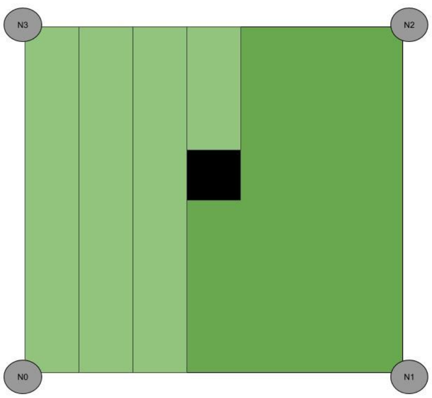
Lawn-O-Matic

Jonathan Scibelli, CSE, Robert Cocomello, EE, Ahmet Yanbul, CSE, and Jeremy Doe, CSE

Our design aims to solve the problem of having to mow a lawn yourself. Using an ultrasonic positioning system, we will provide a solution by building an all electric autonomous lawn mower system. This system will provide a much more efficient and cheaper solution than the current autonomous mowers on the market.



I. INTRODUCTION

The problem that we aim to solve is the annoyance of having to mow your lawn manually. Nobody likes having to mow their lawn, and it can be expensive to hire someone to do it for you, around $1800/yr to be exact. According to the Chicago Tribune, the average American spends 70hrs/yr mowing their lawn [1], over 8 full work days in comparison. Our solution will allow people to instead focus on other matters like spending time with family or taking care of more important tasks around the house, leaving the lawn to our Lawn-O-Matic.

Imagine this instead, you go out to your lawn and place a “node” at each corner of your rectangular lawn. You then grab your lawn mower, place it at the starting node, and let it finish your lawn for you. You watch it mow in straight lines, back-and-forth across the lawn until it reaches the last, ending node. It has finished mowing the lawn and all you have left to do is some weed whacking.

Solutions to this issue have been attempted in the past, however no solution has implemented positioning to mimic the pattern that a human would mow their lawn. In our design, we aim to provide an autonomous mower to mow any lawn (within specifications) in a similar manner to how an individual may mow their lawn manually. In other words, our mower will be aware of its position and its surroundings, and will travel in a straight path across the lawn. Other automatic mower designs on the market operate at random, adjusting their headings upon meeting an obstruction. Inefficiently mowing a lawn in such a way may take hours! Our design attempts to deliver a solution that will efficiently mow a lawn within the time frame of a normal, non autonomous mow. We also aim for our design to be cheaper than other solutions the market has to offer. Automatic mowers on Amazon may range to as much as $2691 [2]. Our design will autonomously mow your lawn at an affordable price.

Our problem will affect individuals who may not be able to physically mow their lawn themselves for any reason, or may not have enough time during their workweek to mow. This project is unlike any other on the market today and will provide a cheap, reliable, and automated alternative for mowing lawns.

The ideal lawn that the Lawn-O-Matic will be mowing is one that is rectangular, completely level, and obstruction free. For a teenager who is looking to make money by mowing lawns, they will still be able to because this system is for flat and rectangular lawns. The Lawn-O-Matic could even mow one section of the lawn while the teen mows and trims in places where the Lawn-O-Matic cannot reach. The Lawn-O-Matic will be great for lawns that need consistent maintenance.

The specifications the lawn mower must adhere to are the following. The mower will be able to mow a lawn that is up to 20 by 20 feet (6 by 6 meters) and will have a margin of error of 6 inches (0.15 meters). This means that even if the mower is 6 inches off of its actual position, it will not miss any grass on the lawn. The specifications of the lawn size comes from the range of the ultrasonic sensors. The range of the ultrasonic transmitters and sensors is 650 feet. Therefore, the largest lawn the Lawn-O-Matic will be able to mow is 650 square feet. The accuracy of the system is determined through calculations based on the speed of sound and the environment in different degrees of temperature and levels of humidity. Using the speed of sound and the equation, $distance = speed \times time$, we were able to come up with projected distance errors if the measurements are off by a certain amount of time. Table 1 shows these values. Temperature and humidity also affect the speed of sound, but after doing some research we’ve concluded that these effects can be ignored. Even at 100% humidity, the speed of sound is only .35% faster than at 0% humidity, given conditions at room temperature and at sea level [3].

