

ECE-445

# Smart Sprinkler Robot System

Final Paper

Denis Kurtovic, Kevin Johnson, and Jose L Orozco Jr.

TA: Mustafa Mukadam

May 1<sup>st</sup>, 2013

## **Abstract**

The project we are doing is the Smart Sprinkler Robot System. We decided on this project because we wanted a cost effective way for people to reduce lawn watering. When someone uses a traditional sprinkler, it is difficult to monitor the amount of water being used, which leads to a lot of waste. For installed sprinkler systems, they have the capability to efficiently water your lawn, but the installation is cost prohibitive. Our project combines the flexibility of a traditional sprinkler with the water savings of an installation sprinkler. This project is exciting because this is an area of the market that is underdeveloped. It has tested our skills from many different areas of Electrical Engineering to put together this product.

# Table of Contents

|                                        |    |
|----------------------------------------|----|
| 1. Introduction.....                   | 1  |
| 1.1 Goals.....                         | 1  |
| 1.2 Functions .....                    | 1  |
| 1.3 Benefits.....                      | 1  |
| 1.4 Features .....                     | 1  |
| 2. Design .....                        | 2  |
| 2.1 Power Electronics.....             | 2  |
| 2.2 H-Bridge Circuit.....              | 3  |
| 2.3 Wireless Communication .....       | 5  |
| 2.4 Microcontrollers .....             | 6  |
| 2.5 GPS.....                           | 6  |
| 2.6 Motors .....                       | 7  |
| 2.7 Soil Probe .....                   | 7  |
| 2.8 Internet .....                     | 8  |
| 2.9 Hose Actuator.....                 | 8  |
| 2.10 Battery .....                     | 9  |
| 2.11 Power Outlet.....                 | 10 |
| 3. Requirements and Verification ..... | 11 |
| 3.1 Power Electronics.....             | 11 |
| 3.2 H-Bridge Circuit.....              | 11 |
| 3.3 Wireless Communication .....       | 11 |
| 3.4 Microcontrollers .....             | 12 |
| 3.5 GPS.....                           | 12 |
| 3.6 Motors .....                       | 12 |
| 3.7 Soil Probe .....                   | 13 |
| 3.8 Internet .....                     | 13 |
| 3.9 Hose Actuator.....                 | 13 |
| 3.10 Battery .....                     | 14 |
| 3.11 Power Outlet.....                 | 14 |

|                                 |    |
|---------------------------------|----|
| 4. Cost Analysis .....          | 15 |
| 4.1 Labor Cost .....            | 15 |
| 4.2 Parts Cost.....             | 16 |
| 4.3 Grand Total .....           | 16 |
| 5. Conclusions.....             | 17 |
| 5.1 Accomplishments .....       | 17 |
| 5.2 Uncertainties.....          | 17 |
| 5.3 Ethical Considerations..... | 17 |
| 5.4 Future Work .....           | 18 |
| 6. References.....              | 19 |

# 1. Introduction

## 1.1 Goals

The overall goal of this project was to design a sprinkler system that determines when a lawn needs to be watered and how much it needs to be watered. Below are the sub-goals for this:

- Wireless communication system between a robot and base station
- A GPS system to determine where the robot is and where it needs to go
- Regulate the power to the different modules
- Gather weather data to predict when a lawn needs watering
- Measure the soil moisture in a lawn and determine how long to water it

## 1.2 Functions

The Smart Sprinkler Robot System is composed of a base station and a sprinkler robot. The base station monitors the weather and sends out the sprinkler robot to take measurements of the soil moisture. When it is determined that an area needs watering, the sprinkler robot waters it. The sprinkler robot repeats this process of measuring and watering until the entire lawn is covered. Then the sprinkler robot returns to the base station until the next time the base station activates it.

## 1.3 Benefits

- Reduces the amount of water wasted from watering
- Can be easily moved and set up at a new location
- Does not require user command to water (automatic system)
- Saves on the time it takes for someone to water (all automatic)

## 1.4 Features

- Records and stores soil moisture data from different lawn areas
- Predict when the lawn needs watering
- Innovative wheel design prevents damage to the lawn
- Determines the optimal time of day for watering
- Wireless communication between robot and base station
- 360° sprinkler watering pattern
- Soil moisture measurement device

## 2. Design

### 2.1 Power Electronics

Initially, our preliminary design included a step-down converter that brought down the battery voltage from 12V to 5V as can be seen in Figure 1. The Arduino sent a 5V pulse width modulated (PWM) signal into a LTC4441 chip. This chip then outputs a 12V signal at 32 kHz. The duty ratio of the step-down converter is given as:

$$D = \frac{V_{OUT}}{V_{IN}}$$

Equation 1. Duty Ratio of Step-Down Converter

The output current ripple is given by:

$$\Delta I_{OUT} = \frac{V_{IN}(1 - D)}{L} DT$$

Equation 2. Output current ripple

Based on these two equations, we chose to use an inductor of 75  $\mu$ H. From here, we chose large enough capacitors for the input and output to filter out any ripple. These values were  $C_{OUT}=470\mu$ F and  $C_{IN}=470\mu$ F. A MOSFET would have also been chosen with the least  $R_{DS,ON}$  to reduce the power loss. The initial circuit is can be seen in its entirety in Appendix C.

