Autonomous Lawn Care Unit

DESIGN DOCUMENT

Team Number 22
Micron Technologies
Dr. Phillip Jones

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

Email: sddec18-22@iastate.edu
Website: https://sddec18-22.sd.ece.iastate.edu

Revised: 4/22/2018 Version 2

Table of Contents

1 Introduction

1.1 Acknowledgement

If a client, an organization, or an individual has contributed or will contribute significant assistance in the form of technical advice, equipment, financial aid, etc, an acknowledgement of this contribution shall be included in a separate section of the project plan.

1.2 Problem and Project 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.

We intend to solve this problem by designing a safe, functional, and affordable autonomous lawn mower. This will give lawn-owners a relatively inexpensive, low-effort means of maintaining their lawn effectively. Our autonomous lawn mower will be able to mow any predefined area with maximum accuracy and efficiency safely.

1.3 Operational Environment

The general environment for our product will be a dry lawn. As with any lawnmower it, it would have trouble cutting wet grass. The wet lawn would also make it difficult for our lawnmower to retain traction and be able to travel up steeper terrain. Our mower will be spending most its time outside. Therefore, it must have some water and dust resistance. The electrical components are very susceptible to water, and they must be kept dry. Lots of dust could cause static electricity to build up, which could kill the components. The next thing we would have to worry about is the mower overheating. We will need to keep the electronic components out of direct sunlight. When we implement something to keep the components dry and dust free, it would also double as a shield from the sun.

1.4 Intended Users and Uses

Our intended users are anyone who needs their lawn mowed. If you own a home, you most likely need to keep your lawn trimmed. There are many reasons why someone may not be able to d0 this. For example, some people do not have the time, and some people have physical incapabilities.

The uses for our product would be to mow a lawn. We picture it being used by residents and not corporations. The primary goal of our product will be to make the lives of the user's a little bit easier.

1.5 Assumptions and Limitations

- Assumptions
  - Residential, non-commercial use.
  - Battery Powered
  - ⅕ Acre Lawn
  - Resident has Wifi setup
- Limitations
  - Steep Terrain
  - Dry Conditions
  - Grass must not be overgrown.

    - Must be under 45 pounds
    - Can only cut within specified parameters created by the wire

1.6 Expected End Product and Deliverables

- Safe and Affordable autonomous lawnmower.
    - Lawn mower can find and avoid hazards in lawn.
    - It is affordable compared to other mowers on the market.
    - Lawn mower can cut the entire lawn on 1 charge
    - Mower meets the basic safety standards set by IEEE
- Android/iPhone App to control lawnmower.
    - This will tell the lawnmower when and where to mow. As well as view mower stats.
    - Android App will be able to control the mower via Bluetooth
    - The app will also keep track of the mowers location via GPS

We will have the mower completed before December of 2018.

2. Specifications and Analysis

2.1 Proposed Design

Our lawn mower design is to create an RC chassis out of C channels, two caster wheels in the front, and two fixed wheels in the back. The two fixed wheels will each be powered by their own motor. We will use a microcontroller to read the sensors and control movement. For the blade, we decided on using a reel blade that will be attached towards the front of the mower. This blade will be powered by its motor. To make our mower autonomous, we will have boundary wire surrounding the portions of the lawn that will mowed. The mower will use the wire for guidance and mow within the desired perimeter. One design alternative was to modify an existing mower and make it autonomous using our microcontroller code.

Prototype SolidWorks Design:
Refer to Figure 4.1, 4.2, and 4.3

IEEE Standards that we will be following:

Safety:

- IEEE C2: National Electrical Safety Code (2007)

Wireless/Mobile Connections:

- IEEE 802.11p: WIFI this standard is part of the IEEE 802 family(LAN and MAN) networks. This standard is for information technology in LAN and man networks for vehicular environments

- IEEE 1609.2: Standard for Wireless Access in Vehicular Environments— Security Services for Applications and Management Messages(IEEE Std 1609.2): Make sure your wireless connection within a vehicle is secure.

- IEEE 1609.3 IEEE S: The purpose of this standard is to provide addressing and data delivery services within a WAVE system, providing multiple higher layer entities access to WAVE communication services. Upper layer support includes in-vehicle applications offering safety and convenience to their users.

