



# Autonomous Lawn Care Unit

sddec18-22

Sam Tinklenberg, Andi Li, Bryton Hayes, Grant  
Duncan, Joel Seaser



# Problem Statement

The problem we intend to solve concerns the time and financial commitment required to upkeep a well-groomed lawn. There is a long list of reasons a certain individual may not be able to mow their lawn, ranging from lack of time to physical limitations. Someone who falls into this category does not have many options to get the job done, without hiring expensive, third-party help.



# Requirements

## Functional

- Algorithm to efficiently mow entirety of area given a mapped perimeter and detect safety concerns
- Object detection and avoidance
- Mobility through standard lawns
- Power efficiency

## Non-Functional

- Lightweight enough to be easily moved by hand
- 3.5 inch maximum grass height
- Bump sensors to prevent unwanted mowing of objects
- The motors will shut off if the mower is lifted off the ground



# Market Research

- Cost
- Cutting area in one charge
- Cumulative Efficiency
- Cutting Width

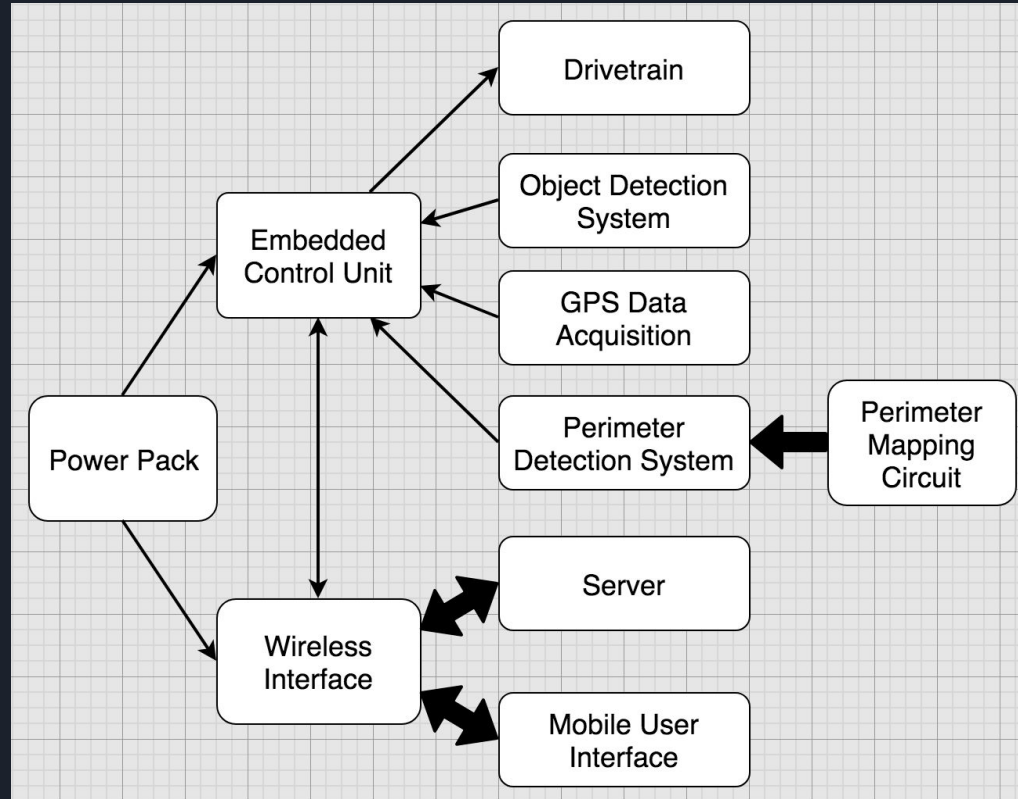




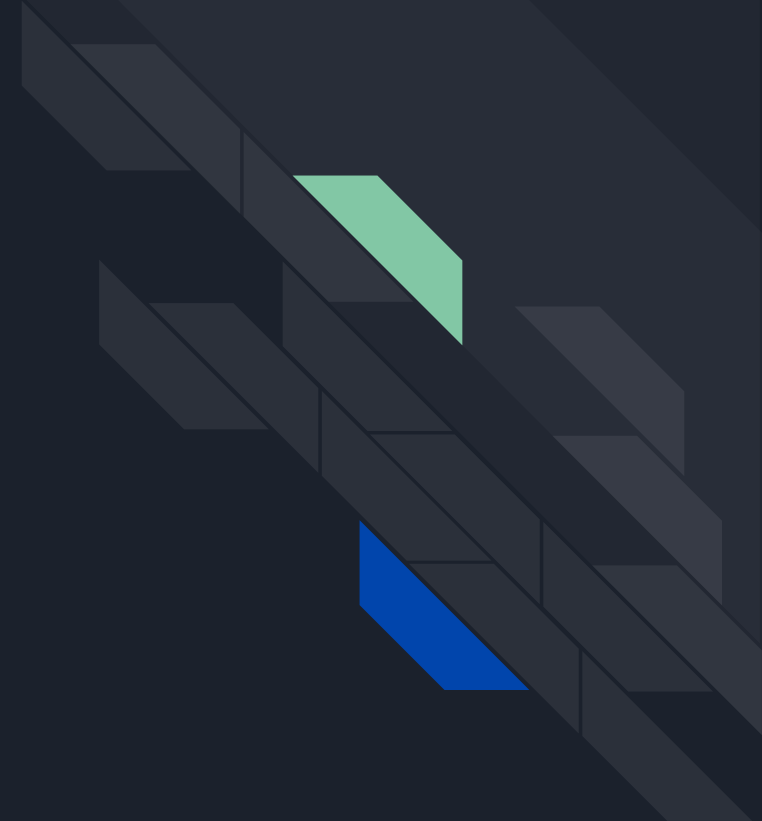
# Target Numbers

- <\$1400
- 2500 sq feet on one charge
- Operating time of 60 minutes (1 full battery charge)
- Cutting height of 1.5 - 2 inches
- 14 inch mowing width
- 15% max incline