In the end, we chose not pursue the step-down converter method for the design review because we had a lot of modules that needed to work together at the same time. This step-down system could have been regulated with a microcontroller to output different voltages. Since most of our circuits required either 3.3V or 5V at the output, we realized that using low-dropout regulators would be a better and cheaper alternative.

The regulator chips we used were linear technology chips, LT1529-5 and LT1529-3.3. These chips work well because they have a low dropout voltage as the current increase. This can be seen in Figure 2 below taken from the LT1529 datasheet:

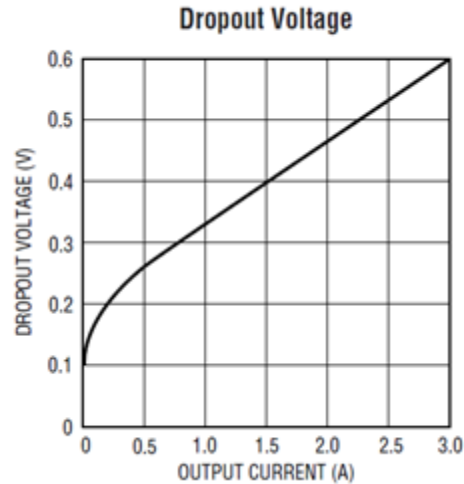


Figure 1. Dropout Voltage of LT1529 Chips[6]

With the components being powered on the base station and the robot, the system never needs to draw more than about 1A of current to power everything. Therefore, we see a minimum output voltage around 4.7V or 3.0V at any given time. This is still enough to power all the circuits individually without causing problems. The final main power distribution circuits for the robot and the base station are shown in Appendix C.

## 2.2 H-Bridge Circuit

The H-Bridge circuits that we initially designed did not work exactly as intended. The first design change made was adding a coupling capacitor of 470 $\mu$ F at the input of the battery. This was used to eliminate any noise spikes that were occurring from the irregularities of the battery.

During testing, the circuit did not work with the initial design. The LT1160 chip was not outputting 12V at the gate outputs when it was receiving the correct input voltage. The reason for this was that the Boost pin on the LT1160 chip was not being powered with  $V_{CC}$ . In addition, the  $T_{SOURCE}$  pin was not grounded. Without these changes, the output voltage did not have a reference voltage to match and was therefore only outputting 0V. The final design of the H-bridge can be seen in its entirety in Figure 2 below.

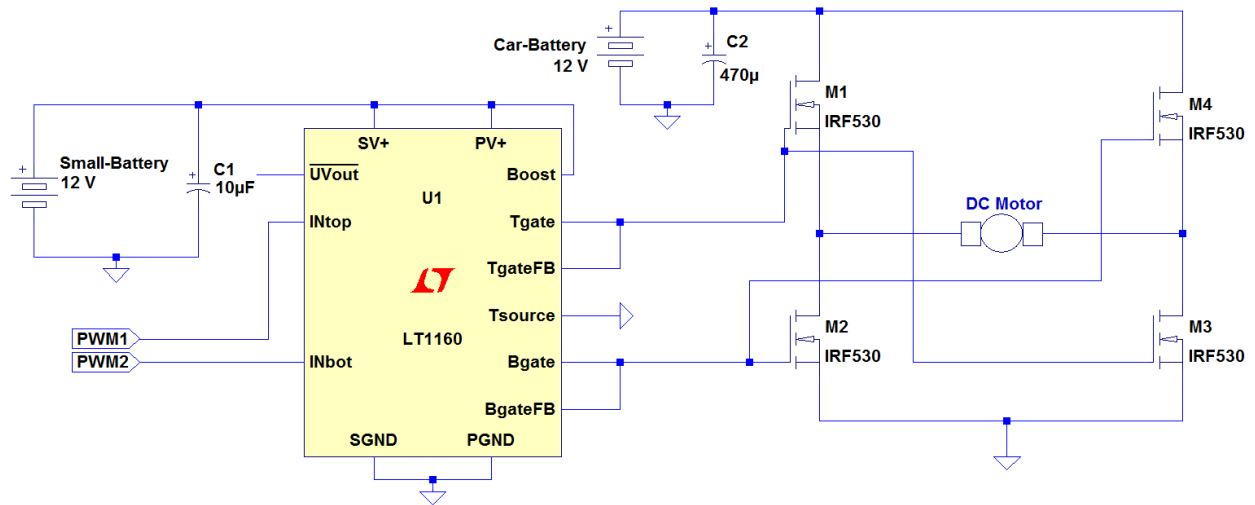


Figure 2. Final H-Bridge Gate Driver Circuit

Another significant change we made was the gate driver MOSFETs. Due to availability, we went with the IRF530Ns which are NMOS Power MOSFETs that can handle 100V up to 22A. In addition, the  $V_{GS}$  only required a minimum of 10V to turn on which works well since the LT1160 chip outputs around 12V. Another reason these MOSFETs were a great choice is because the  $R_{DS,ON} = 0.064\Omega$ . This implies that the maximum power lost in a MOSFET would be:

$$P_{LOSS,MAX} = (R_{DS,ON}) * (I_{ON})^2$$

Equation 3. Maximum Power Lost in a MOSFET

Since the main motors draw a maximum of 2.1A, the  $P_{LOSS,MAX}$  is about 0.28224 A. During the testing process, the motors drew around 0.5A while running. Since the power loss was minimal, it was not necessary to put heat sinks on any of the MOSFETs because they did not heat up at all.