Time Error (ms)	Distance Error (mm = inches)
.001	.34029 = 0.0134
.01	3.4029 = 0.134
.1	34.029 = 1.34
1	340.29 = 13.4
10	3402.9 = 134.0 = 11.1ft

Table 1: Sensor Accuracy

When completed, Lawn-O-Matic will be comparable in size to that of an average electric push lawn mower. The lawn mower will be 16”x16” with an aluminum base, weighing approximately 3 lbs [4]. The final product, consisting of an ultrasonic transmitter circuit, an Intel Edison, a LiPo battery for power, and a printed circuit board for the charging system will weigh about 7 lbs. The nodes on the corners, used for calculating the lawn mower’s location on the lawn, will be very similar in size and shape to that of a normal lawn lamp decoration. The specifications can be seen below in Table 2.

Specification	Value
Weight	7 lbs
Range	650 square feet
Battery Life	1 hour
Cost	~$500
Accuracy	6 inches

Table 2: Specification requirement for Lawn-O-Matic solution

II. Design

A. *Overview*

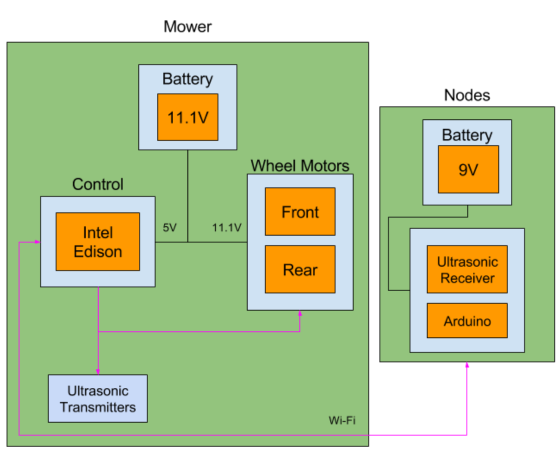


Fig. 1: Our block diagram that shows our design separated by nodes, mower control/Intel Edison, and charging.

The design consists of two main blocks, as shown in figure 1, representing the lawn mower and a single node. On the lawn mower itself, there is a controlling computer, four ultrasonic transmitters, four motors and an 11.1V battery. An Intel Edison is used as the main controlling computer, maintaining the ultrasonic transmitters and wheel motors. Each node operates under the control of an Arduino microcontroller, effectively providing input into the ultrasonic positioning system. The Edison communicates with each Arduino via WiFi, the main medium used for communication throughout the design.

Each node consist of a single ultrasonic receiver, an Arduino, and a 9V battery for power. The Arduino receives information from the ultrasonic receiver and transmits it over WiFi to the Edison.

B. *Block 1: Nodes*

Each of the four nodes is a system of three items that communicate with each other. They include an Arduino, ESP8266 WiFi controller, and the ultrasonic receiver. We had housing for the nodes 3D printed with stakes for them to stick into the ground. The nodes have a 9V battery pack attached to them with a switch attached to the front that when switched on, provides power to all devices inside. A female MIDI connector sits on the bottom of the node and this attaches to a male connector located on the mower. We chose to use a MIDI connector because it was simple to use and provided enough connections. The datasheet for the WiFi controller [5] specifies an operating voltage of between 3.0-3.6V and uses 802.11 b/g/n protocols. The Arduino is used as the main controller to handle the WiFi and incoming ultrasound. We chose Arduinos because they are simple to use. We did not want to overcomplicate the four nodes. The input to this system is an ultrasonic signal and the output is a UDP datagram message which includes the node ID and the time of when the ultrasonic signal was received. The Arduino communicates with the WiFi module using a C++ library that implements and uses serial communication over a BAUD rate of 9600bps (a requirement for the ESP8266). The library includes functions that connect the Arduino to a wireless network, create UDP and TCP connections, and also send and receive data. We decided to use UDP because it is faster and we want to constantly send data. The WiFi module first establishes a connection with the wireless network by the function, `wifi.init(SSID, PASSWORD).` Once it establishes a connection it registers a UDP connection to the Intel Edison over a port and sends a message to the Edison. Each node does this over a different port. Previously, once the Edison had received messages from all four Arduinos, it broadcasted a message, “Start”, back to the Arduinos to tell it to start listening for ultrasonic signal. We realized that there was some delay in WiFi that caused error in the clock synchronization. We changed our plan to have it send a pulse over a wire through the MIDI interface that would be connected to all the nodes. At this time, all nodes are listening for ultrasonic signals. When the Arduino reads the ultrasonic signal, it creates a timestamp and sends this timestamp to the Edison over the UDP connection and keeps doing this until the process is complete.

