Trail Mix Dispenser

By: Andrew Ma, Mat Jacob,
Kanav Kariya
TA: Dongwei Shi

Outline

- Introduction
- Design and Requirements
- Successes and Challenges
- Future Work

Introduction

Problem Statement

- When you’re on the go, it’s difficult to meet your nutritional requirements
- Trail mix can be portable, customizable and nutrient dense
  - It is time-consuming but necessary to weigh out each ingredient manually on a regular basis

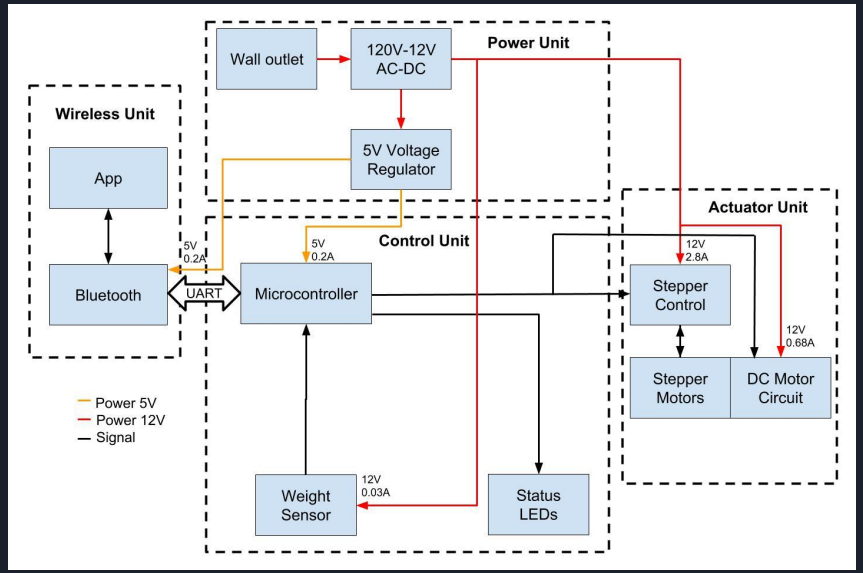
Objective

- Design a fully automated and Bluetooth enabled Trail Mix Dispenser that is capable of
  - receiving commands and recipes from an Android device
  - autonomously dispensing four unique ingredients
  - weighing each individual ingredient out as specified by the recipe
  - mixing the final trail mix together

Design and Requirements

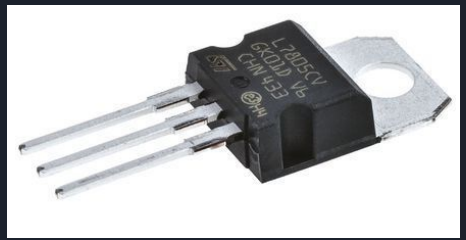
System Overview

- Power Unit
- Actuator Unit
- Wireless Unit
- Control Unit



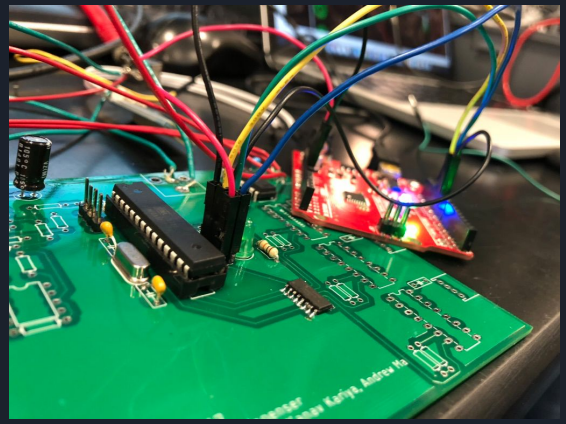
Power Unit

- 12V power supply must supply $V_{out} = 12 \pm 0.5V$
  - V_{out} measured at 12.20V
  - Steppers, DC Motor, Load cell and 5V Regulator
- 5V voltage regulator must supply $V_{out} = 5 \pm 0.3V$
  - V_{out} measured at 5.04V
  - MCU, the BT Unit and decoder



Control Unit – Microcontroller

- Atmega328P
- Microcontroller from Arduino Uno
  - Easy to program
  - Many libraries available
- 10-bit ADC
- Pins
  - 13 GPIO (3 PWM)
  - 6 ADC



Control Unit – Microcontroller

- Exponential Moving Average (EMA) for more stable weight readings (microsmooth)
- Rest of code was straightforward using data from load cell regression