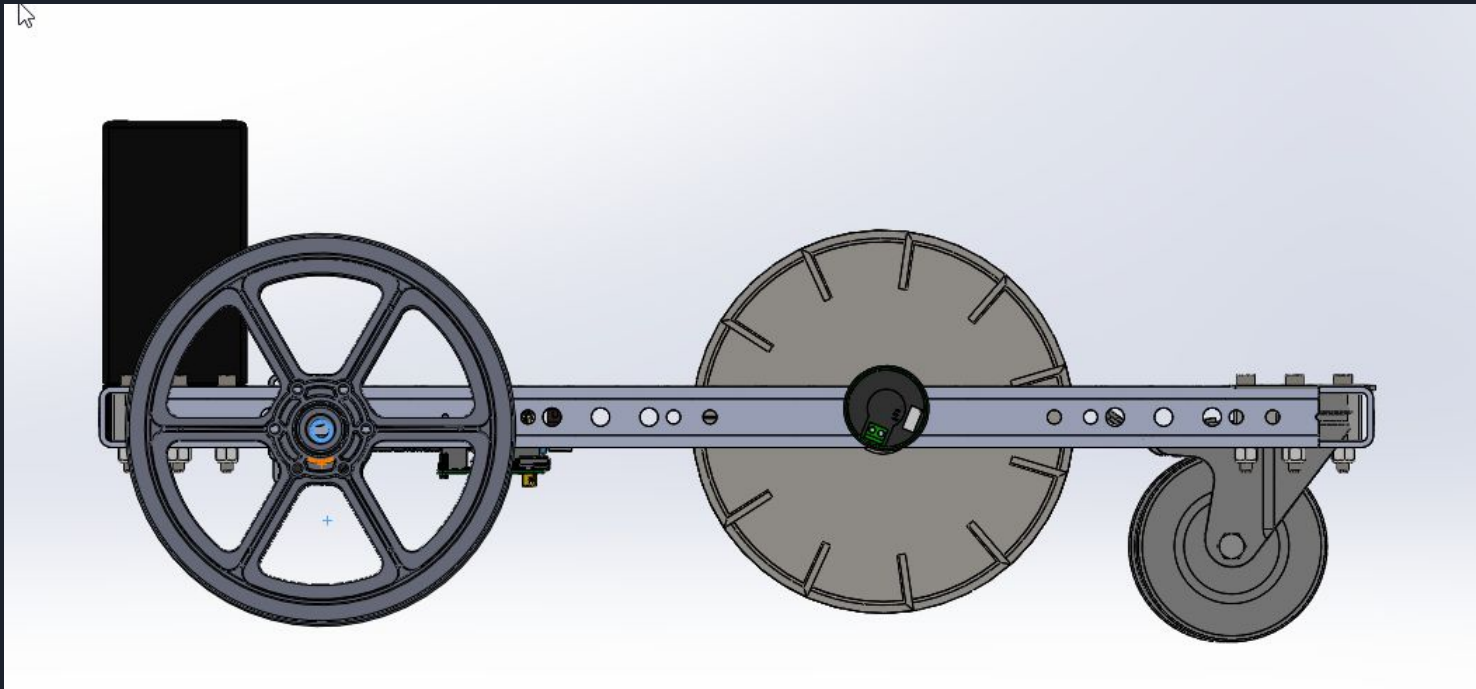
# Conceptual Diagram



# Chassis, Blade, and Drivetrain

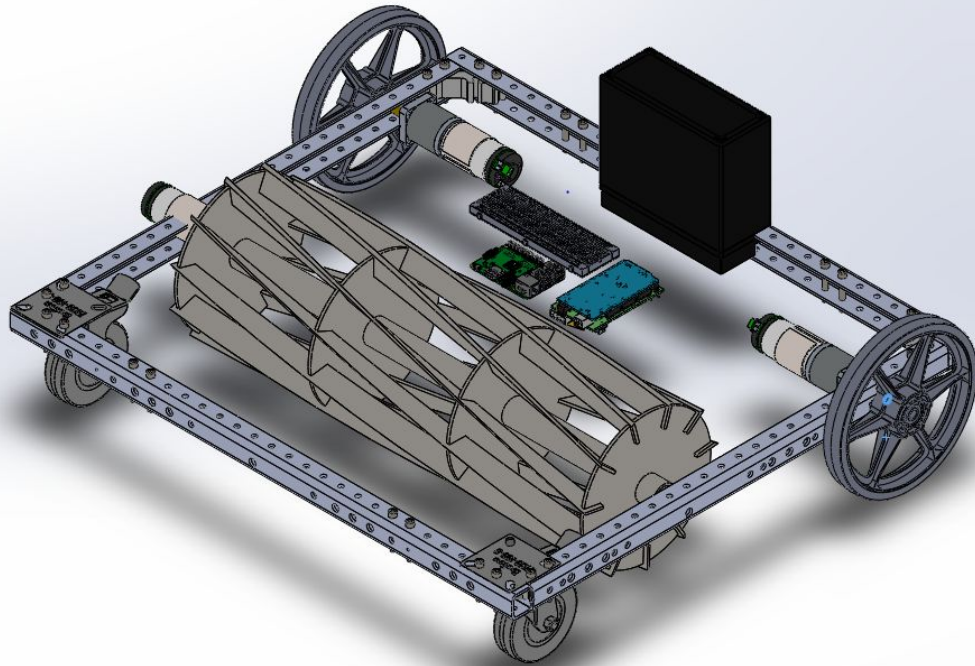


# 3D Model Side View





# 3D Model Top View



# Chassis

- C-channel chassis
- Lightweight
- Rigid
- Modular



# Blade

- Reel Blade
- Safety
  - Lower RPM
  - Doesn't "throw"
- Clean Cut
- Wide Cutting Radius
- Direct Drive
- Controlled by 3rd motor



