ECE-445

Smart Sprinkler Robot System

Design Review

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

TA: Mustafa Mukadam

February 26th, 2013

Table of Contents

1 - Introduction 2
1.1 Statement of Purpose 2
2 - Objectives 3
2.1 Goals 3
2.2 Functions 3
2.3 Benefits 3
2.4 Features 3
3 - Design 4
3.1 Block Diagrams 4
3.2 Block Description – Base Station 5
3.3 Block Description – Sprinkler Robot 18
4 - Requirements and Verification 30
4.1 Testing Procedures – Base Station 30
4.2 Testing Procedures – Robot Sprinkler 36
4.3 Tolerance Analysis 45
5 - Cost Analysis and Schedule 46
5.1 Labor Cost 46
5.2 Parts Cost 47
5.3 Grand Total 47
5.4 Schedule 48
6 - Ethics and Safety 50
6.1 Ethics 50
6.2 Safety 50
7 - Appendix 51
7.1 References 51
7.2 Extra Figures 52

1 - Introduction

1.1 Statement of Purpose

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. We are excited about this project because this is an area of the market that is underdeveloped. It will also test our skills from many different areas of Electrical Engineering to put together this product.

2 - Objectives

2.1 Goals

The overall goal of this project is to design a sprinkler system that will determine when a lawn needs to be watered and how much it needs to be watered. Below are the sub-goals that will make this possible:

- 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

2.2 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.

2.3 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)

2.4 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
- Retractable garden hose prevents tangling