The LT1160 chips were a great choice because they had built in circuit protection that prohibited the circuit from shorting itself. From Figure 4 below, the top waveform is that of an input signal into INtop. Another constant 5V signal was being inputted into INbottom (not shown in Figure 4). Whenever the voltage is HIGH in the top waveform, the bottom waveform (which is the output at Bgate) drops down to a LOW never letting both the inputs be a HIGH value. In addition, the gate driver does a good job of filtering any noisy inputs with large spikes as can be seen from the comparisons of the first and second waveforms.



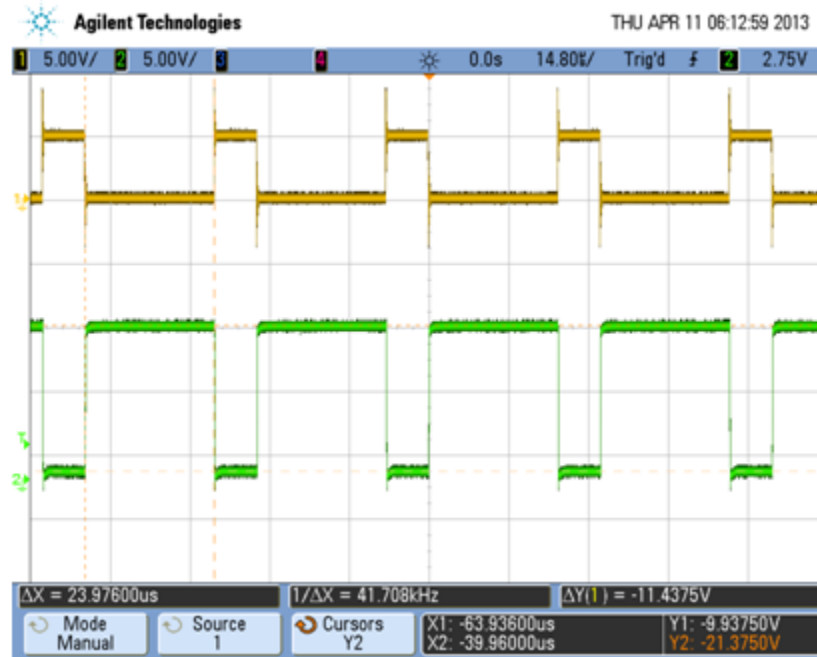


Figure 4. H-Bridge Circuit Protection Example

The rise time and the fall time of the output voltage were not really an issue for our system because as another safety feature, we never let the motors switch direction instantly. This was done by setting the code on the Arduino to have a one second delay before reversing the rotational movements of the motors.

The reason this setup is using two different batteries is because we wanted the system to power all of the electronics first. After that, the motors would be powered up second. This is done to ensure that the gate drivers would not produce a voltage spike in the MOSFETs, causing them to accidentally turn on.

## 2.3 Wireless Communication

The initial wireless communication design included two circuits, a transmitter with an encoder and a receiver with a decoder which can be seen in the appendix.

The decoder and encoder were meant to break apart data into bytes. From these bytes, we could capture and send individual bits through the digital input and digital output pins on the Arduino. This initial design proved to be unnecessary because the TXM-900-HP3 transmitter and the RXM-900-HP3 receiver can directly transmit serial data streams through the data line inputs on the chips themselves. In addition, this direct transmission allows the system to be robust by making the Arduino at the base station and the Arduino on the robot act as one system. With this setup, only one Arduino is executing commands at a time while the other one is on standby waiting for a wireless signal to be generated.

The wireless transmission also uses a coded system. There are a set of three byte characters that need to be seen by the Arduino in order for the system to begin recording the data. At the end of the transmission, there would be another set of three byte characters that indicate when the transmission is over. This works well with our setup because the Arduino keeps transmitting the data over and over again in a loop so it ensures that at some point the transmission will get through to the system. Within the code, there are placeholder values that are looked for in case the transmissions are broken apart into sections. A small portion of this code can be seen in Appendix D.4.

The only downside of the whole wireless communication system was the distance it could transmit back and forth. After about 15 feet of transmission, the received strength indicator (RSSI) was not large enough to be able to receive the data being transmitted from Arduino to Arduino. In future works, a directional antenna would have been a better choice because it can transmit farther.

## 2.4 Microcontrollers

The microcontrollers we are using are two Arduino Mega 2560 and one Arduino Ethernet. We chose to use the Arduino Mega because it has four serial communication ports. It was important that our microcontrollers had at least three serial communication ports so that they could transmit and receive data through our wireless connection. We could have used other brands of microcontrollers that had enough serial communication ports, but the Arduino offers the easiest programming platform. This is because the Arduino uses open source software and there are many tutorials available online. The Arduino Ethernet will be discussed in more detail in the internet section.

The Arduino Mega for the base station controls the flow of water to the sprinkler by turning on and off the hose actuator. It also activates the robot to start the soil measurement program. The Arduino Mega for the robot reads in the GPS data and navigates the robot. Once the robot is at a measurement site, the Arduino inserts the soil probe into the ground and records the soil resistance.