To test this block, two LEDs were included in the circuitry. A red LED turned on when the Arduino connected to the WiFi and then a yellow one turned on when it received the “Start” message. We knew it was working correctly because when each node was powered on, the red LEDs started turning on and then each of the four yellow LEDs turned on all at the same time, demonstrating that we had timing correctly implemented. To test the ultrasonic receivers, code was included in the firmware that would output, “Received”, to the serial monitor when a signal was received. We moved the receiver away from the transmitter and the Arduino stopped outputting to the serial monitor. We also hooked up an oscilloscope to the output of the ultrasonic receiver to monitor incoming transmissions.

To find the total power dissipated by each node, measurements were made in the lab using a multimeter. To find the power dissipated by the Arduino, the current at the input was measured to be about 105mA. The arduino is powered by a 9V battery so that gives about 945mW of power. Then for the ESP8266 WiFi module, the measured current at the input was found to be 70mA and the module is powered by 3.3V so that gives 231mW. In total, the power dissipated by the node is 1.176W. The capacity for a 9V battery is 500mAh and the battery’s load current is 105mA. To approximate how many hours the battery will last, dividing the charge by the current gives 4 hours and 42 minutes.

Techniques used in this block came from mainly two classes: Software Intensive Engineering taught by Professor Irwin and Electronics taught by Professor Bardin. Software Intensive Engineering taught us how to use both UDP and TCP and how you can use it to communicate between a server and multiple clients. C++ was also another skill learned in Software Intensive Engineering which was required for this block. For the ultrasonic receiver circuit, skills obtained from Electronics were used. We needed a circuit to amplify the incoming ultrasonic signal into something that the Arduino could read digitally. The circuit included two general purpose operational amplifiers and a rectifying circuit at the end to digitize the signal. All of this was covered in Electronics.

C. Block 2: Ultrasound Transmission

The controlling unit in this design will operate on an Intel Edison [6] microcomputer located on the lawn mower. The Edison controls four ultrasonic transmitter sensors, the mower motor, and the four wheel motors. In addition to controlling the devices on the mower, the Edison also operates as both a UDP server for the corner nodes, and a web server for user preferences.

The Intel Edison was chosen because of its minimal power usage, multicore architecture, and small size. Operating at less than a watt [7], the Edison makes for an ideal microprocessor for our lawn mower design. Battery power is a very crucial specification. When compared to a Raspberry Pi model B (part number 756-8308), another possible microcomputer for this design, the Intel Edison will provide up to 25 more minutes of performance in our lawn mower design (derived in Equation 1 - time was calculated assuming the current through the ultrasonic transmitter circuit, roughly 2uA, is negligible, and each motor requires about 450mA at a low rotational speed). The Edison’s lower power dissipation is made possible due to its lower current draw: 150mA [8] as opposed to the Raspberry Pi’s 1.2A! (the 2012 model B) [9].

$$\frac{2.2Ah}{4(450mA)+150mA+1.8\mu A} - \frac{2.2Ah}{4(450mA)+1.2A+1.8\mu A}$$
$$= 0.394\ hours\ \approx 25\ minutes$$

Equation 1: Increase in battery life by using Intel Edison as opposed to the Raspberry Pi.

The dual-core Intel Atom processor within the Edison allows for multiple operations to run in parallel. Performing the UDP server and the mower control operations on separate cores will eliminate the possible unwanted delay due to context switching within the OS and hardware sharing within the processor. This effectively creates a *pseudo-realtime* microprocessor for the motor controls. It is important to note that the Intel Edison processor does not behave like that of a processor on a microcontroller, which executes instructions in *realtime*; thus the term *pseudo-realtime* is used.

Measuring to just about 35mm by 25mm in size [6], the Edison is a great choice for embedded systems where space is of an importance. This size will easily fit into our lawn mower chassis with room to insert our breadboard circuit for the ultrasonic transmission and our PCB design for the battery charger circuit. Other microcomputers, such as the Raspberry Pi (part number 756-8308) measuring to 85mm by 54mm [10], would be otherwise too large.