3 - Design

3.1 Block Diagrams

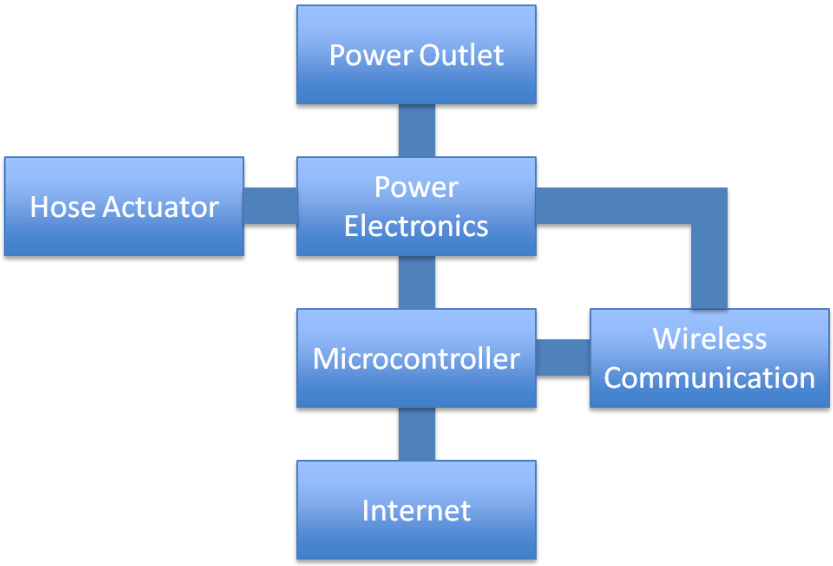

Figure 1. Base Station Module Block Diagram

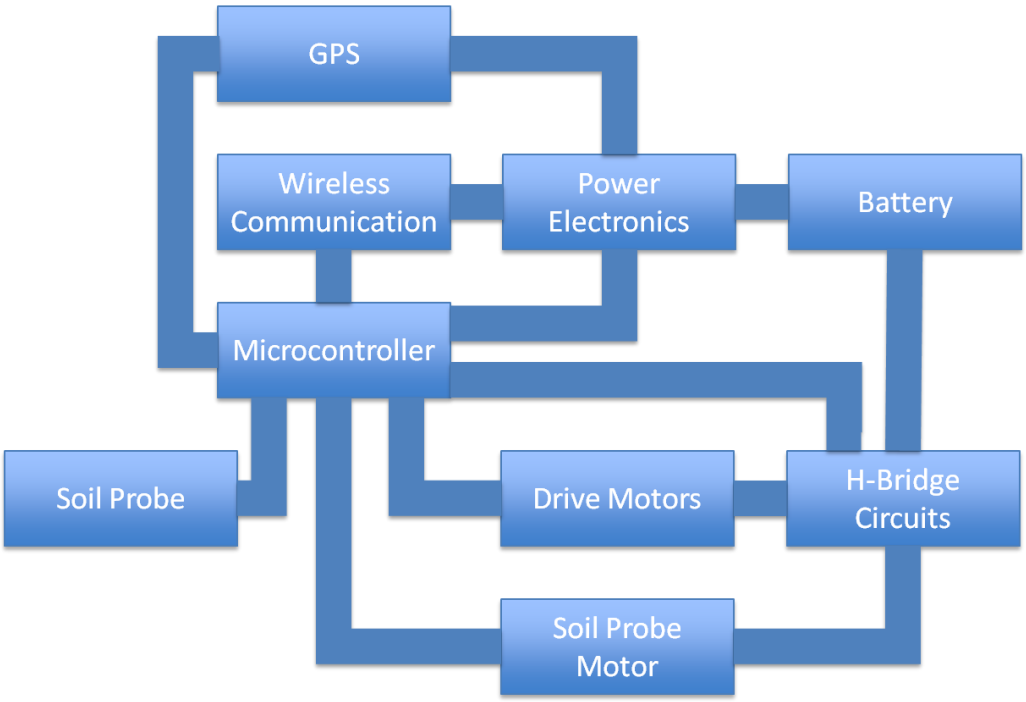

Figure 2. Sprinkler Robot Module Block Diagram

3.2 Block Description – Base Station

Power Outlet	
Inputs	None
Outputs	120V ± 5% at 60 Hz ± 0.5% to Power Electronics

The power outlet is a National Electrical Manufacturers Association (NEMA) 5-15 (15 A/125 V grounded) (Type B). It is connected to the power electronics through a grounded power cord.

Hose Actuator	
Inputs	24 VAC±1% at 60 Hz ± 0.5% from Power Electronics
Outputs	None

The hose actuator turns the water flow on and off to the sprinkler on top of the robot. This switching is controlled by the power electronics. It will do this switching whenever the microcontroller determines that the sprinkler should be switched on or off. The hose actuator is connected to an outdoor water wall outlet on one side of it and on the other side it connected to a garden hose which connects to the sprinkler robot.

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 psi and for flows between 1-30 GPM.

Figure 3. Rain Bird JTV-100

Power Electronics	
Inputs	120V ± 5% at 60 Hz ± 0.5% from Power Outlet 5 VDC ± 5% from Microcontroller
Outputs	24 VAC ± 1% at 60 Hz ± 0.5% to Hose Actuator 5 VDC ± 0.5% to Wireless Communications 5 VDC ± 0.5% to Microcontroller

The electronics used to power the actuator in the base station steps down the 120 VAC from the power outlet using a transformer which is connected to a switch that feeds a new 24 VAC to its input. The transformer being used is a 3FD-348 Tamura which takes a voltage of 115V and brings it down to 24V when the transformer terminals are connected in parallel (Figure 4). The switch used will be an IRFP240 which is rated for 200 V and 20 A. Whenever the V_{GS} is 2V or higher (up to 20V), the switch will turn on and allow water to flow through the actuator.

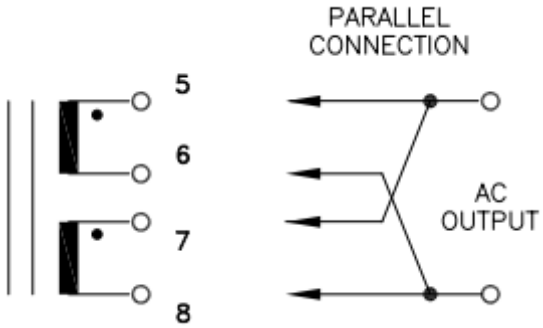

Figure 4. Parallel Connection

A simulation was done to test the voltage step down with the circuit below. The simulation shows that the circuit was able to transform a 120VAC signal into a 24VAC.