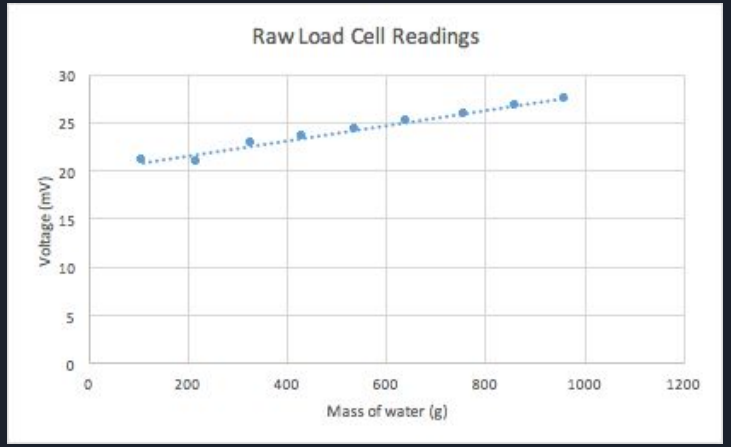
```
void dispense(int stepper){
  digitalWrite(DECODER_ENABLE, LOW);
  digitalWrite(DECODER_SELECT[0], stepper & 0x01);
  digitalWrite(DECODER_SELECT[1], stepper & 0x02);

  tone(DECODER_ENABLE, 31, 500);
}

float readWeight(){
  int weightReading = analogRead(LOAD_CELL);
  int averageWeight = ema_filter(weightReading, averagePtr);
  float weight = ((float)averageWeight-baseWeightReading) / 0.0987;
  return weight;
}

void dispenseWeight(int stepper, float dispenseWeight){
  float weight = readWeight();
  Serial.println("Dispensing " + String(dispenseWeight)
                 + " grams from container " + String(stepper));
  float initialWeight = weight;
  while(weight+0.5 < initialWeight + dispenseWeight){
    Serial.println("start loop" + String(weight));
    dispense(stepper);
    for(int i = 0; i < 5000; i++){
      readWeight();
      delay(1);
    }
    weight = readWeight();
    Serial.println(weight);
  }
}
```

Control Unit – Weight Sensor

- Load Cell with 3kg capacity, 12V excitation voltage, 2mV/V rated output
  - 0.008mV/g
  - Zero balance of 0V
- Support 1kg of trail mix

Control Unit – Weight Sensor

- Verified using known masses and Fluke DMM
- 2.5% error from theoretical mV/g
- Zero balance had shifted to 19mV



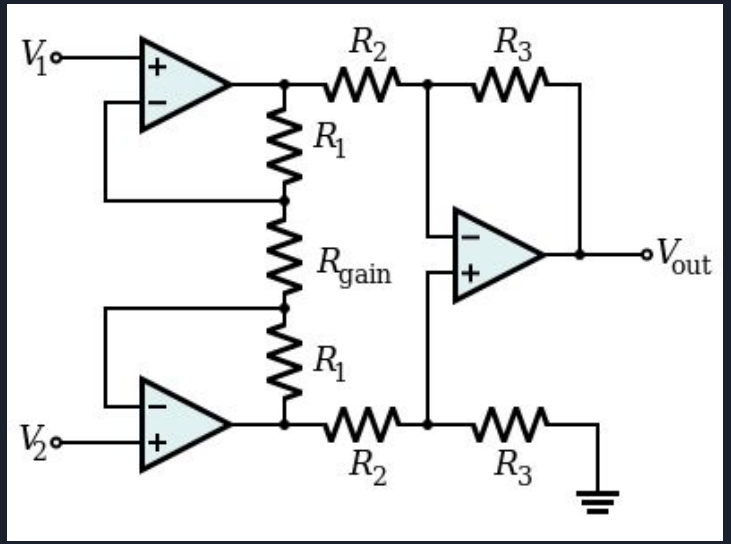
$y = 0.0078x + 19.999$

$R^2 = 0.9841$

Control Unit – Weight Sensor

- Instrumentation Amplifier
  - Accurately amplifies the raw load cell reading to a readable level for the Atmega328p ADC