## 2.5 GPS

A Venus GPS device is responsible for keeping track of the position and direction of the sprinkler robot. The Venus GPS updates the microcontroller of the sprinkler robot with the coordinate values, and then the microcontroller transmits the coordinate values to the base station. This allows the microcontroller at the base station to send the robot the next command.

The GPS is in direct contact with the microcontroller so it does not need to send signals through the transmitter chips. This setup allows the microcontroller to receive the relevant data

to operate a real-time system that updates the robots position continuously so it can keep track of its position.

The reason we used this GPS unit was because it had an accuracy of 2.5 meters and supported Wide Area Augmentation System (WAAS). WAAS is a system that improves GPS accuracy by using reference stations in the vicinity.

A National Marine Electronics Association (NMEA) message is the output of the Venus GPS device, which is a binary message that will be sent to the Arduino. The way this works is shown in Appendix F.

## 2.6 Motors

The two drive motors are both 1LPV6 motors that run on 12V and a maximum of 2.1 A. These motors were used to control the robot's movement. An H-Bridge circuit is used to make the motors spin clockwise or counter clockwise. The gate drivers and the H-Bridge Circuits are used for the overall control of the motors. Please refer to the gate drivers and H-Bridge portion for functionality.

For the soil probe motor, we have a Pittman GM9434G807. On it, we have a potentiometer so that we would know when the motor is moving. If we are sending power to this motor and it is not moving, the Arduino will turn off power to the soil probe motor so that the measurement system is not damaged.

## 2.7 Soil Probe

The soil probe is used to measure the soil resistance. It uses two prongs which are inserted vertically into the soil by the soil probe motor. Then by using voltage divider rule, it takes a measurement of the soil resistance. This is accomplished by comparing the voltage drop across a 1M $\Omega$  reference resistor to the voltage drop across the soil. The Arduino Mega 2560 does this by reading in an analog value as show in Figure 5.

$$\text{Soil Resistance } (\Omega) = \frac{\text{Analog In}(V)}{5(V) + \text{Analog In}(V)} \times \text{Reference Resistor}(\Omega)$$

Equation 4. Voltage Divider Rule

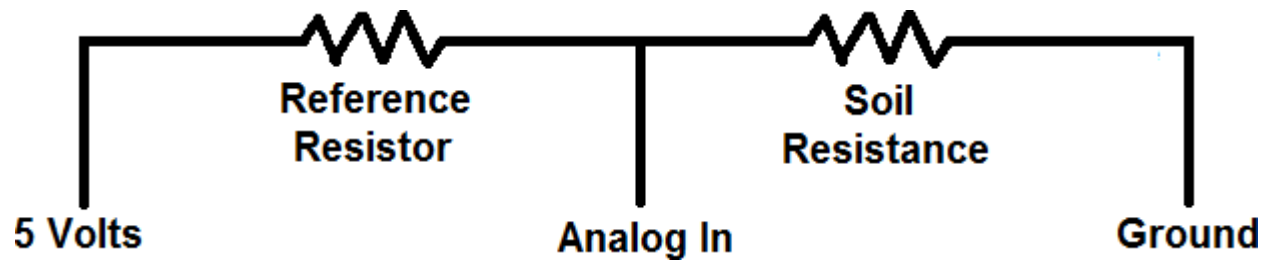


Figure 5. Soil Probe Circuit.

## 2.8 Internet

The internet communication is used to gather a 24-hour weather forecast. This is done using an Arduino Ethernet. Another alternative we could have used would have been an Ethernet shield for an Arduino Mega 2560. However, some users reported having errors or bugs with this Ethernet shield so we decided to use the Arduino Ethernet for its reliability. The weather forecast is collected by the Arduino Ethernet online from forecast.weather.gov. Once it has the weather data, the Arduino reads the forecast for precipitation for the next 24 hours. Using this data, it decides whether or not to send the sprinkler robot out to check the soil.

## 2.9 Hose Actuator

The hose actuator's purpose is to control the flow of water from a water source to the robot. To power up the hose actuator we need 24 VAC $\pm$ 1% at 60 Hz  $\pm$  0.5%. To do this we use a power plug, a transformer, an n-channel power MOSFET, an operation amplifier, and a signal from the microcontroller (Arduino). Please refer to Figure 6 to see the hose actuator circuit.

The actuator that we are using is the Rain Bird JTV-100. It has a solenoid power requirement of 24 VAC with an inrush current of 0.30 A and a holding current of 0.19 A. It is designed for pressures between 15-150 pounds per square inch (psi) and for water flows between 1-30 gallons per minute (GPM).

The Tamura 3FD-348 transformer is used to drop the 120 VAC down to the required 24 VAC. The reason we used this transformer was because when the output pins are connected in parallel, we can get the 120 VAC dropped to 24 VAC with RMS Amperage of 0.1 Amps. The output of the transformer has two pins that serve as the AC output, and we run one directly to the Hose Actuator, and the other to the gate of the power MOSFET that we use.

We use the LM 741 operational amplifier to multiply the Arduino logic HIGH by two, which comes out to be 11.28VDC. The output of the operational amplifier was fed into the GATE of the power MOSFET to open and close the circuit with the power MOSFET.