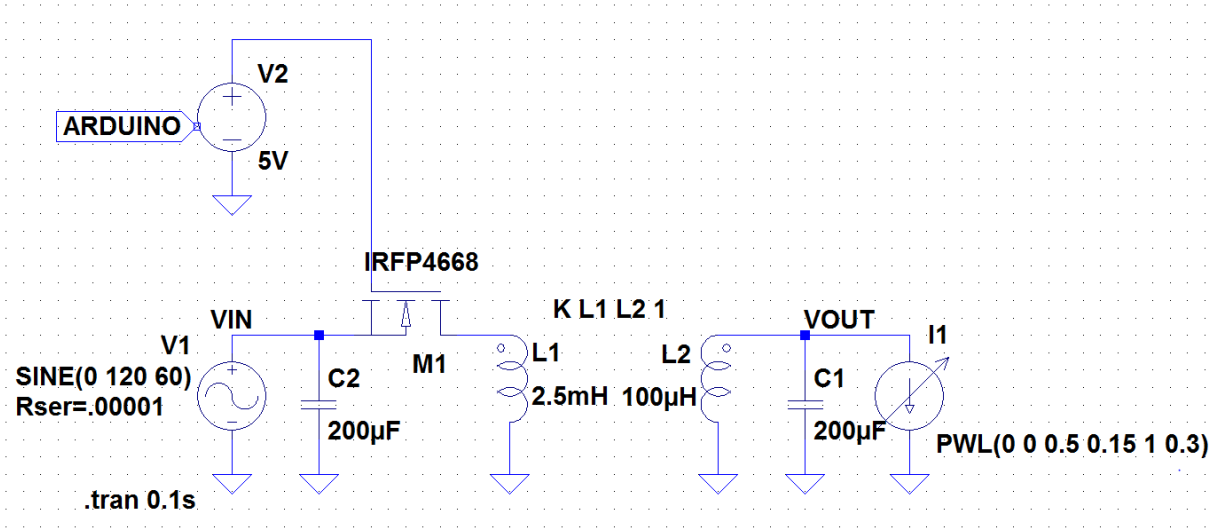

Figure 5. Test Circuit for Powering Hose Actuator

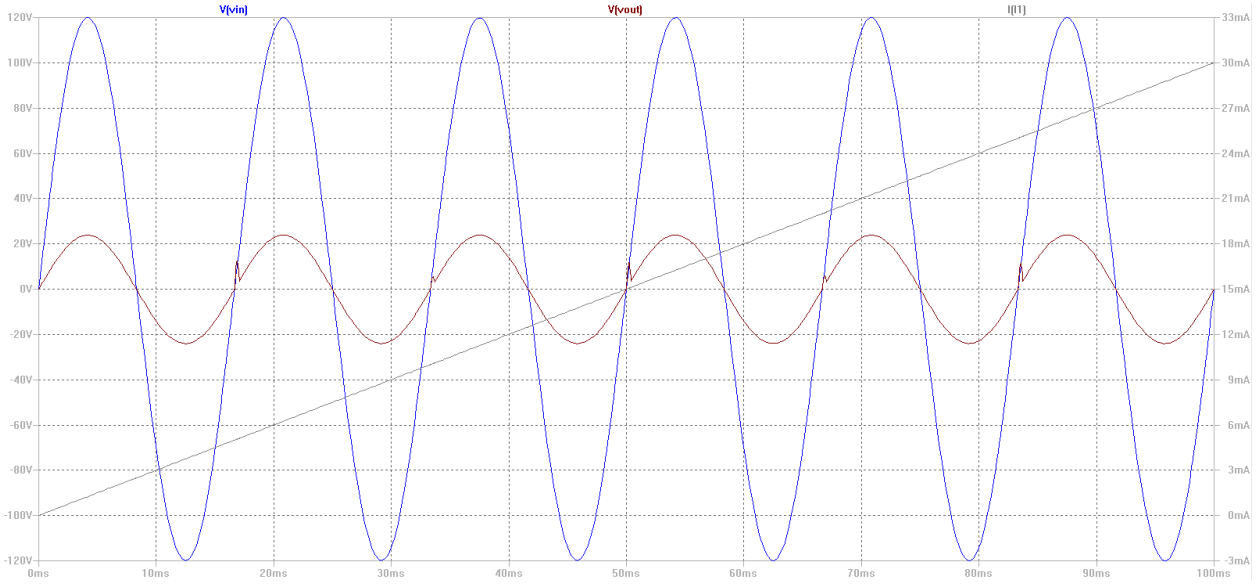

Figure 6. Simulation of Test Circuit for Powering Hose Actuator

The power distribution within the base station also requires a 12V, 7AH battery and USB connector from the computer to the Arduino. The 12V battery is fed into a low-dropout regulator which converts anything over 5.5V to 5V. Once the battery power drops below 5.5V, the LDO will turn off. This LDO powers the Linx receiver and transmitter circuits. The LT1529-5 takes in 12V and outputs 5V with a maximum of 5A (based on simulations). The TXM-900-HP3-XXX chip takes a maximum of 17.0 mA. The RXM-900-HP3-XXX chip takes a maximum of 21.0 mA. The Lical-ENC-MS001 and the Lical-DEC-MS001 both take a maximum of 780µA each at 5V operating voltage. Therefore, the LT1529-5 never has any issues powering these specific circuits over an extend period of time. The computer handles the direct power distribution to the Arduino which contains fuses in its USB connector. This ensures that a short circuit will not ruin the microcontroller while it is being powered.