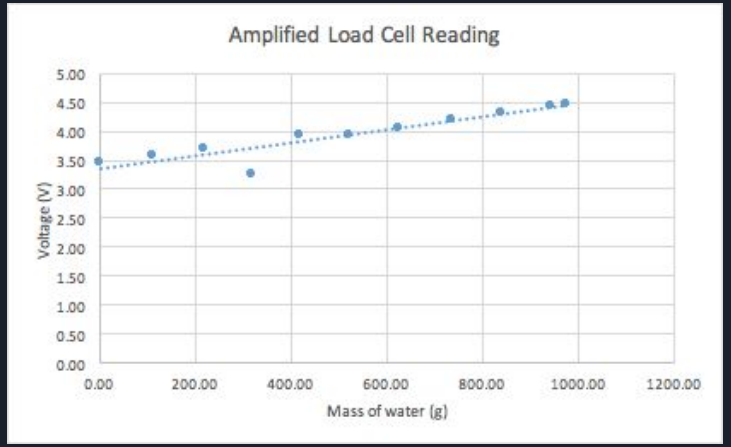
$$\frac{V_{\text{out}}}{V_2 - V_1} = \left(1 + \frac{2R_1}{R_{\text{gain}}}\right)\frac{R_3}{R_2}$$



Source: https://en.wikipedia.org/wiki/Instrumentation_amplifier

Control Unit – Weight Sensor

- Theoretical Gain of 160, Measured Gain of 1.1/0.0078 = 140
  - 13% error
  - constant across all readings
- Shifted zero balance to 24mV

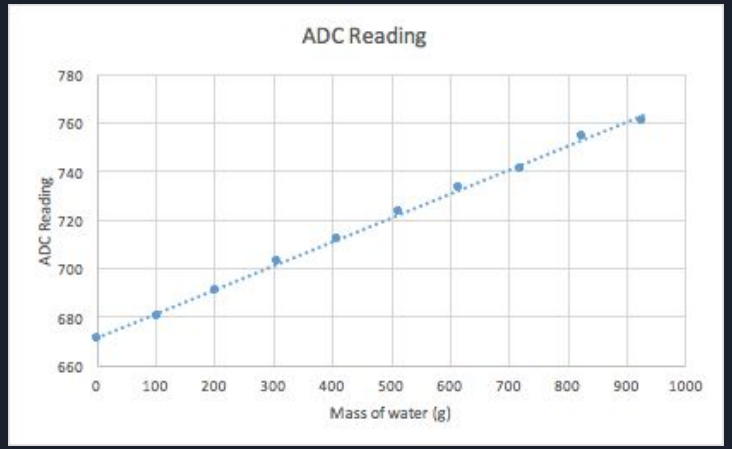


$y = 0.0011x + 3.3623$

$R^2 = 0.84386$

Control Unit – Weight Sensor

- Shifted zero balance to 58.5mV
  - added voltage divider circuit for overvoltage protection
- 10.13 g/count resolution
  - Noise addressed using EMA



$y = 0.0987x + 671.31$

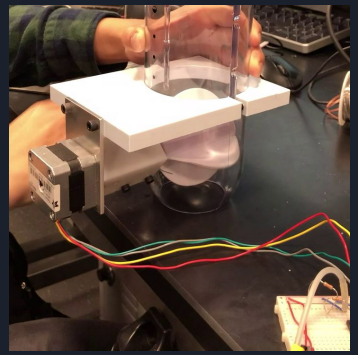
$R^2 = 0.99852$

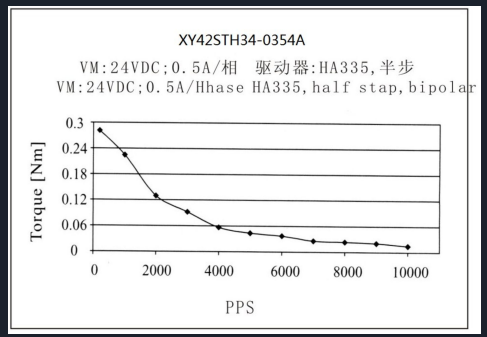
Actuator Unit – DC Motor

- Intended to mix ingredients
  - Not physically implemented
- Controlled using MOSFET
- Diode to prevent back-EMF
- Functioned electrically
  - Spun after dispensing

Actuator Unit – Stepper Motor

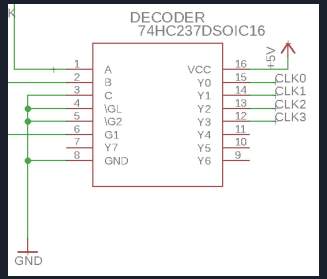
- Rated with 12V, powered with 16V
  - Drew less than 0.5A at all times
- Torque Requirement was lacking
  - Insufficient Torque
  - Should have used a torque wrench to set requirement