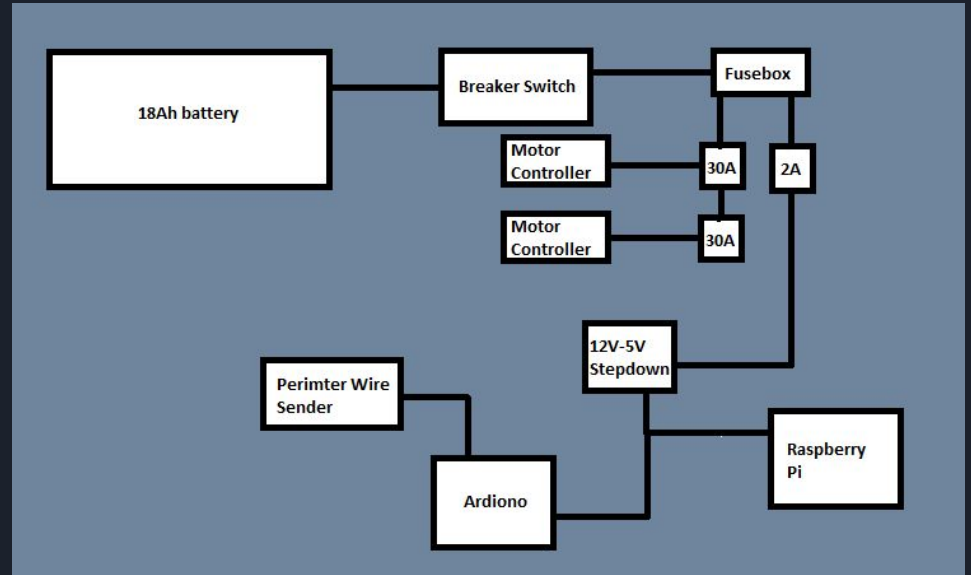
# Drivetrain

- DC motors
- Planetary Gearboxes
- 8-inch Plaction Tread  
Rubber Drive Wheels in the  
front
- 4-inch Caster Wheels in the  
rear

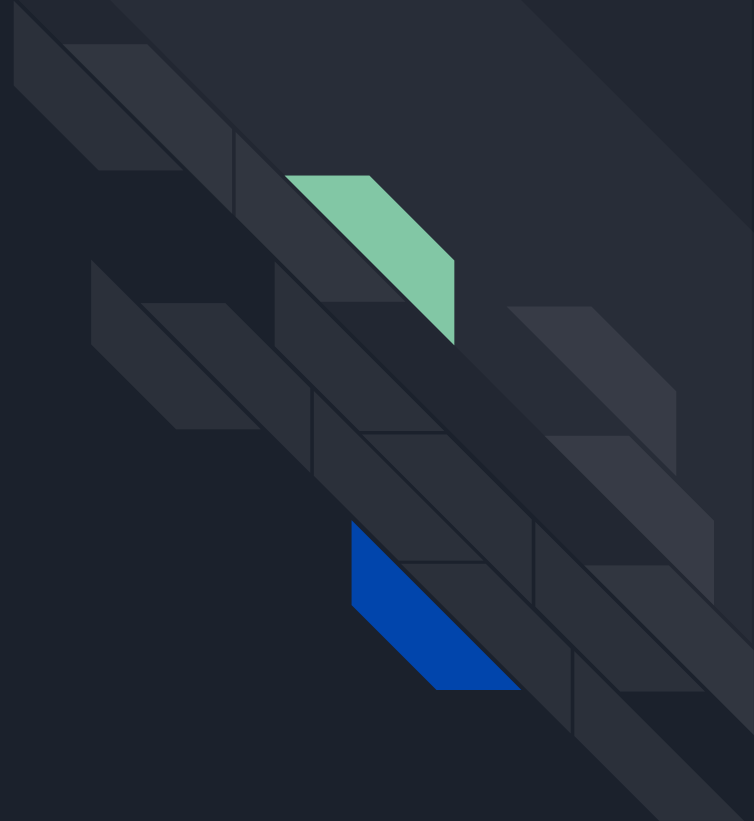


# Electrical System

- 18 Ah battery
- Fusebox
  - 30A fuse for motors
  - 2A fuse for the lower voltage devices
- Breaker switch
- 12V-5V step down for microcontrollers