The software running on the Edison communicates with the onboard ultrasonic transmitter circuit via electrical signals generated as short pulses from the Edison's GPIO pin. The four ultrasonic transmitter sensors within the circuit provide for the location system by periodically transmitting ultrasound to a maximum distance of 650 feet. The sensors are separated equidistant from each other, each 90 degrees from its previous, to maximize the area transmitted. The Edison activates the 40kHz ultrasonic transmission by pulsing a single GPIO pin HIGH to 3.3V then LOW to 0v at a frequency of 50kHz for 14ms every 2s, as shown in Figure 2 below. The delay between each transmission is important to compensate for the many possible delays over WiFi. The transmission process occurs on a single thread on the Edison to allow continuous output to the GPIO pins.

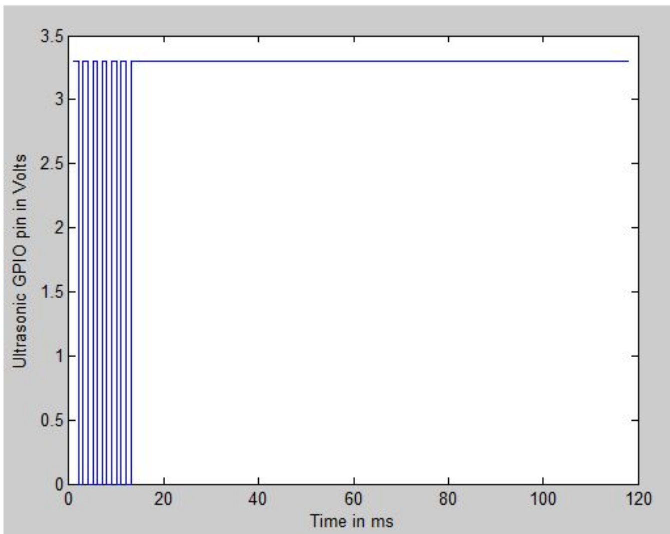


Fig. 2: Time plot of 14ms GPIO pulse used to activate the ultrasonic transmitter sensors. The first 120ms is shown to clearly display the pulse.

The mower motor operates only within the boundaries of the user's yard. With help from the ultrasonic sensors to determine an accurate position, the Edison may activate or deactivate the mower by controlling the power flowing into the mower motor via a relay. However, due to limitations in UMass's safety rules on SDP, we have opted for no blades on this prototype.

If the Edison calculations determine that the mower is within the boundaries of the lawn, it will indicate the mower is on by lighting an LED within the power switch..

The Intel Edison will have an active UDP server during the operation, listening specifically for packets from each of the four corner nodes. The server operates as the Edison's main source of input data needed to compute its calculations and determine its position. Running on its own thread, the server will actively listen for incoming packets to break apart and store locally for the Edison's local calculations. The UDP server is also a very important for synchronizing the clocks across each node and the Edison itself on startup. Timing is very critical for calculating the in-flight time of each ultrasonic pulse as they travel from the transmitter on the mower to the receiver on one of the four nodes. Thus, the Edison relies on the walltime from not only itself, but each of the Arduinos as nodes too.

Once started, the Edison creates a one time UDP server to synchronize the time between the nodes and itself. It first waits for each node to properly connect to WiFI by actively listening over UDP. Each node will send a packet over UDP once connected, notifying the Edison that such node is "online." Once the Edison received a packet from each node, it transmits a "start" packet to each node, notifying the nodes to create a local timestamp as a reference to the initialization of the system. Using the difference between any time after the start and the start itself, it will not matter if the clocks are set to the wrong time and date on any device. The Edison also records a timestamp a few milliseconds after to accommodate for the delay over WiFi. With this design, the Edison and Arduinos always refer to a timestamp relative to the start of the system.

D. Block 2: Mower Control

Each wheel on the mower has their own motor, for a total of four motors. The two left motors and the two right motors are each connected in series.The Intel Edison controls the left two motors independently from the right two motors. To steer the mower and adjust for error, the Edison must rotate one pair of the wheels faster than the opposing pair. In the event that the mower must turn around and travel in the opposite direction (such as in the case of reaching the length of the yard), then the mower will stop, rotate one pair of motors forward while rotating the otherkbackwards to perform a zero turn. The Edison will perform this process by writing to a DRV8835 dual motor driver carrier. The carrier has an H-bridge on it to control the four DC motors. The Edison may increase or decrease the power to the motors, and thus their rotation speed, by sending an electrical pulse of variable width to its GPIO pins connected to the ECSs. This process is called pulse width modulation (PWM). To control the direction of rotation, the Edison writes either a high or low to the carrier, changing the direction the current flows in the motors.

