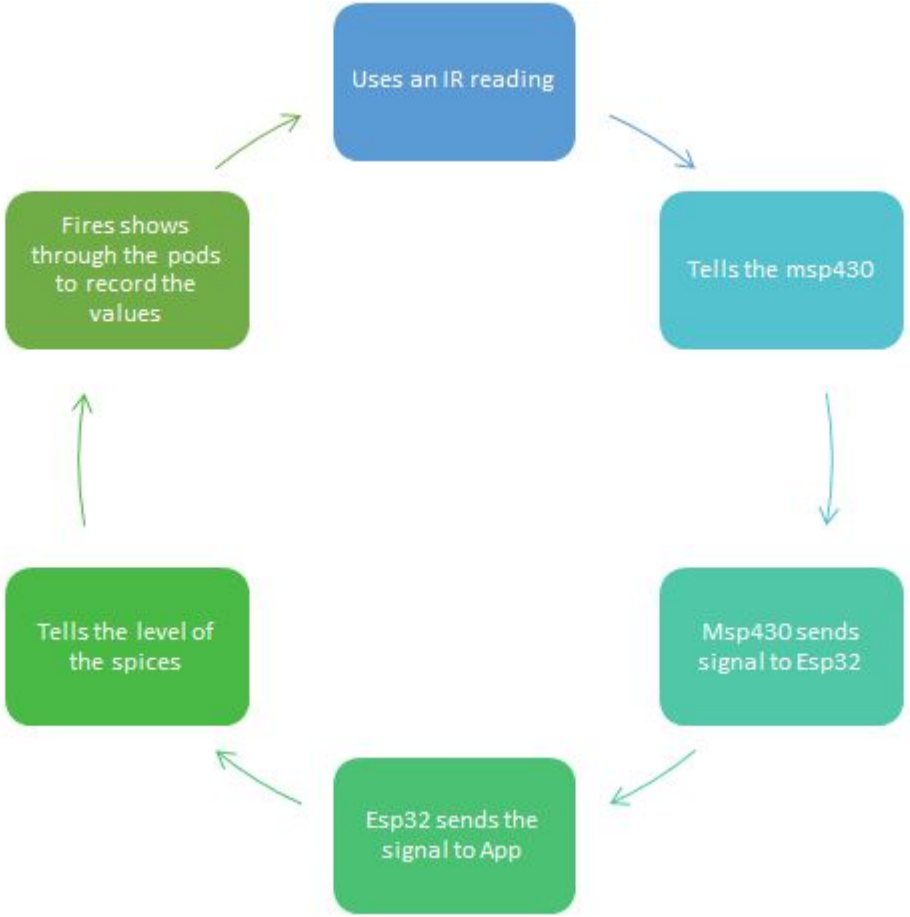
- ESP-32 is the device used to receive Bluetooth messages.
- Using the ESP-32 to send messages will introduce how the user can go to the spice requested.
- Connecting to the app and how does the connection between the app and the device get implemented.

LEDs

- The LEDs will be used to debug, visually.
- The purpose of the LED is to indicate what is going on in the device.