The equation to calculate how much time the battery will last is as follows:

$$HoursOfOperation = \frac{Ah}{I}$$

Equation 1. Hours of Operation

The total maximum current will be 0.03956A for the battery at any given time which implies that it will be able to operate for 176.95 hours at a 7Ah load at that amperage. The LDO also consumes current but no more than 35mA maximum. With this added component, the battery will last for a minimum of 93.88 hours.

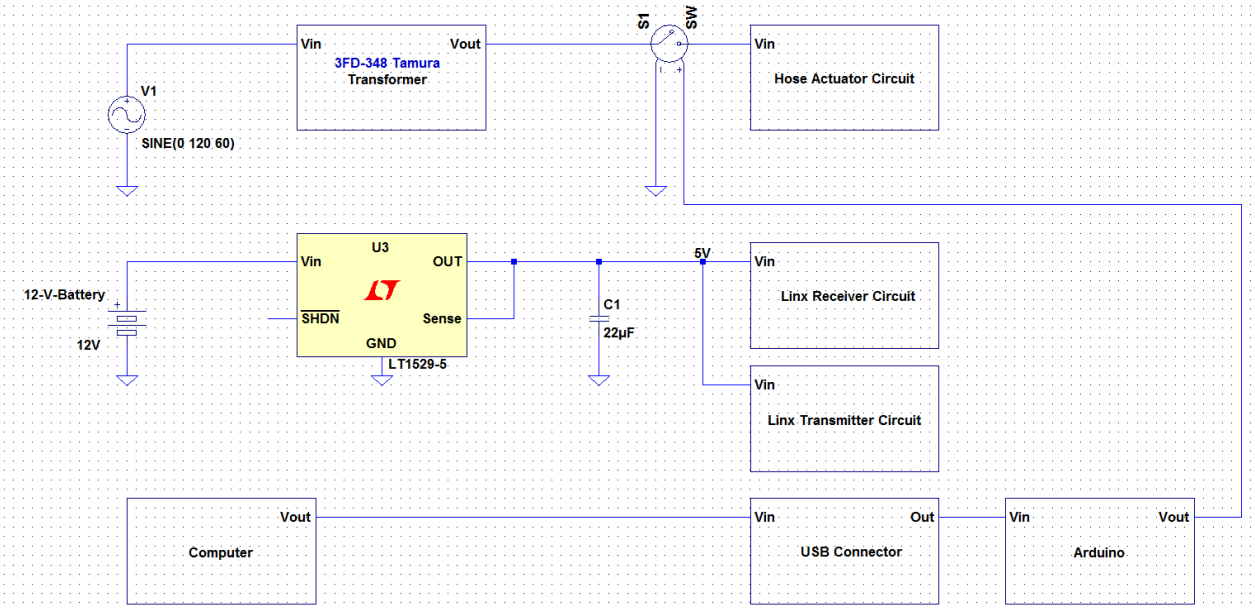

Figure 7. Power Distribution of Base Station

Microcontroller	
Inputs	5 VDC ± 0.5% from Power Electronics Digital Signal from Wireless Communication Digital Signal from Internet
Outputs	5 VDC ± 5% to Power Electronics Digital Signal to Wireless Communication Digital Signal to Internet

An Arduino Ethernet board will serve as the base station's microcontroller. The Arduino's purpose is to control the power electronics, wireless communication, and internet updates of the base station.

The Arduino board is powered by a USB connector. The USB connector itself is powered by a low-dropout regulator which supplies the connector with 5V±0.5. The board is built with internal fuses at the USB connector so it will blow the fuse if there is a short circuit.

The Arduino is connected to the internet through an Ethernet cable. From the internet, the Arduino gathers weather information to decide when to send the robot out to measure the soil moisture. If predictions call for rain or there has been rain recently, then the robot will not be sent out.

The base station's wireless communication will be transmitting signals to the sprinkler robot, and the Arduino will provide the appropriate signal that will need to be sent to the sprinkler robot. The base station's wireless communication will also receive signals from the robot, and the Arduino will analyze that data to be able to issue the next command to the sprinkler robot.