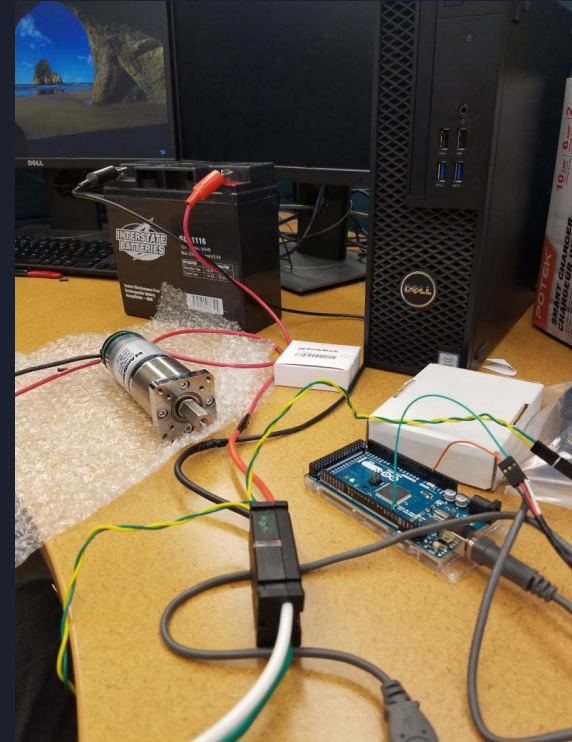


# Embedded Components



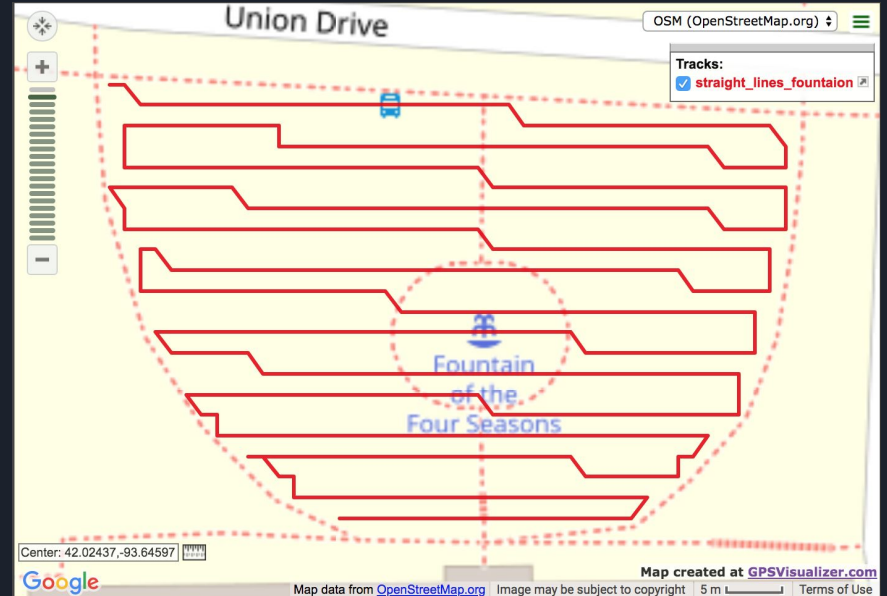
# Motor Control

- 16-bit PWM Motor Control
- Custom Arduino Library



# GPS Data Acquisition

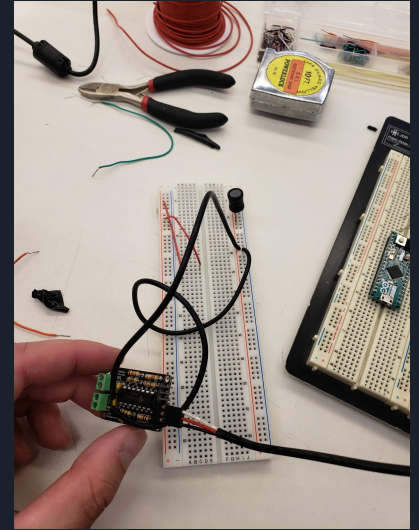
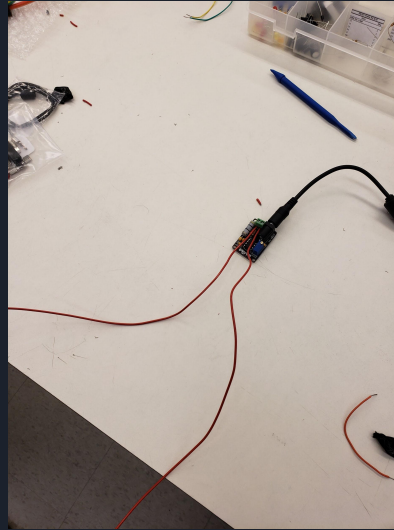
- 20 Hz WAAS GPS
- NMEA 0183