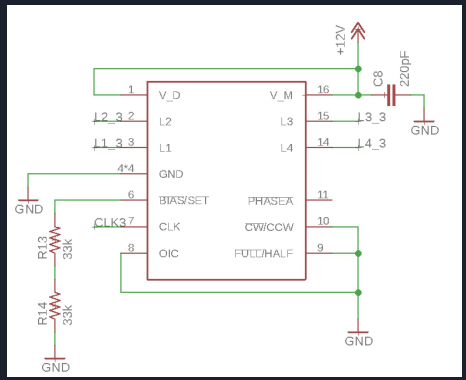




Actuator Unit – Stepper Control

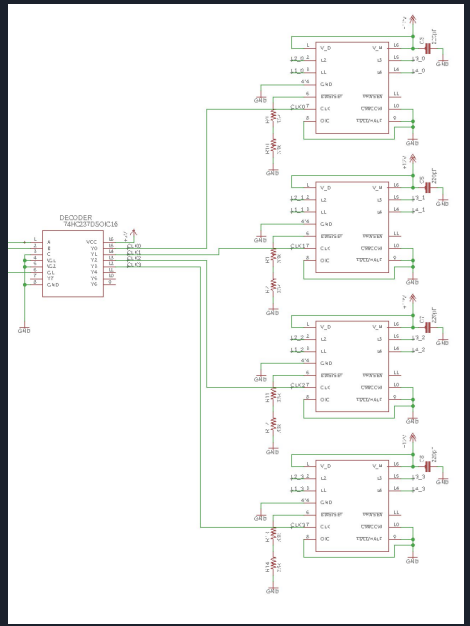
- MC3479P
  - Rated for 350 mA/phase
  - Rising edge on CLK input triggers a full step
- Using a decoder, we can use multiple drivers with few extra pins





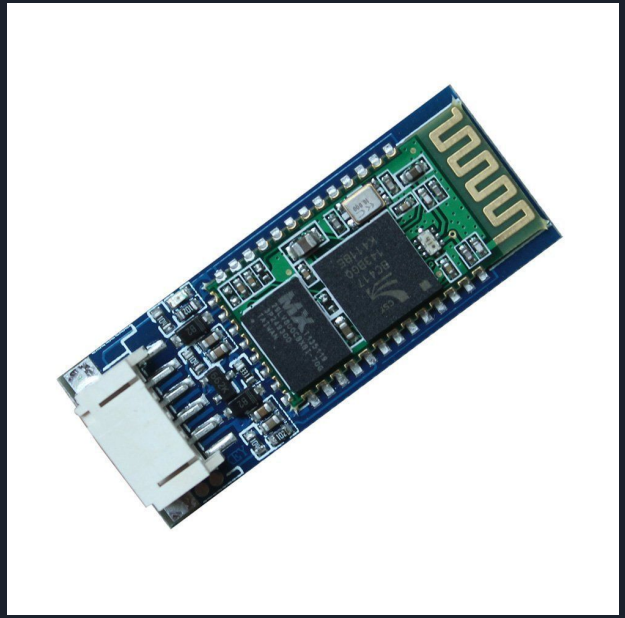
Actuator Unit – Stepper Control

- 3 to 8 decoder
- Controls 4 steppers with 3 inputs
- AB inputs specify motor
- Can use enable input to step at 20Hz to 1kHz
- Outputs connect to CLK of each MC3479P



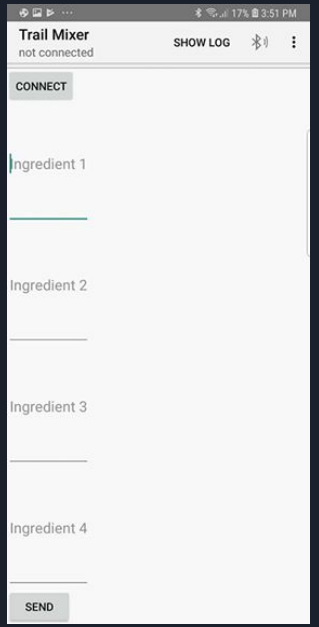
Wireless Unit – Bluetooth Module

- HC-05 Module
- UART compatible
  - No library needed
- Range greater than 5 meters
- Operates at 3.3V



Wireless Unit – Android App

- Android App sends “recipes” to HC-05 via Bluetooth
- Optional debug information can also be displayed (using Show Log)