The power MOSFET that we use is the IRF 530N. This is used as a switch to control the signal going to the hose actuator.

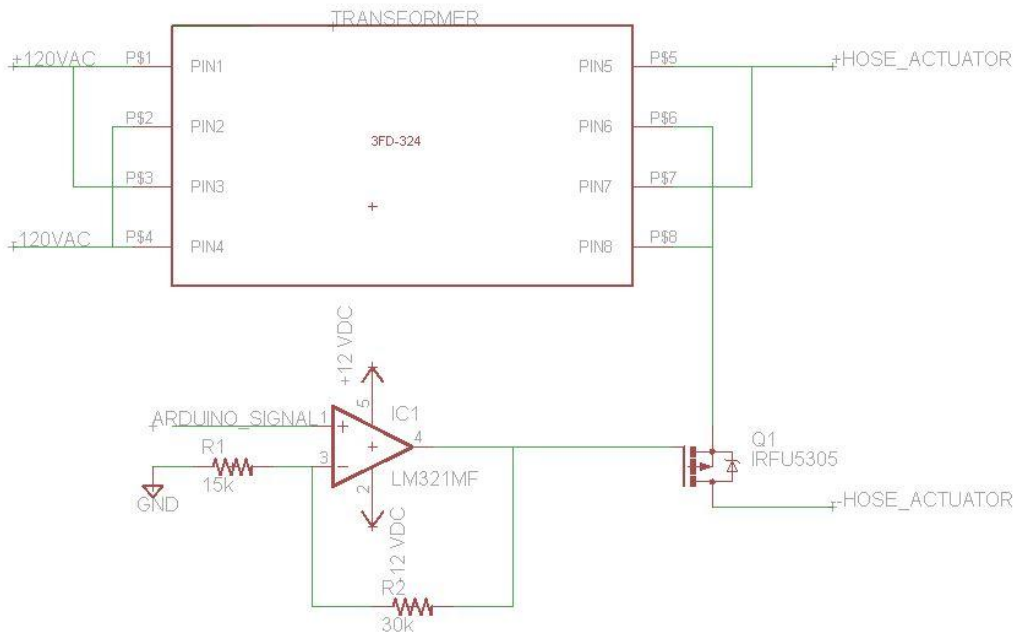


Figure 6. Hose Actuator Circuit

## 2.10 Battery

The initial system was composed of three batteries. One battery was used on the base station to power the transmitter and receiver. The battery being used for this is a SLA7-12 battery rated at 12V and 7Ah (amp hours). The maximum current ever being drawn by this setup is 38.0 mA which means the battery we had on the base station never ran into issues.

On the robot, two different batteries are being used. The first battery is a SLA7-12 battery rated at 12 and 7Ah that powered the transmitter, receiver, gate drivers, Arduino, and GPS. The maximum current draw of this setup at any one time is around 677.56 mA. Although this is much greater than the base station system, the system never operated at such a high current because not every component was running at the same time. The second battery on the robot was initially an Xtreme Plus battery that was rated for 12V and 35 Ah. This battery failed to charge from 5V to 12V so we bought a new Die-Hard battery that was rated for 12V and 42 Ah. This battery works just as well, if not better, because it is able to last longer when powering the system. The maximum current being drawn by the motors ever should only be 4.21A so the Die-Hard battery would have no problem running the motors for a long time.

An unexpected issue that arose with the batteries was spiking of the voltage as a result of plugging and unplugging the battery. During the testing aspect of the course, we decided to just plug and unplug the batteries over and over again when they were not in use or when we were testing. This resulted in sparking of the battery and, in turn, sharp spikes in the voltage output.

As a result, one of the drive motors had an issue where it failed and began drawing more current than it should have.

In addition, a mechanical issue arose where the Die-Hard battery weighed down the car to the point where it couldn't move on the grass. To solve this issue, an attempt was made to use another SLA7-12 battery which was much smaller and light weight but did not last as long as the larger battery. This did not completely fix the issue because the frame of the car was still too heavy to keep the car moving on the grass for long periods of time.

## 2.11 Power Outlet

We did not have to design the power outlet, but we did need the power outlet to power up the computer and the hose actuator at the base station. We specifically needed the power outlet to provide  $120\text{VAC} \pm 5\%$  at  $60\text{ Hz} \pm 0.5\%$ .

The power outlet is a National Electrical Manufacturers Association (NMEA) 5-15 (15A/125V grounded) (Type B). It is connected to the transformer and the base station computer through a grounded power cord.

## 3. Requirements and Verification

### 3.1 Power Electronics

The initial testing of the low-drop out regulators was done with a high impedance at the output. This produced an output of  $5V \pm 0.5\%$ . Once all the modules were running together, the output voltage still produced the same  $5V \pm 0.5\%$  and  $3.3 \pm 0.5\%$ . All of the requirements were met for the low-drop regulators.

### 3.2 H-Bridge Circuit

Initially, just powering the  $SV+$  and  $PV+$  pins was not enough. The boost pin also needed to be powered with 12V and the  $T_{SOURCE}$  need to be grounded. Once that was done, the system was powered correctly. PWM signals were sent from the Arduino to the inputs on the gate drivers and the output was the correct value.