Hardware:

- IEEE Std 1012-2016: Verification and validation (V&V) processes are used to determine whether the development products of a given activity conform to the requirements of that activity

2.2 Design Analysis

In the current stage of the project we have ordered the microcontrollers, motor controller, reel blade, and the GPS system. We started testing each of these parts to make sure they work at the capacity needed for our project. So far each component of our project has been running very well and meets our standards. We have also created a 3D SolidWorks model(Figure 4.1 and 4.2) with our current ordered parts and the parts we plan on buying. One modification we may change with our model would be the reel blade motor placement. Currently we have the motor in-line with the reel blade. This could present a problem in the future as our mower will be very wide. If this method doesn't work we will attach a belt to the reel blade and mount the motor in another location.

3 Testing and Implementation

3.1 Interface Specifications

Listed below is the testing plan for each of our main components to this project.

Testing types:

Software:

- Make sure that there is a good connection between the microcontrollers and the motors of the mower
- Make sure that the mower is moving in the correct direction
- Test if the Arduino and the raspberry pi have a solid connection

Hardware:

- All parts of the mower are connected securely and safely
- All the sensors are receiving accurate data
- The mower is operating at the correct speeds
- Battery is charging correctly

- The mower should be able to withstand certain conditions

Mobile:

- The app should be connected to the mower correctly
- Data from the mower should be shown to the user
- Server for logging in should be working correctly
- The app is able to control the mower wirelessly via Bluetooth

3.2 Hardware and software

Hardware: PWM motor control and GPS data acquisition

We will be testing these parts thoroughly because our project heavily depends on these 2 components. The motor control is pertinent to the mower as it controls the movement of the blade and the mower itself. GPS has to be tested for accuracy as we need to know where our mower is at all times.

Software/Mobile: Android app and MySQL database

We will be testing the android app's ability to connect to the mower through Bluetooth to manually control it and its ability to send and receive data through the MySQL database.

3.3 Functional Testing

Examples include unit, integration, system, acceptance testing

3.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

SAM/GRANT ADD HERE

3.5 Process

Hardware:

Software/Mobile:

The testing of the software and mobile application were very simple. Either the command from the phone was received by the raspberry pi or it wasn't. There are logs on the raspberry pi to view the HTTP requests. If the request adds to the database, then it is very easy to see whether or not the data got inserted into the database. There are also logs to view attempts to add lines to the database.

3.6 Results

Hardware:

We wrote Arduino code to generate a PWM signal on the digital pins and communicate with the motor controller. A 1ms pulse corresponds to full speed backward, and a 2ms pulse corresponds to full speed forward, and a 1.5 ms pulse stops the motor from moving. Initially, we tried using the Arduino analogWrite() function to generate the waveforms, but this only offers an 8-bit resolution, and we had considerable error. analogWrite() uses a 20ms period.

	Expected Pulse Width (ms)	Expected Duty Cycle (%)	analogWrite() Value	Measured Pulse Width (ms)	Measured Duty Cycle (%)	Error in Pulse Width (%)
Full Speed Backward	1	4.896	12 13 (12.48)	.974 1.056	4.765 5.141	2.6 5.6
Half Speed Backward	1.25	6.12	15 16 (15.606)	1.216 1.293	5.956 6.332	2.72 3.44
Stopped	1.5	7.344	18 19 (18.727)	1.456 1.533	7.085 7.461	2.93 2.2
Half Speed Forward	1.75	8.568	21 22 (21.848)	1.693 1.776	8.276 8.652	3.26 1.49
Full Speed Forward	2	9.792	25 (24.97)	2.008	9.812	0.4

Table 2.1: 8-bit resolution PWM using analogWrite()

To achieve 16-bit resolution, we modified the code to use the Arduinos built in timers and adjusted the prescalers to use a 4ms period. With 16-bit resolution, we got very insignificant error and were confident in the waveforms we would be sending to the motor controller.

	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

Table 2.2: 16-bit resolution PWM using timers

Once we had valid PWM waveforms and our hardware, we set up the motor controller, battery, motor, and Arduino to test motor control. We wrote code to change the timer match values on the fly and test the motor with different speeds in each direction and calibrate the motor to be stopped. We got this working with very few issues, and are confident in our ability to control the drive motors. Our plan for future testing of the motors depends on setting up the device itself and adding the motors to determine values for desired speeds and turning capabilities.

We also used an Arduino library to handle GPS serial communication and parsing of GPS data. We setup the GPS module on a breadboard and are successfully receiving our desired GPS data including latitude, longitude, speed, time, and date. The next step is to configure and validate data using the WAAS capabilities of the chip and add it to the device to test the error in our application.

