Autonomous Lawn Mower

-Team sddec18-22

Team Members

-Sam Tinklenberg: Team Leader / Software Development

-Andi Li: Meeting Facilitator/ Software Development

-Bryton Hayes: Test Engineer

-Grant Duncan: Software Lead

-Joel Seaser: Hardware Lead

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 incapabilities. Someone who falls into this category does not have many options to get the job done, without hiring expensive, third-party help.

Client Requests/Requirements

-Program that will efficiently mow entirety of area given a mapped perimeter

-System to map perimeter of the lawn

-Object detection and avoidance

-Mobility through standard lawns

-Power efficiency

-Streamlined interfacing

-GPS module for directional guidance and mapping

Deliverables

-Safe and Affordable autonomous lawnmower.

- Affordable compared to others on market.
- Find and avoid hazards in lawn
- Cut entire lawn on one battery charge.

-Android App to control lawnmower

- Set schedule, control via bluetooth, and see stats.

Operating Environment

-Residential

-Dry Lawn

-Water and dust resistant

-Slight hills

Market Research

-Cost

-Cutting area in one charge

-Cumulative Efficiency

-Cutting Width

Basic Implementation

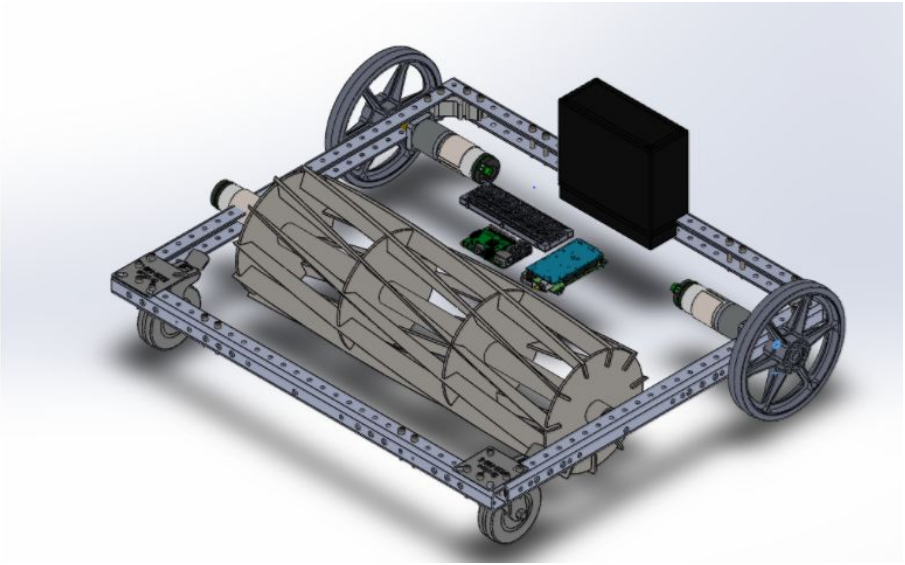
-3 main pieces