The motors are each rated at 71 rpm with a stall torque of 85 oz-in. The torque is quadrupled for the four motors and multiplied by the diameter of the wheels to get the maximum weight that can be carried by the motors.

If the mower travels off the path while mowing, a feedback control loop will correct it. The edison will change the speed of the motors on the side that needs correcting until the mower is back on the path. Our feedback loop is a proportional control loop with a proportional gain of 1ms/ft to provide a fast response. Since the motors operate via PWM signals, increasing the PWM signal slightly (i.e. by 1ms in our 50ms PWM pulse) results in the motors increasing by 0.11 MPH, enough to adjust the mower back on course while traveling at least 6 inches from its desired location. Any more responsive, and the mower will begin to oscillate in the feedback control. The PWM signals must also be whole numbers, thus we could implement a proportional gain less than 1ms/ft.

E. Block 3: Power

The 11.1V battery is a 20C, 2200mAh lithium poly-ion battery which will supply power to both the Intel Edison and the four wheel motors that move the mower. For convenience of the consumer, this battery will be charged at the same time as the 36V battery. A custom printed circuit board will be designed and fabricated to draw power from the mower battery to the 11.1V lithium poly-ion battery.

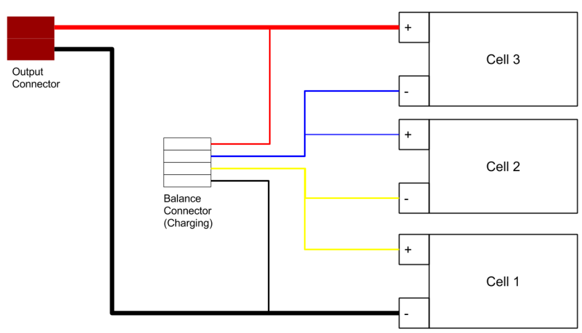


Fig. 3: Schematic of the lithium poly-ion battery.

The battery is charged via the four connections on the balance connector, as seen above in Figure 3. The 11.1V lithium poly-ion battery has an actual maximum voltage of 12.6V, with each cell having a maximum voltage of 4.2V when fully charged. The maximum current allowed into each cell during charging is 2.2A. For consumer convenience, there will be an LED on the circuit indicating when each cell of the battery is currently charging.

The charging circuit is designed to charge each cell up to 4.2V and no higher. When a cell is in its "charging" state a maximum current of 1.67A is allowed through said cell, 1.67A being about 75% of the maximum current of 2.2A. A simulation of the circuit in its "charging" phase can be seen in Figure 4. As the voltage of a cell increases and approaches 4.2V, the current going into said cell decreases until the voltage of the cell is, or nearly is, at 4.2V. At that point the current going through the cell will be practically zero making sure that the cell does not exceed its maximum voltage of 4.2V, damaging the battery. A simulation of the circuit in its "charge complete" stage can be seen in Figure 5.

The top left of the circuit in the schematics makes up the LED circuit, where D1 simulates the actual LED, that indicates when a cell is currently charging. As seen in Figure 4, when the cell is charging there is current going through D1 indicating it is lit and in Figure 5, when the cell is completely charged there is essentially no current going through it. The schematics do not show it due to program limitations, but all 3 cells will have an LED.

The most important components used in the circuit are the LM350T voltage regulator and the NP2N2222A transistor [9, 10]. The LM350T, parts U1, U2 and U3 in the schematic, in conjunction with resistors R2, R3, and R4, make up the voltage regulator of the circuit and ensures that the cells do not exceed their limit of 4.2V. The NPN2N2222A transistor, parts Q1, Q4, and Q6 in the schematic, in conjunction with R1 sets the current limit of 1.67A.

After running simulations, we put the circuit together on a breadboard for testing on an actual li-po battery. Our design was able to successfully charge our li-po battery, even balancing out each cell to have about the same voltage as seen in Figure 6. The current in each cell also decreased towards zero as the voltage increased, as seen in Figure 7.