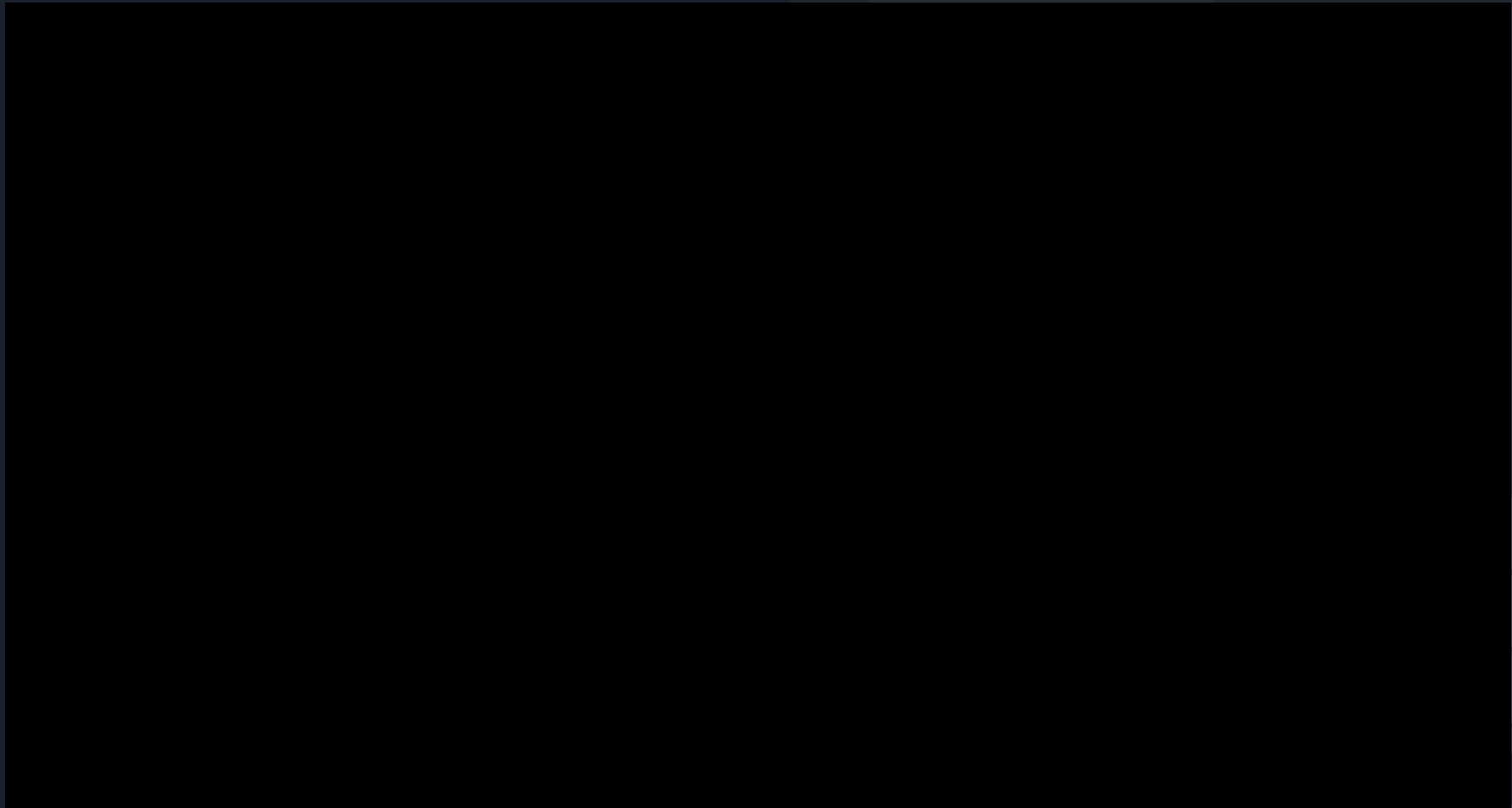




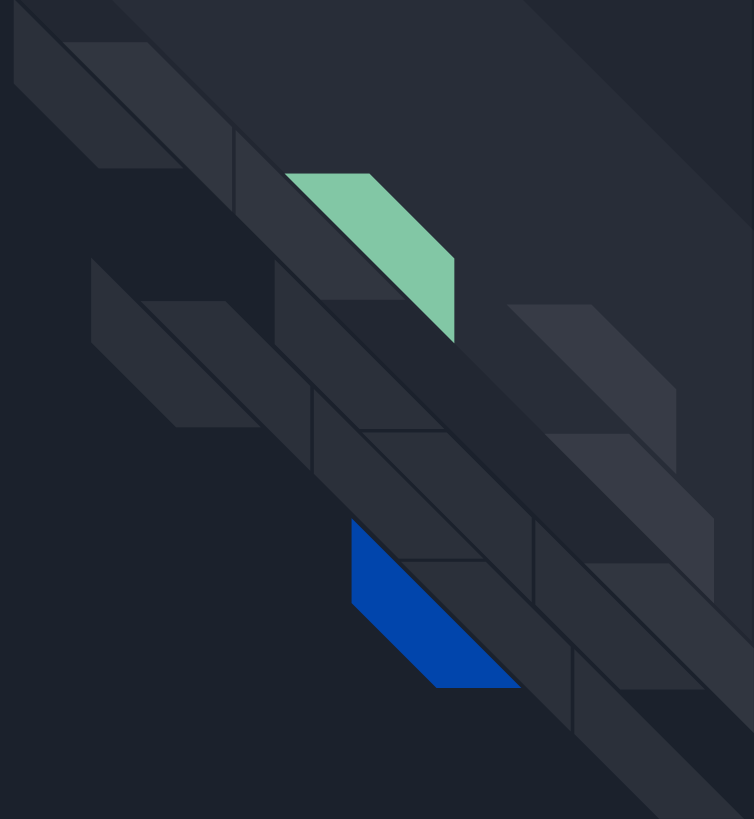
# Perimeter Wire

- 45 kHz Square Wave Emitter Circuit
- 24 Gauge Wire
- 1 mH Inductors
- Tank Circuit with Amplification





# Wireless Interface

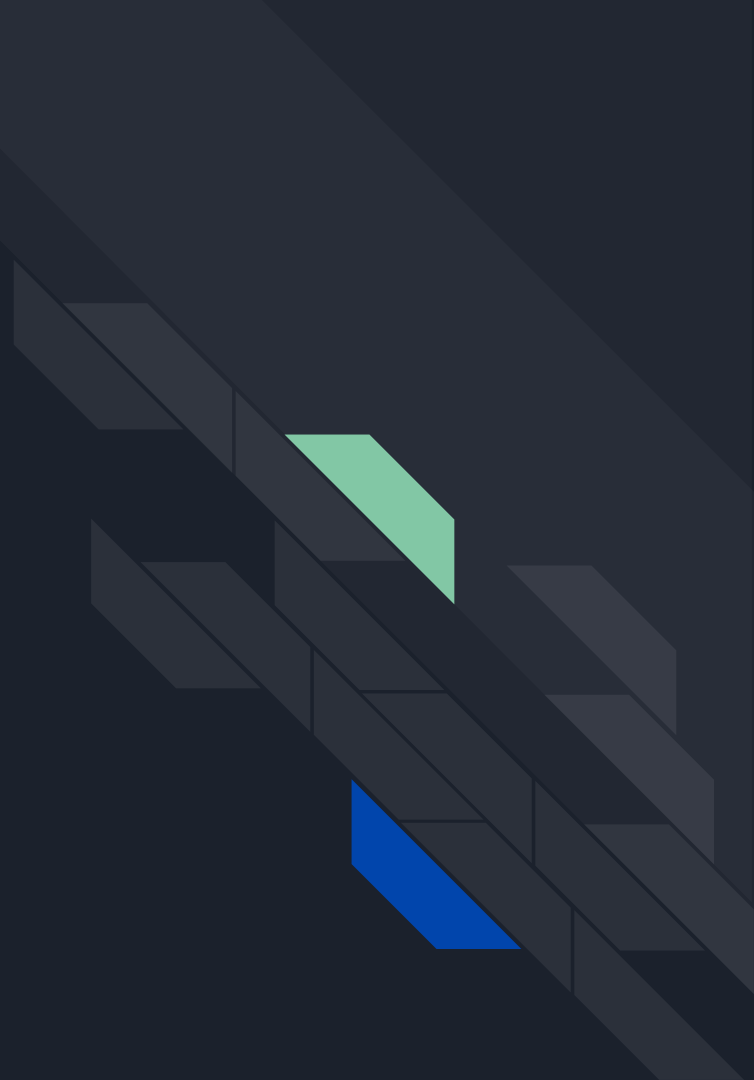




# Raspberry Pi

- **Wireless Communication to Android App**
  - Pi is hosting a REST Api that is being served via FLASK.
  - MariaDB to store tables for the database.
  - Python backend for interacting with the database tables.
- **Wired Communication to Arduino**
  - Serial connection over physical cable.
  - GPS Data
  - Information about when to mow.

# Mobile Application





# Android

- Design
  - Coding done in Java, Kotlin, SQL, and XML
  - MVVM (Model/View/ViewModel) pattern
  - Uses fragments which are contained in one activity
  - Room Persistence database is used for a layer of abstraction over SQLite
  - ViewModel is used to share data between fragments, handle device rotation, and keep the database out of the fragments
  - Google Maps API is used for History view
  - OpenWeatherMap is used to fetch current weather data



# Android

- User Interface
  - Scheduler
    - Users can see, at a glance, the next scheduled mow
    - A CalendarView lets the users select a date to see scheduled mows
    - Users can add, edit, or delete scheduled mows
    - Schedule data is temporarily stored on a SQLite database
  - History
    - Users can see a list of past mows
    - Clicking on a mow allows the user to see the GPS path the mower took overlaid on a Google Map
  - Weather
    - Shows up-to-the minute weather reports for the area
    - Sends alerts to the user's notification of inclimate weather
  - Manual Control
    - Allows users to connect bluetooth to the mower and manually control it