Software/Mobile:

Tests were successful. We ran the mobile code and the raspberry pi HTTP server successfully added it to the database.

4. Closing Material

4.1 Conclusion

Mowing the lawn is something the majority of homeowners need to do on a semi-weekly basis. Sometimes they may not have the time or ability to mow it. With our project, we aim to solve this problem in an affordable and effective way. With our approach, we be able to make a device that is much cheaper than what the current market has to offer. Our mower will also have extra features such as a mobile app, lawn grooming, and automatic charging which should make it even more helpful to the consumer. Our end goal is to create a product that the every-day homeowner can use without any hassle.

4.2 References

"Arduino Code Techniques." *Arduino Code Techniques – Www.wiki.ardumower.de*, wiki.ardumower.de/index.php?title=Arduino_code_techniques.

"Arduino First Steps." *Arduino First Steps – Www.wiki.ardumower.de*, wiki.ardumower.de/index.php?title=Arduino_first_steps.

"Bumper Sensor." *Bumper Sensor – Www.wiki.ardumower.de*, wiki.ardumower.de/index.php?title=Bumper_sensor.

"Drivetrains & Driveshafts." *RC Car Action*, 6 Feb. 2017, www.rccaraction.com/drivetrains-driveshafts/.

Gonzalez Martin, Igor. "Connecting a Parallax GPS Module to the Arduino." *Arduino Playground - GPS*, playground.Arduino.cc/Tutorials/GPS.

Lamb, Robert. "How Reel Mowers Work." *HowStuffWorks*, HowStuffWorks, 19 May 2008, home.howstuffworks.com/reel-mower.htm.

"Perimeter Sender (English)." *Perimeter Sender (English) – Www.wiki.ardumower.de*, wiki.ardumower.de/index.php?title=Perimeter_sender_%28English%29.

Rowe, Martin. "Measure PWM Motor Efficiency." *EDN*, www.edn.com/design/test-and-measurement/4379582/Measure-PWM-motor-efficiency.

" ." *IEEE-SA - The IEEE Standards Association - Home*, standards.ieee.org/.