The design of the custom PCB was completed and the board was fabricated. Upon completing assembly of the components, the board was tested. An initial test determined the output voltage was incorrect. Further voltage tests throughout the circuit concluded in between a 2 and 3 voltage drop from the simulations and circuit on the breadboard. Since this error appeared everywhere within the circuit, we plugged the battery in to test current. The current test resulted in no current flowing into the battery, resulting in no charging.We concluded that there was an issue in the PCB design on the software side of things and unfortunately had no time to correct this mistake.

Techniques used to build this block come from the courses Circuit Analysis I & II and Electronics I. These courses made us familiar with circuits, how to design, simulate, and test them using various electronic test instruments.

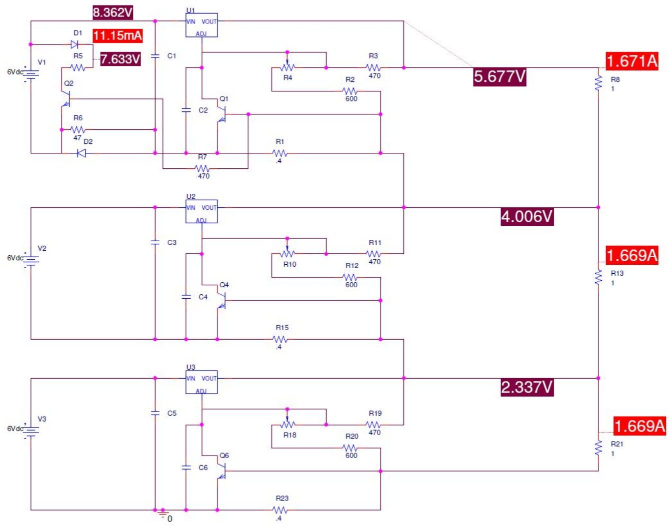


Fig. 4: Schematic of the circuit that will charge the lithium poly-ion battery. This is a simulation of the circuit when it is in its "charging" state.

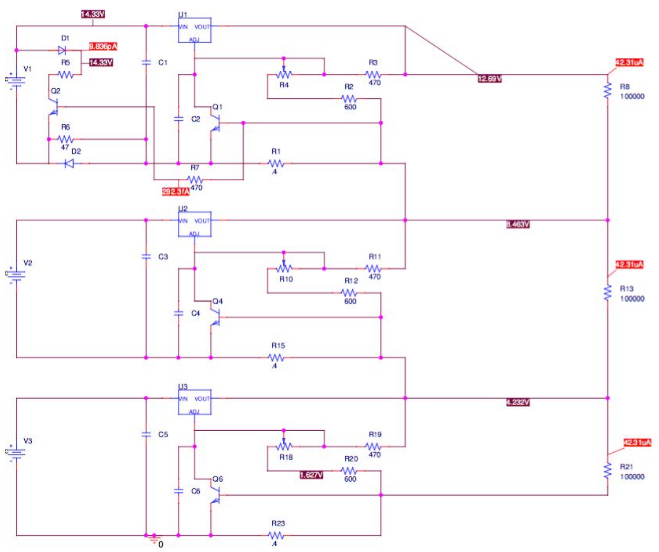


Fig. 5: Schematic of the circuit that will charge the lithium poly-ion battery. This is a simulation of the circuit when it is in its "charge complete" state.

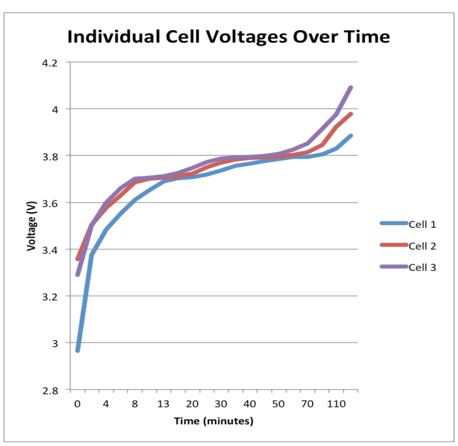


Fig. 6: Graph of voltage in each cell of a li-po battery as it was charging in a 2 hour timespan.