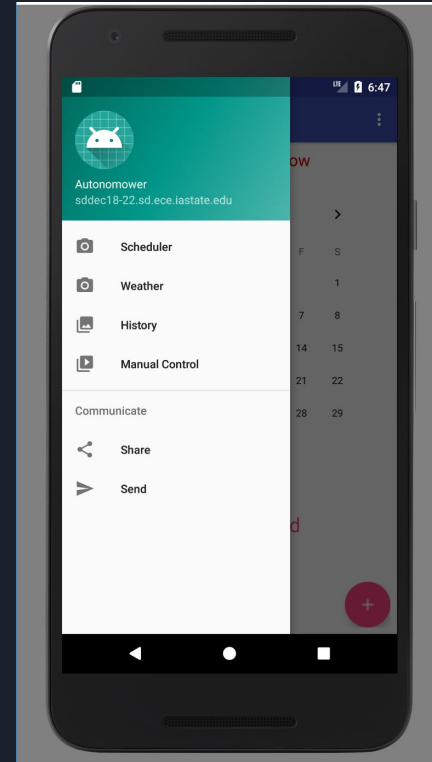
# Android

- **Wireless Connections**
  - **Local Wifi Connection**
    - When the phone and the mower are on the same local WiFi connection, the schedules from the SQLite database are converted to JSON and sent to the mower
    - Incoming JSON data is converted to our SQLite Database and updates schedules mowing history
  - **Bluetooth**
    - When the app is in the Manual Control screen, it establishes bluetooth connection with the mower
    - The user is able to use joysticks to drive the mower



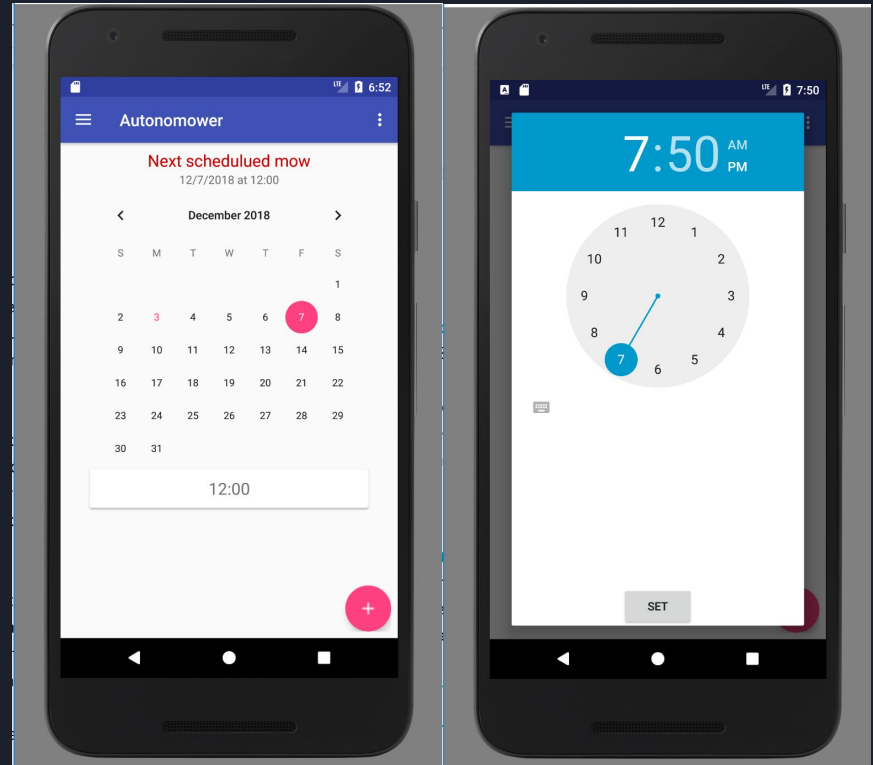
# Android App Navigation

- Side navigation tray



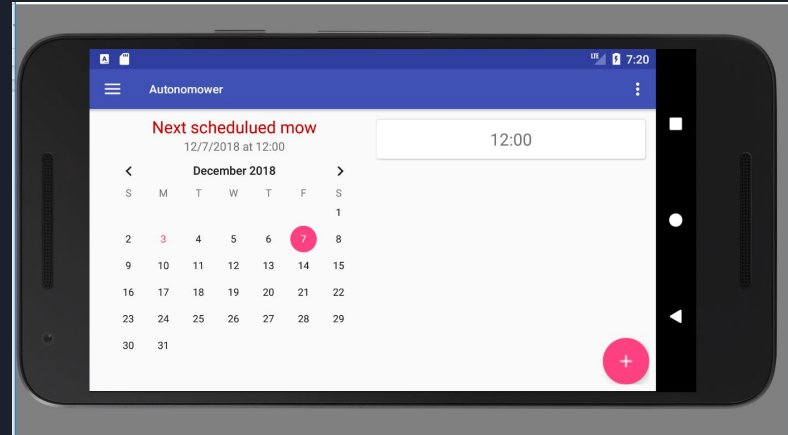
# Scheduler

- Users can see next scheduled mow at the top of the screen
- CalendarView allows the user to select a date and see mows associated with it
- Swipe-to-Delete feature makes deleting mows easy
- Pushing the + FAB on the bottom opens a TimePicker for the user to add a new mow