The frequencies of the PWM signals at the output matched those of the input. The duty ratios for the signals did not change in any way. The  $V_{GS}$  for the MOSFETs required 10V to turn on and the circuit was providing 12V which was more than enough. All requirements were met for the H-Bridge circuit.

### 3.3 Wireless Communication

The initial design of the wireless communication included an encoder and decoder chip that would be used to break apart data streams into bytes so individual bits could be captured along the digital inputs of the Arduino. This proved to complicate the system and instead, a more robust approach was used where transmissions would be sent directly across the data stream.

Since we took this different approach, we did not need to verify that the Arduino was sending the correct data type to the encoder. We also did not need to worry about the decoder chip receiving the correct data and sending the correct data to the Arduino. Instead, the only necessary requirements we need to verify was one and three from the previous requirements and verifications table in Appendix B.

The wireless communication was sending and receiving the correct data in the form of a byte string sent to and from the Arduino on the base station and the Arduino on the robot. Two different transmission channels were used, one at 903.37 MHz and one at 921.37 MHz. This allowed two-way communication between both the Arduinos.

Initially a test was done using a hard wired connection to verify that the correct data was being transmitted between Arduinos on their serial communication ports. Both Arduinos were set

to 9600 bits per second rate (BAUD) which falls in the range of the acceptable BAUD rate for the receiver and transmitter. Once that was done, the receiver and transmitter were inserted in between the two Arduinos and the same test was performed. The verifications were met and the Arduinos were communicating wirelessly between each other.

The only issue that occurred here was the fact that the wireless communication would only work up to 15 feet. After that, the received strength indicator (RSSI) was too low to receive a strong enough signal. This issue could have been prevented if we had used directional antennas instead.

### **3.4 Microcontrollers**

We were able to verify that both of the Arduino Megas worked. We were able to successfully power the Arduinos using the voltage we specified. We could send information wirelessly between the Arduinos. An image of this working can be found in Appendix E.3. The code that we used for these microcontrollers can be found in Appendix D.1 and D.2

The Ethernet Arduino was also successfully verified. By running the program found in Appendix D.3, we were able to make a 24-hour forecast for precipitation. Using that information, we were then able to decide to activate the sprinkler robot or not. A picture of this program running can be found in Appendix E.1.

### **3.5 GPS**

The first way we tested the GPS receiver, was that we used the GPS antennas that are on top of Everitt Laboratory, and the coordinates were less than 1 meter off. This may have turned out to be this accurate because the antennas on the roof of Everitt Laboratory are state of the art GPS antennas.

The other way we tested the GPS was we cross verified the coordinates from Google maps with the coordinates that the GPS receiver was getting. We found that the coordinates the GPS was giving us was within a 2.5 meter deviation of what the actual coordinate was. This was done with the GPS unit installed on the robot, and the robot was moving around.

We were pleased with our results because it allowed for us to work on the robot's navigation.

### **3.6 Motors**

When we tested the drive motors, we connected them directly to the battery to see them move forward. Then we reversed the polarity of the robots connection to the battery and saw them move in reverse.



We did not use the potentiometers on the drive motors because our GPS navigation covered that functionality. In addition, we were able to determine that the wheels would make a full turn in 6.5 seconds. We were able to use that data for navigational purposes.

We were able to adjust the speed of the wheels with pulse width modulation, but we needed to run the motors at full speed because of how heavy the chassis of the robot was.

For the final demonstration the left drive motor failed to operate. We believe that this could have been caused by a voltage spike from our battery. We then added 3A circuit breakers to ensure that the other motors were protected from voltage spikes coming out of the battery.

For the soil probe motor, we determined that we did not need to know the position of the probe. Instead we designed a safety system that made sure if the motor was powered and the slide was not moving, it would shut off power to the motor. This allowed us to initialize the probe to a base position before inserting it into the ground. In addition, this protected the soil probe from being damaged.

### 3.7 Soil Probe

The soil probe was able to successfully measure the resistance of the soil. The error for the measurement was also within our 5% specification for the range of soils that we tested. We verified this by testing resistors of known value to the resistance measured by the soil probe. These measurements can be found in Appendix G. Pictures of the program the soil probe working can be found in Appendix E.

### 3.8 Internet

We were able to verify that the internet worked by plugging an Ethernet cord into our computer and accessing a website. This was done every time before we set up the Ethernet Arduino to make sure that the Ethernet port was working.

### 3.9 Hose Actuator

The first way we tested the hose actuator was by supplying a 5 and 24 VAC signal at 60 Hz to see if the hose actuator would turn on at 24 VAC and stay off with the 5 VAC signal. Those tests passed and we then moved on with trying to do this with circuitry.

An extra way we tested the hose actuator, that was not part of our initial requirements, was with the circuitry that was going to be used to control the hose actuator. The way we tested the hose actuator was that we first attached a  $1M\Omega$  resistor to serve as a load at the output of the transformer, and sent signals from the Arduino to turn on the switch and measured the output voltage. When the Arduino would send a signal to turn off the switch, the voltage read was 7.8

VAC. When the Arduino would send a signal to turn on the switch, the voltage reading was 23.89 VAC. The tests lead us to being able to control the hose actuator successfully.