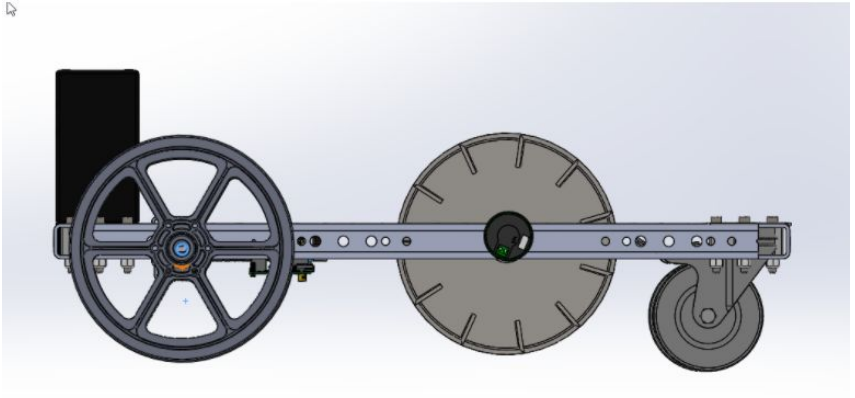
- Arduinos

-Android App

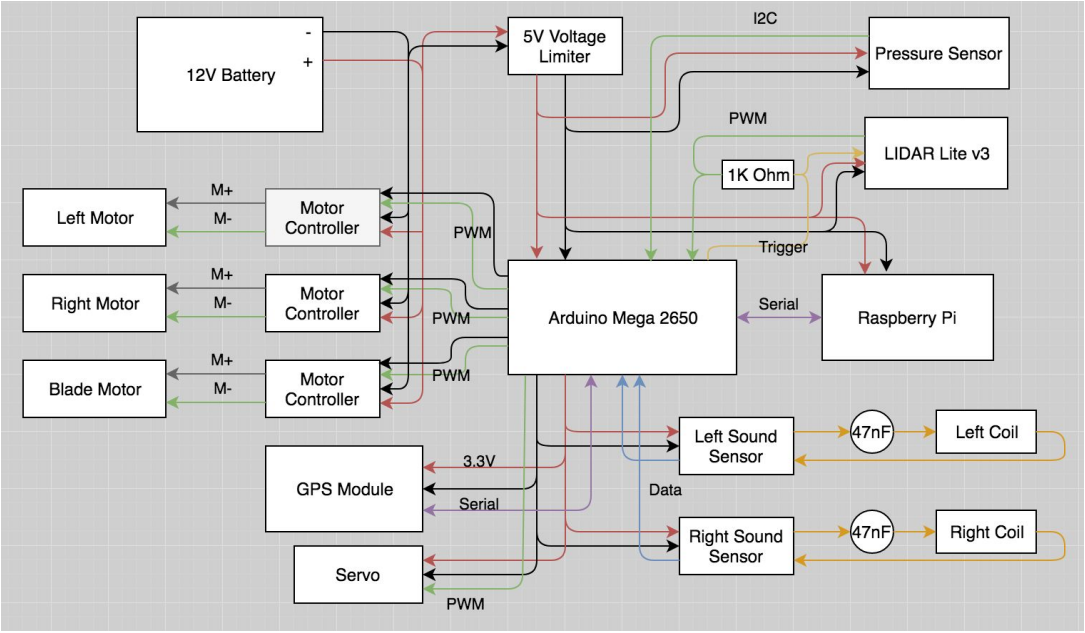
-Raspberry Pi

Prototype Sketch





System Block Diagram



System Description

-Arduino Mega controls motors, handles sensor/GPS data, and communicates with Raspberry Pi

-GPS module equipped with WAAS for accurate positioning data

-LIDAR and pressure sensor components for object detection and avoidance

-12V battery and 5V voltage limiter to power system

-Sound sensors and coil for perimeter wire detection

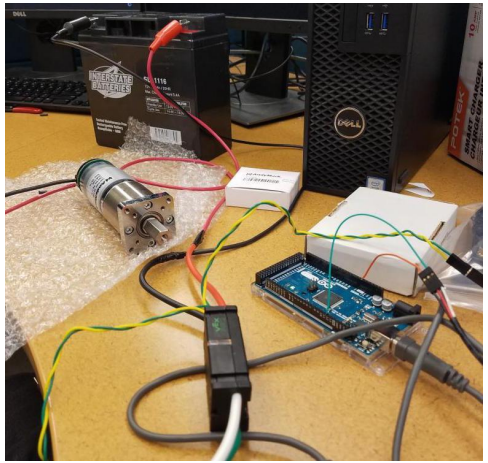
-Arduino micro and motor drivers to control perimeter wire

Testing and Evaluation plan

Software	Hardware	Mobile
-Microcontrollers work with the 3 motors -Mower is responding correcting to direction inputs -Communication is working properly between arduino and raspberry pi	-All components are powered and connected to arduino -Be able to move up 20% grade hill -Mow the average size lawn in one charge	-App be able to connect to the raspberry pi -Data from the mower should be shown to the user -Working log in to server -Be able to remote control the mower

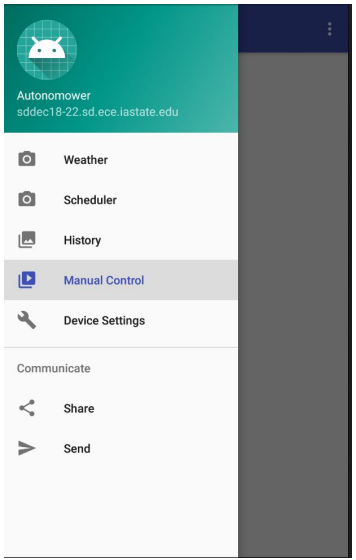
Initial Motor Testing

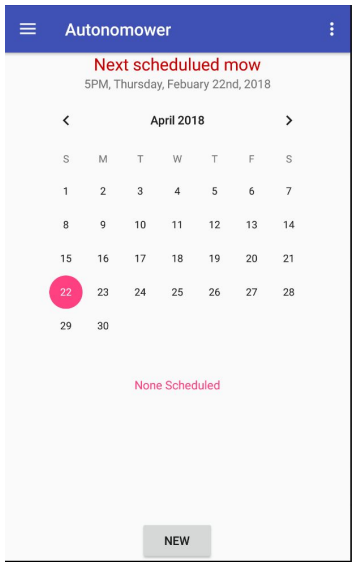
-Tested and validated 16-bit resolution PWM waveforms for motor control