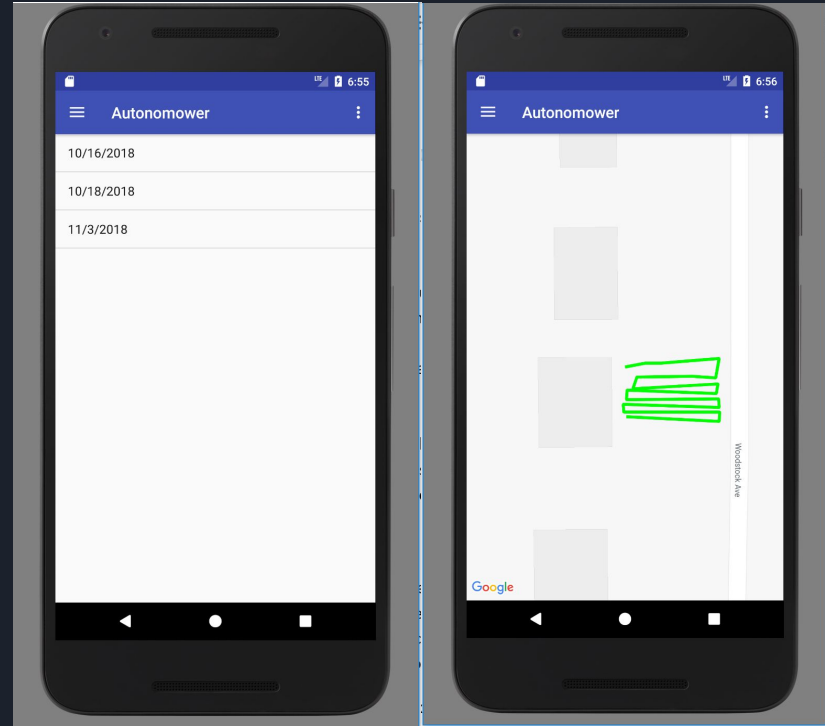
# Scheduler (Horizontal)

- Rotating the device in this screen has an alternate user interface
- Rotation causes no loss of data



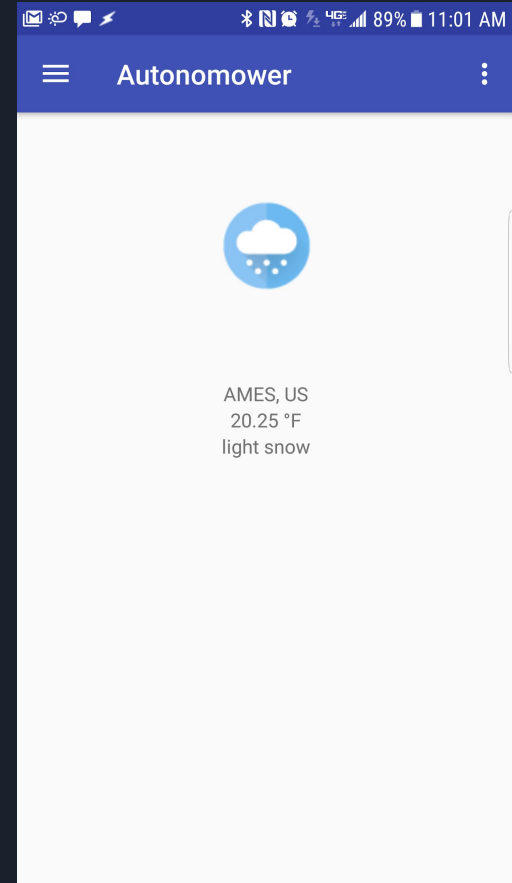
# History

- Initial screen shows a sorted list of past mows
- Clicking on a mow, opens a Google Map fragment with Polylines representing the path of the mower on that mow



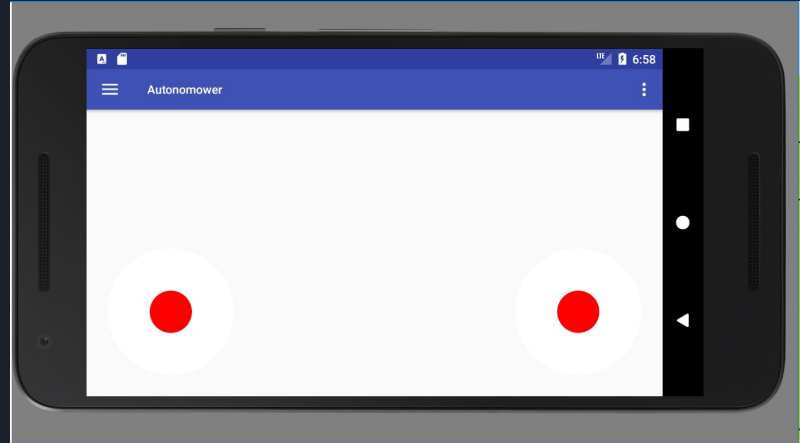
# Weather

- Simple interface which shows the current weather
- When weather detects inclement weather (rain, snow, sleet etc), sends a warning to the user's notification tray



# Manual Control

- Allows the user to use joysticks to control the mower
- Locks the user's screen orientation to horizontal





Questions?