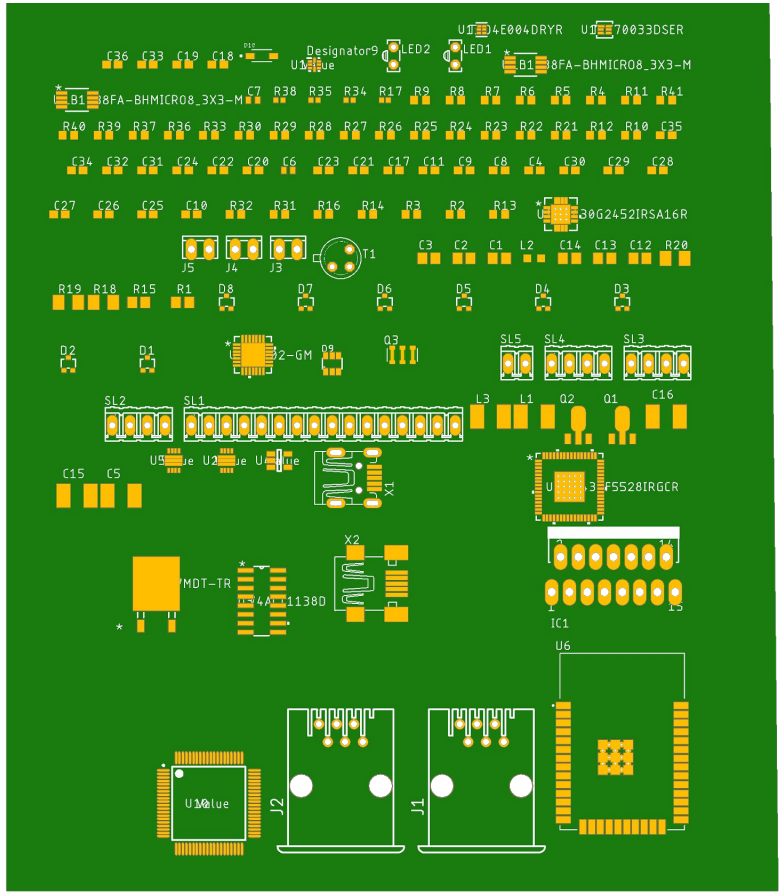
Light Diode

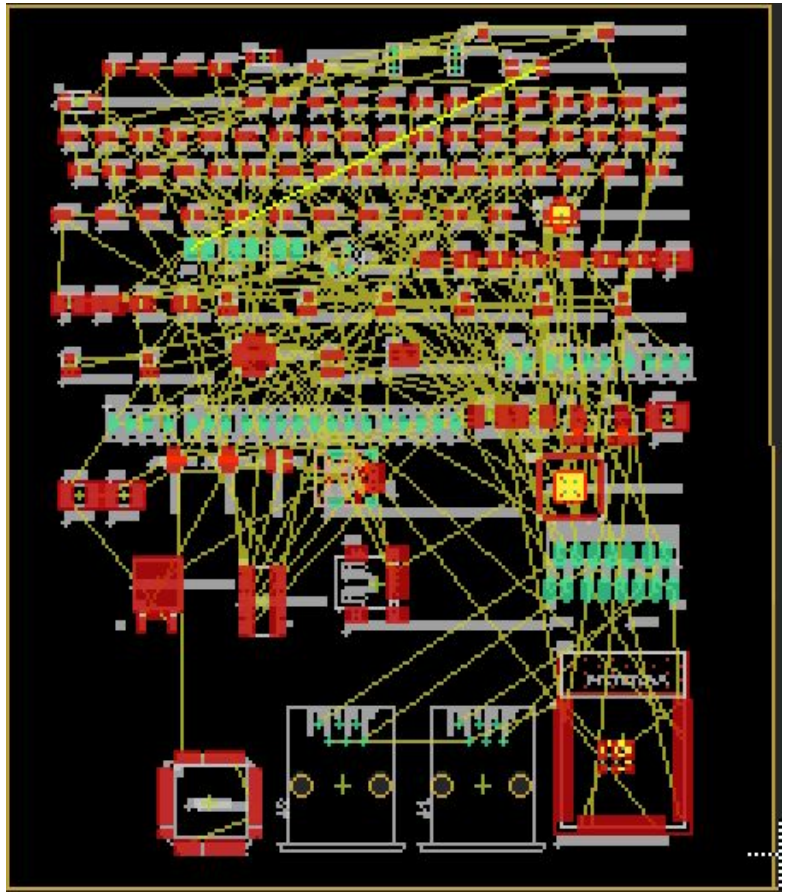

Uses an IR reading
Tells the msp430
Msp430 sends signal to Esp32
Esp32 sends the signal to App
Tells the level of the spices
Fires shows through the pods to record the values

Measuring Cups

- This is to ensure that the appropriate amount of spice is dispensed.
- With the use of light sensors this is possible.
- This will be instructed through the app.

PCB





Questions?