### **3.10 Battery**

When checking the batteries with a multimeter, the two small batteries were both outputting  $12\text{VDC} \pm 5\%$  which is good enough to power all the components need. However, the initial large battery being used to power the motors on the robot failed to meet verification. It outputted an initial voltage of 5.87V and would not recharge to 12V.

Instead, we went out and bought a Die-Hard car battery that was rated for 12V and 42 Ah. This battery outputted  $12\text{VDC} \pm 5\%$  which was good enough to power all of our motors.

### **3.11 Power Outlet**

The power outlet was tested with a volt-meter and an oscilloscope. We found that the voltage coming out of the wall power outlet was 125.54VAC at 60Hz with both tests, so this was a successful verification.

## 4. Cost Analysis

This section covers the cost of our time designing this project in addition to the cost of materials used. For labor costs, we estimated our total time spent working on this project. It took us a combined 540 hours to complete this project which puts the total labor costs at \$50,625. For the parts cost, we have two different values. The total combined price is the cost of purchasing all of the components that we used. The total personal cost is how much was actually spent on parts that were not provided to us.

### 4.1 Labor Cost

|       | Hourly Salary | Hours per Week | Total Weeks | Total Hours | Cost     | x2.5     |
|-------|---------------|----------------|-------------|-------------|----------|----------|
| Denis | \$37.50       | 15             | 12          | 180         | \$6,750  | \$16,875 |
| Kevin | \$37.50       | 15             | 12          | 180         | \$6,750  | \$16,875 |
| Jose  | \$37.50       | 15             | 12          | 180         | \$6,750  | \$16,875 |
| Total |               |                |             |             | \$20,250 | \$50,625 |

## 4.2 Parts Cost

| Name                          | Part Number                  | Quantity | Unit Price | Combined Price | Personal Cost |
|-------------------------------|------------------------------|----------|------------|----------------|---------------|
| Arduino                       | Arduino Mega                 | 2        | \$56.08    | \$112.16       | \$0.00        |
| Arduino                       | Arduino Ethernet             | 1        | \$59.95    | \$59.95        | \$59.95       |
| Copper Breadboards            |                              | 10       | \$1.00     | \$10.00        | \$10.00       |
| Dayton DC Motors              | 1LPV6A                       | 2        | \$230.00   | \$460.00       | \$0.00        |
| Pittman Gear-Motor            | GM9434G807                   | 1        | \$100.00   | \$100.00       | \$0.00        |
| Linx Transmitter Chip         | RXM-900-HP3-SPO              | 2        | \$31.94    | \$63.88        | \$63.88       |
| Linx Receiver Chip            | TXM-900-HP3SPO-ND            | 2        | \$39.24    | \$78.48        | \$78.48       |
| Radio Frequency Antenna       | ANT-916-CW-RAH               | 4        | \$7.06     | \$28.24        | \$28.24       |
| LDO 3.3 VDC                   | LT1529CT-3.3#PBF-ND          | 1        | 6.78       | \$6.78         | \$0.00        |
| LDO 5.0 VDC                   | LT1529CT-5#PBF-ND            | 2        | 6.78       | \$13.56        | \$0.00        |
| Retractable Garden Hose       | Water Hose With Reel         | 1        | \$99.99    | \$99.99        | \$99.99       |
| 360° Pulsating Sprinkler      | 9536CH                       | 1        | \$8.97     | \$8.97         | \$8.97        |
| Tapped FGHxMIP Adapter        | LFA-679                      | 2        | \$3.97     | \$7.94         | \$7.94        |
| Watts Male Hose x FPT Adapter | A-669                        | 2        | \$3.97     | \$7.94         | \$7.94        |
| Rain Bird Sprinkler Valve     | JTV-100                      | 1        | \$13.33    | \$13.33        | \$13.33       |
| 6 ft. Hose Reel Leader Hose   | 887 6                        | 1        | \$7.47     | \$7.47         | \$7.47        |
| Two 12" Wheels                |                              | 2        | \$12.50    | \$25.00        | \$0.00        |
| Venus GPS                     | GPS-11058                    | 1        | \$49.95    | \$49.95        | \$49.95       |
| Venus GPS Antenna             | GPS-00464                    | 1        | \$12.95    | \$12.95        | \$12.95       |
| Capacitor 22 μF               | UPR1V220MAH                  | 3        | \$0.05     | \$0.15         | \$0.00        |
| Power Mosfet                  | IRF530N                      | 13       | \$2.04     | \$26.52        | \$0.00        |
| Gate Driver                   | LT1160                       | 3        | \$2.68     | \$8.04         | \$0.00        |
| Capacitor 10 μF               | P10425TB-ND                  | 3        | \$0.04     | \$0.12         | \$0.00        |
| Capacitor 470 μF              | EGPA101ELL201ML20S           | 2        | \$2.89     | \$5.78         | \$0.00        |
| Resistors 100kΩ               |                              | 8        | \$0.01     | \$0.08         | \$0.00        |
| Dipswitch                     | SDA04H0SBR                   | 4        | \$1.24     | \$4.96         | \$0.00        |
| Resistors 200Ω                |                              | 2        | \$0.01     | \$0.02         | \$0.00        |
| Resistors 1MΩ                 |                              | 1        | \$0.01     | \$0.01         | \$0.00        |
| Transformer                   | Tamura 3FD-348               | 1        | \$5.05     | \$5.05         | \$5.05        |
| USB Connectors                | Mediabridge-Hi-Speed USB 2.0 | 2        | \$5.00     | \$10.00        | \$0.00        |
| 12 V, 7Ah Battery             | SLA7-12                      | 2        | \$15.00    | \$30.00        | \$0.00        |
| 12 V, 42Ah Battery            | Die Hard                     | 1        | \$60.00    | \$60.00        | \$0.00        |
| Inline Fuses                  |                              | 3        | \$4.00     | \$12.00        | \$12.00       |
| 3Amp Circuit Breakers         |                              | 3        | \$2.93     | \$8.79         | \$0.00        |
| Operational Amplifier         | LM-741                       | 1        | \$0.70     | \$0.70         | \$0.00        |
| TOTAL                         |                              |          |            | \$1,338.81     | \$466.14      |