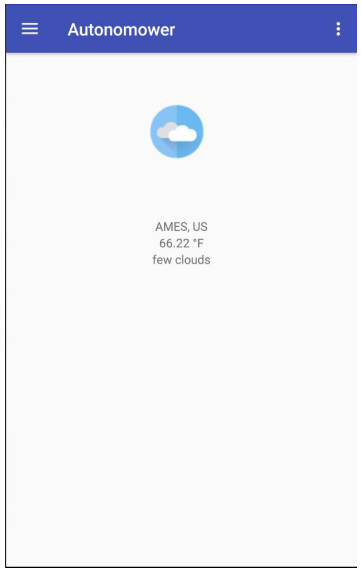


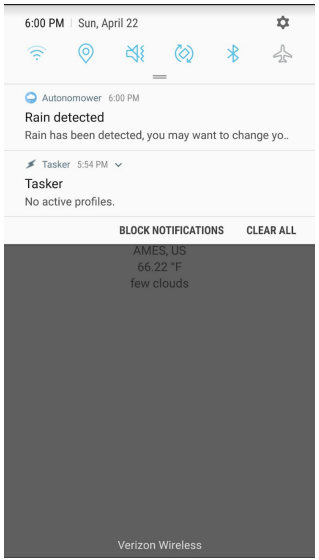
	Expected Pulse Width (ms)	Expected Duty Cycle (%)	setMotor() value	Measured Pulse Width (ms)	Error in Pulse Width (%)
Full Speed Backward	1	24.414	16000 (15999.71)	.9991	0.09
Half Speed Backward	1.25	30.518	20000 (19999.97)	1.249	0.08
Stopped	1.5	36.621	24000 (23999.57)	1.499	0.06
Half Speed Forward	1.75	42.725	28000 (27999.83)	1.748	0.11
Full Speed Forward	2	48.828	32000 (31999.43)	1.998	0.1

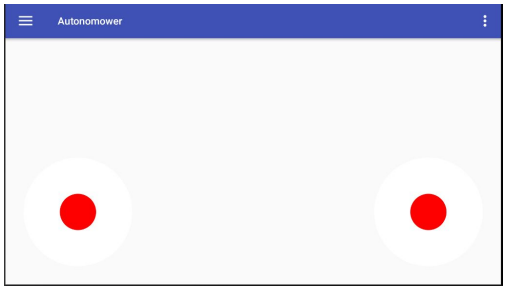
User Interface











Risks

-Lack of parts available to us for a reasonable price

-Not having weather to permit mowing Fall semester

-Lack of knowledge for mechanical side of the lawn mower

Work Breakdown Schedule

February	March	April
Hardware: Research the mower design and different design options based on market research Software: Order the Arduino and Raspberry Pi and start drafting code Mobile: Start App	Hardware: Calculate motor and drivetrain requirements based on problem and client requirements Software: Start testing the the basic functions of our microcontroller, start making prototype sketch in a CAD Program Mobile: Begin writing the UI for the app	Hardware: Order some parts needed for the lawn mower, set up motor, motor controller and GPS, disassemble reel blade Software: Test the microcontroller with the motors and GPS module Mobile: Connect the app with the microcontroller

2nd Semester Timeline

September	October	November	December
Hardware: Assembly of chassis and perimeter wire setup Software: Basic driving features of the lawn mower and perimeter wire detection and sensor data acquisition Mobile: Connect a camera on the mower with the app	Hardware: Refine our mounting mechanism of the lawn mower blade and testing rotation speed Software: Start working on autonomous code for the mower Mobile: Move the mower with the app	Hardware: Creating the docking station for the lawn mower and implementing object detection Software: Implementing the perimeter wire and GPS with the mower Mobile: Add more functionality to the app	Hardware: Finishing the docking station and fine tuning mower Software: Put finishing touches on the autonomous code Mobile: Put finishing touches on the app

Questions?