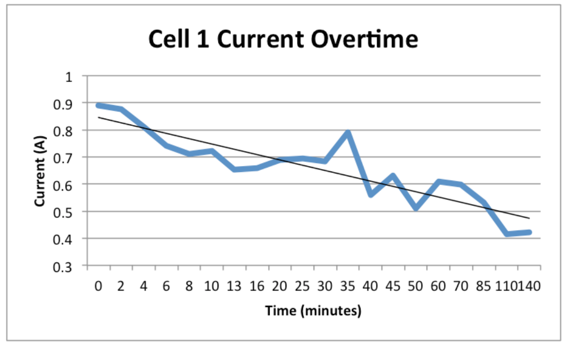


Fig. 7: Graph of current in cell 1 as it charged; coincides with voltage in Figure 6.

The current into the DC motors was measured to ultimately determine the battery life of the LiPo battery. The current during a single motor's maximum rotational speed was measured to be 1.1A. The motors will operate at a constant high speed while underway, consuming 1.1A each, and dissipating 48.84W of power between the four. At this speed, the mower will have a maximum battery life of 30 minutes, as derived below in Equation 2 (150mA from the Intel Edison, 1.8uA from the ultrasonic transmitter circuit). If we were to do this again, we would use a more powerful battery to meet our goal of a minimum 1 hour battery life.

$$\frac{2.24h}{4.4A+150mA+1.8\mu A} = .5h = 30min$$

Equation 2: Battery life time

E. Web Server

The Intel Edison onboard the mower also hosts a server for the user to interact with. The server's main purpose is to allow the user to set the size of their lawn so the Edison could compute its calculations accordingly. The Edison will also provide the user with real time monitoring data hosted locally to the Edison over a separate web server. Users connected on the same network will have access to an accurate position of the mower in the lawn. This product is intended to be used by a consumer, thus it is important that the user's experience is well itemized when bridging the gap between the software and hardware of the system.

The web server was created using Python's Flask libraries and hosts an HTML page with embedded JavaScript displaying the mowers locatio. For simplicity, the mower's location is represented as a point on a cartesian plane using JavaScript's Chart.js libraries, periodically updating in time to provide the user with realtime locations.

III. Project Management

FPR Deliverables	Status
Motor Control with Edison	Completed
Charging Circuit PCB	Incomplete
Web Server	Completed
Ultrasonic Positioning	Incomplete

Table 3: FPR Deliverables

As Table 3 states, both the motor control and web server have been completed. The charging circuit PCB and ultrasonic positioning are both incomplete. The issues with the PCB comes when transferring the design to a manufactured circuit board.

The control system on the lawn mower is able to transmit ultrasonic beams as well as communicate with the nodes via Wi-fi. The location algorithm has been completed in the control system. The algorithm has been tested and confirmed to be accurate. The integration of the communication and location has not been completed. There is more to be done to complete the implementation of the location detection system.

The team has been working well to communicate and complete integration of the systems as progress is made. Helping each other with their weaknesses is an important aspect of the teamwork.

The following expertise of each member of the team is as follows:

As a computer systems engineer, Jon's expertise is in hardware and software architecture within application specific designs. His previous experiences in software engineering at ISO New England and Brown Brothers Harriman have helped him gain advanced engineering insight at the professional level. Also, his previous tricopter, RC car, and automated plant watering system projects involved embedded system design, software architecture, and automation.

As a computer systems engineer, Jeremy's expertise is in embedded systems and software engineering. He interned as a software engineer for Allscripts Healthcare Solutions where he gained valuable experience in the software lifecycle and has been able to combine it with experience in school projects in hardware design.

As a computer systems engineer, Ahmet's expertise is in application development and software engineering. His previous experiences in school-related and personal projects have helped him gain valuable engineering experience.

As an electrical engineer, Robert's expertise is in circuitry. His experience in electronics and circuits shows up in the design and testing of the charging and ultrasonic sensor circuits. He has knowledge of components such as BJTs and MOSFETs to contribute to design. His ability to use and interpret measurements from electronic test instruments such as oscilloscopes, multimeters, and function generators contributes to debugging of circuits.