## 4.3 Grand Total

### Grand Total:

$$\$1,338.81 + \$50,625.00 = \$51,963.81$$

## 5. Conclusions

The Smart Sprinkler Robot System turned out to be a great project. Not only were we able to get every module to function correctly, but we were successful with getting all the modules to work together.

### 5.1 Accomplishments

We are able to gather weather information from the internet and analyze that information to determine if the robot should go out and start probing. The information to and from the base station and robot is sent via wireless communications. The robot is able to navigate to a given area with its GPS navigation.

The robot has a water hose attached to it to provide water to the sprinkler that is attached to the top of it. We have a hose actuator that controls the flow of water to the sprinkler so we can water a lawn when the robot's soil measures mandate for it.

All modules are powered through power electronics, which powers up by a car battery and a smaller battery.

### 5.2 Uncertainties

We still have a number of uncertainties regarding our finished project. The first is how well the Smart Sprinkler Robot would perform in a national market. The second is if the size of the robot would be a deterrent for people to have it in their lawn. Lastly, it is uncertain to know how well the Arduino will run for long periods of time.

### 5.3 Ethical Considerations

The Smart Sprinkler Robot System is a product that is able to do all of the details listed in the benefits and features. In order to make this robot marketable, it has to comply with the IEEE Code of Ethics. Of the ten that IEEE has listed, the first code is the one that applies to this project, which is as follows:

1. *To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment*

## 5.4 Future Work

The main problem that we had while working on this project was weight distribution. The chassis of the robot was too heavy for the motors to move the robot around. Using tracks instead of wheels could be a great solution to this as well. The wheels on the robot did not have enough traction to move the robot around without struggling on grass.

We definitely would like to use a relay at the batteries. The batteries would spark a bit at times when we would plug and unplug the batteries.

Another uncertainty is that we were not able to achieve wireless communication at far distances, so we would want to use better antennas in the future to make this Smart Sprinkler Robot System work in large watering areas.

## 6. References

- [1] Arduino Ethernet Schematic [Online]. Available: <http://arduino.cc/en/uploads/Main/arduino-ethernet-schematic.pdf>
- [2] Arduino Mega Schematic [Online]. Available: <http://arduino.cc/en/uploads/Main/arduino-mega-schematic.pdf>
- [3] *IEEE Code of Ethics* [Online]. Available: <http://www.ieee.org/about/corporate/governance/p7-8.html>
- [4] International Rectifier [Online]. Available: <http://www.irf.com/product-info/datasheets/data/irf6635.pdf>
- [5] Digikey [Online]. Available: <http://media.digikey.com/pdf/Data%20Sheets/NXP%20PDFs/IRF530N.pdf>
- [6] Linear Technology [Online]. Available: <http://www.linear.com/product/LT1529>
- [7] Linear Technology [Online]. Available: <http://www.linear.com/product/LT1160>
- [8] Linx Technologies [Online]. Available: <https://www.linxtechnologies.com/resources/data-guides/lical-enc-ms001.pdf>
- [9] Linx Technologies [Online]. Available: <https://www.linxtechnologies.com/resources/data-guides/ant-916-cw-rah.pdf>
- [10] Linx Technologies [Online]. Available: <https://www.linxtechnologies.com/resources/data-guides/lical-dec-ms001.pdf>
- [11] Linx Technologies [Online]. Available: <https://www.linxtechnologies.com/resources/data-guides/lical-enc-ms001.pdf>
- [12] Spark Fun Electronics [Online]. Available: [http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0003\\_v1.4.19.pdf](http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0003_v1.4.19.pdf)
- [13] Spark Fun Electronics [Online]. Available: [http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0003\\_v1.4.19.pdf](http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0003_v1.4.19.pdf)
- [14] Spark Fun Electronics [Online]. Available: [http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0008\\_v1.4.11-datalogging.pdf](http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0008_v1.4.11-datalogging.pdf)
- [15] Spark Fun Electronics [Online]. Available:
- [16] Spark Fun Electronics [Online]. Available: <http://php2.twinner.com.tw/files/onshine/ANT555-2006-NEW.pdf>

[17] Tamura [Online]. Available: <http://www.tamuracorp.com/clientuploads/pdfs/engineeringdocs/3FD-3XX.pdf>

[18] Texas Instruments [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm741.pdf>