4.3 Appendices

Figure 4.1 SolidWorks model left plane of view

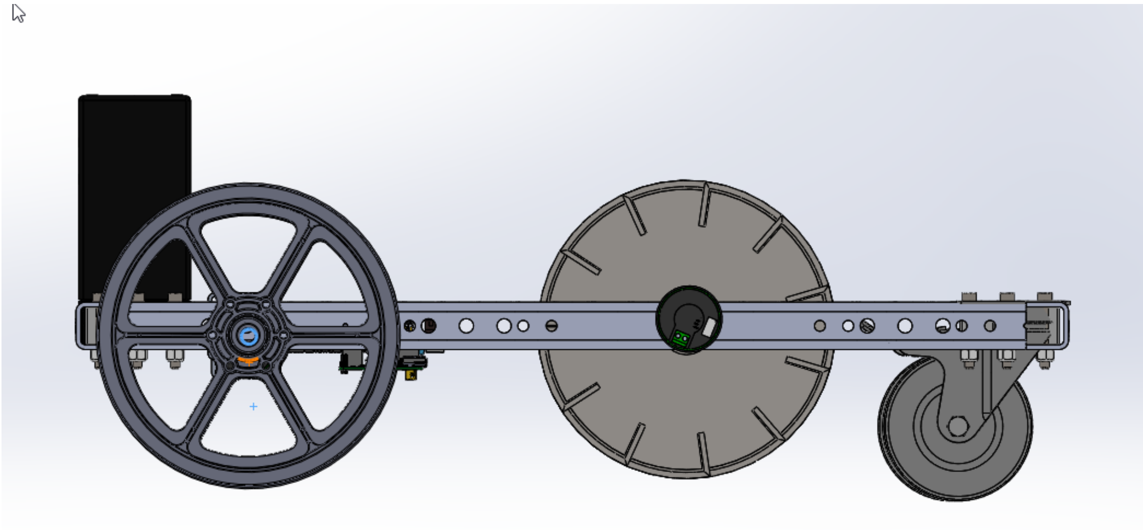

Figure 4.2 3-D view

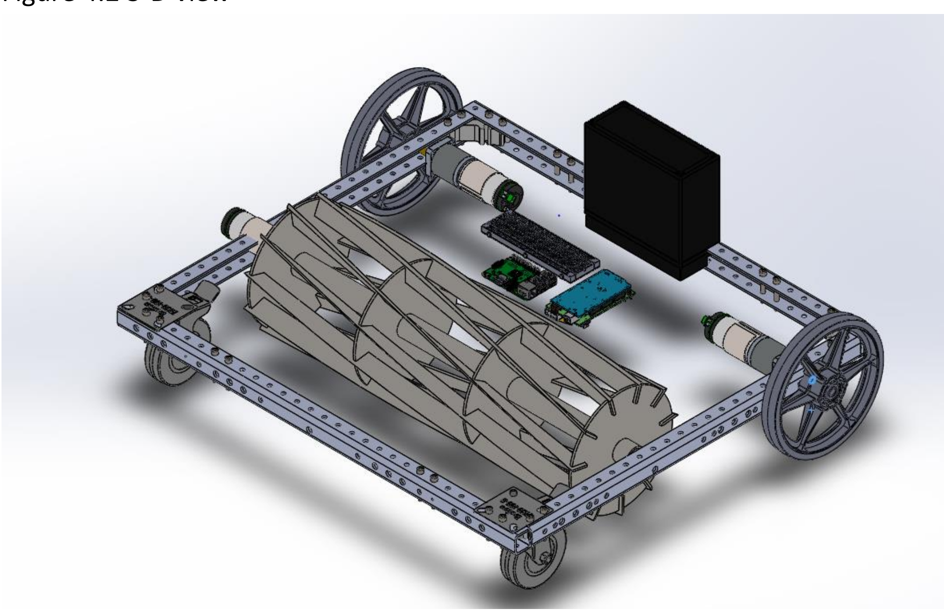

Figure 4.3 Wiring Diagram

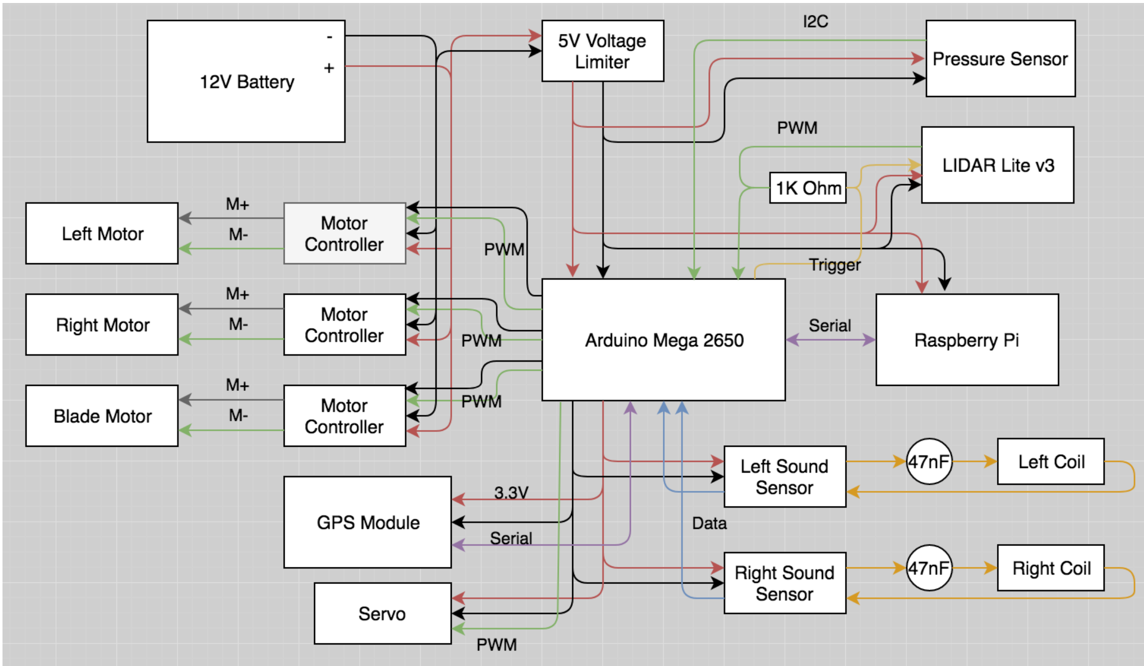

Figure 4.4: Motor control testing setup