IV. Conclusion

The current state of our project consists of our FPR deliverables: the finished 3D printed and assembled ultrasonic receiver nodes, the UDP communication over WiFi, the Intel Edison/Arduino ultrasonic transmission circuitry and code, the Intel Edison position calculations, the synchronization of each CPU clock among the four corner nodes and the Edison, and a schematic and simulations of our charging circuit for the 11.1V lithium poly-ion battery. We have completed full motor control of the mower with the Edison and a complete web server built into it. The incompleted deliverables are the positioning system and the PCB charging circuit. We've achieved where we are by working diligently, meeting as a group at least two times a week, meeting with our advisor, Professor Anderson, once a week, and making sure everyone was keeping up with their part and helping each other when needed.

We've done a cost analysis of the entire system and met our goal of completing the project under $500. Table 4 lists of all the parts that have been included in the system. Production wise, we can build this with just over $320.

Part	Development	Production (1000)
Motor System	$150	$11.062
Intel Edison	$50 (Previously owned)	$50
Node Housings	$72	$72
Node Internals	$88	$88
PCB	$5	$5
Arduino Block	$14.95	$12.71
lm350T	$2.61	$1.11
Resistors	$5.58	$2.89
Capacitors	$4.50	$1.30
Transistors	$4.40	$.93

Circuit Boards	$29.70	$22.50
AC/DC blocks	$30	$30
Chassis	$35	$35
Total	$497.74	$322.50

Table 4: Cost analysis

The end goal was to provide a cheap and reliable method of mowing lawns with little to no physical labor. The main difficulties that we expected to face and did face included the location system's increase in accuracy and the mower's increase in battery life. Both have proved challenging, however we've strived for an affordable automatic lawn mower for consumer use. In the end, the team is proud of how much was able to be accomplished. Team 5 has overcome many challenges throughout the year and SDP has been an incredible learning experience. A subset of the group will continue to work post-graduation to complete the non-working subsystems.

Acknowledgment

We would like to thank our advisor Neal Anderson for guiding and supporting us over the course of senior design. He has set an example of excellence as a mentor, instructor, and role model. We would also like to thank Professors Doug Looze, Mike Zink, and Christopher Hollot as evaluators during this project.

References

[1] C. Ingraham, "Lawns are a soul-crushing timesuck and most of us would be better off without them," in *Chicago Tribute*, chicagotribune.com, 2015. [Online]. Available: http://www.chicagotribune.com/news/opinion/commentary/ct-stop-mowing-your-lawn-20150805-story.html. Accessed: Dec. 23, 2016.

[2] *Amazon*. [Online]. Available: https://www.amazon.com/Husqvarna-220-AC-8-7-Inch-Automower/dp/B0025Z2P3I/ref=sr_1_4?ie=UTF8&qid=1485715076&sr=8-4&keywords=automatic+lawn+mower. Accessed: Jan. 29, 2017.

[3] J. Brennan, "How does humidity affect speed of sound?," Sciencing, 2017. [Online]. Available: http://sciencing.com/humidity-affect-speed-sound-22777.html. Accessed: Feb. 4, 2017.

[4] S&F Supplies, "Weight Chart for Aluminum Sheets" Painted Aluminum Sheets

[5] "ESP8266 802.11bgn Smart Device Datasheet," in *Adafruit*. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/ESP8266_Specifications_English.pdf. Accessed: Jan. 29, 2017.

[6] Intel Corporation, "Intel Edison Compute Module," HG331189 datasheet, June 2016 [Revision 005].

[7] SFUptownMaker, "The Edison is Not a Raspberry Pi," *Sparkfun,* Sep. 25, 2014. [Online]. Available: https://www.sparkfun.com/news/1603. [Accessed Jan. 29, 2017]

[8] Peterson,Plourde, J, "First Three Days With an Intel Edison," OpenICE, July 24, 2015. [Online]. Available: https://www.openice.info/2015/07/24/three-days-with-intel-edison.html#battery

[9] "Raspberry Pi Faqs - Frequently Asked Questions". *Raspberry Pi*. N.p., 2017. Web. 6 Feb. 2017.

[10] RS-Components, "Raspberry Pi Model B," 756-8308 reference sheet, April 2012.

[11] Texas Instruments, "3-Amp Adjustable Regulators," LM350T datasheet, May 1998 [Revised March 2013]

[12] ON Semiconductor, "Small Signal Switching Transistor" NPN2N2222A datasheet, Nov. 2013 [Rev. 2]

[13] Jameco Electronics, "Ultrasonic Sensor Set" 40T/R-12B datasheet