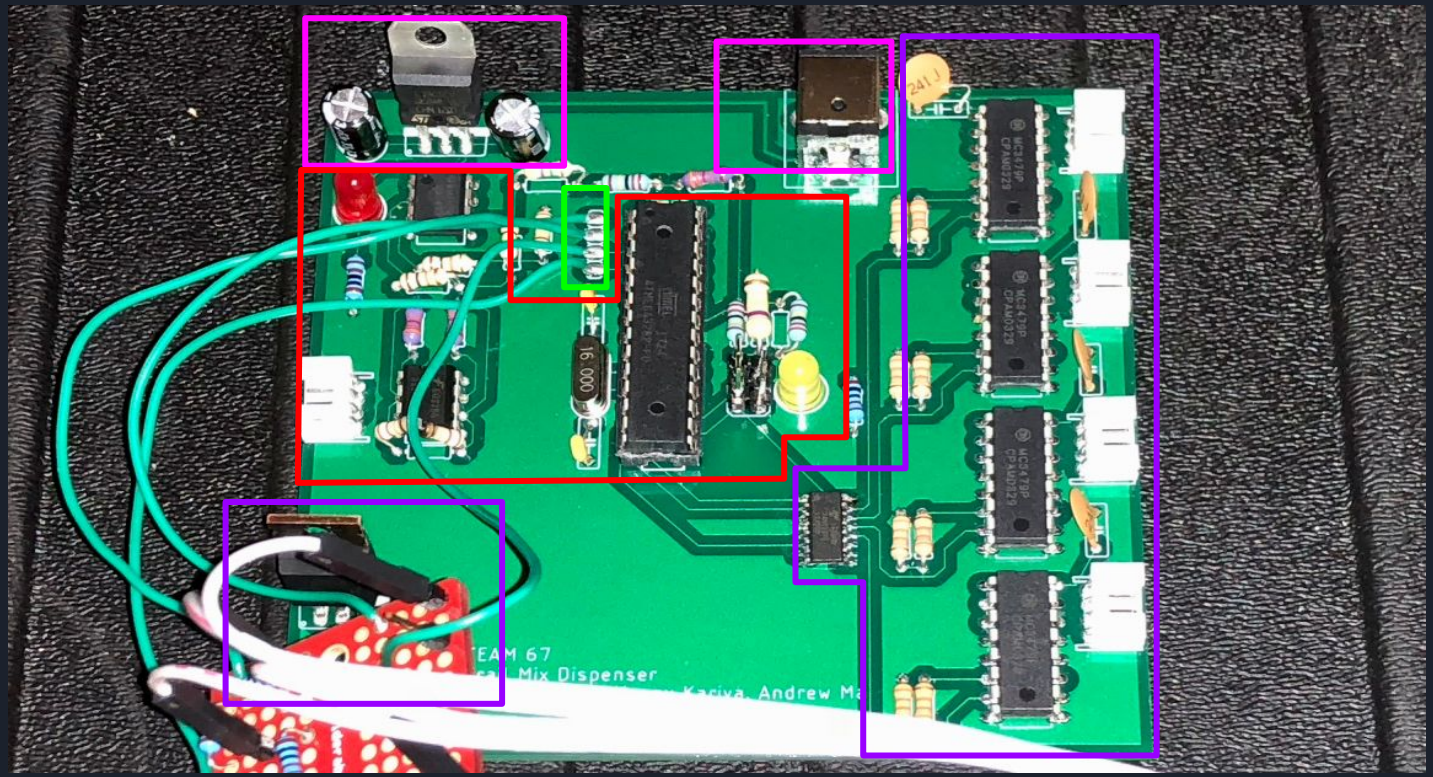


Successes and Challenges

Functionality Overview

- Electrical parts of project functioned fully!
- Stepper motors lacked enough torque to consistently dispense without stalling
- DC Motor was electrically functional, but not mechanically implemented

Functionality Overview



Roadblocks

- Load Cell offset
  - Gain adjustment to prevent saturation
- Electromagnetic and radio frequency interference
  - Insulated wire in aluminum to provide shielding
- Torque of stepper motors was limited
  - Operating at 31 steps per second still stalled for certain ingredients
  - Back EMF from stalled motors made weight readings inaccurate

Future Work

The Road Ahead – Short Term

- Stepper motors with higher torque to prevent stalling
  - More reliable dispensing
- Diode to prevent back EMF from steppers
  - More stable weight readings
- New load cell, calibrated to a 0mV reading with no mass
  - Increased resolution in weight readings
- Alter the bracket to have a slot for the PCB and power unit
- Properly shield wires from electromagnetic interference

The Road Ahead – Long Term

- Dispense ingredients into a bowl accurately without spilling
- Redesign the App to provide usage statistics and a cleaner UI
- Have a load sensor in each ingredient dispenser to enable the user to know when they're running low
    - Integrate with Amazon API to automatically order more
- Build a better bracket to make it easier to place in a user's apartment that has minimal space available

Questions?