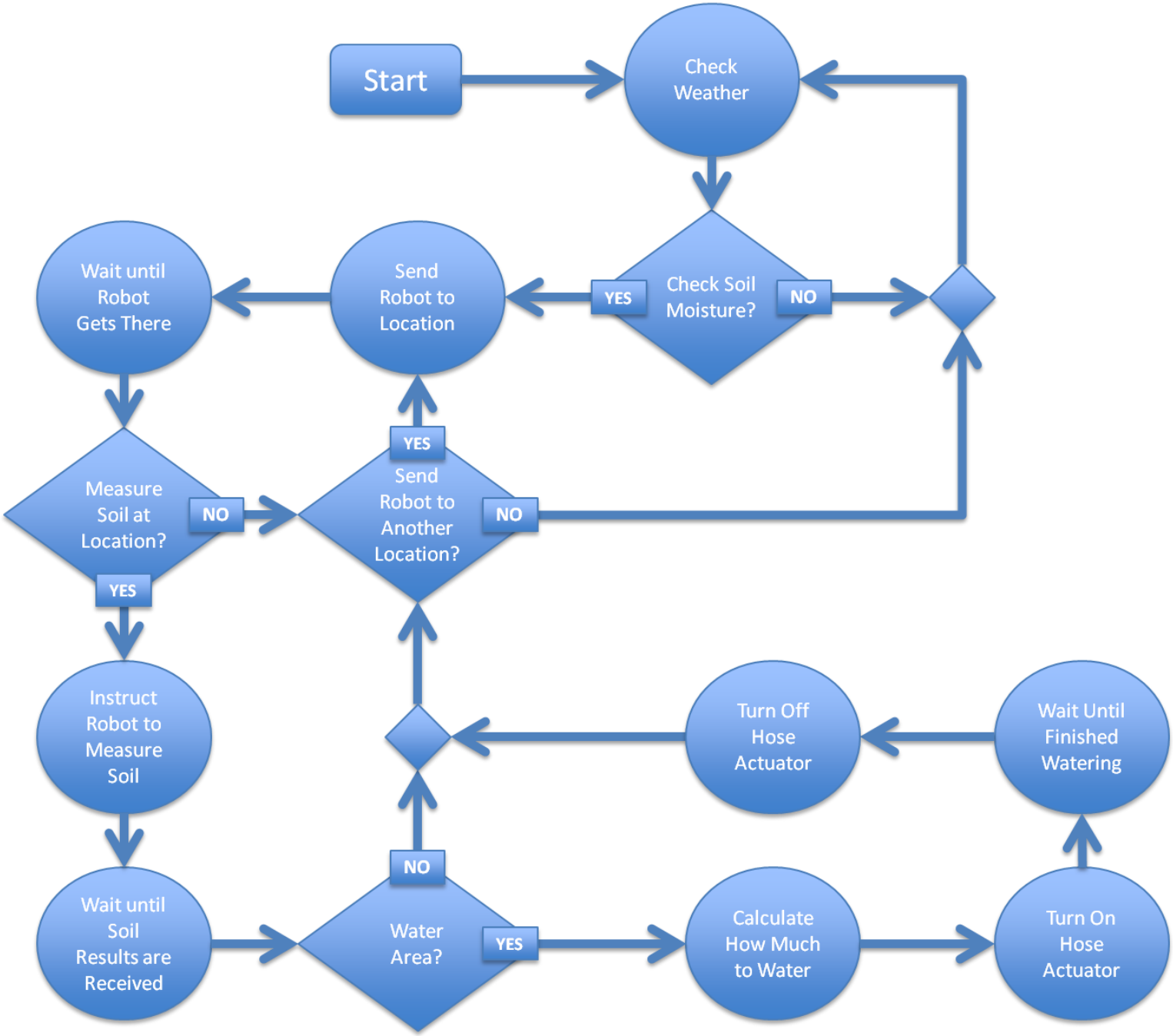

Figure 8. Base Station State Diagram

Wireless Communication	
Inputs	5 VDC ± 0.5% from Power Electronics Digital Signal from Microcontroller
Outputs	Digital Signal to Microcontroller

A transmitter and receiver circuit near the base station is responsible for sending and receiving information between the base station Arduino and the robot Arduino using serial communication. This feature is important because it allows the base station to update GPS locations and motor controls for the robot.

The receiver and transmitter chips being used operate at different frequencies because multiple channels are required for this application of wireless communication. Because of this, the Linx chips operate at a range of 902-928 MHz to allow for multiple channels. FCC regulations allow free use of this band so this works perfectly for our setup.

The following figures show preliminary simulations that the TXM-900-HP3SPO can transmit a single tone within the 902-928 MHz range and the RXM-900-HP3SPO can receive it. The two figures below were captured using a spectrum analyzer during a different semester. They were part of a final project that required making a receiver and transmitter that could transmit at different channels.

This time we are transmitting with the data stream feature provided by the Linx chips. A base design was found [8] on how to create a circuit that could transmit bytes of data by encoding them and then receive and decode them.

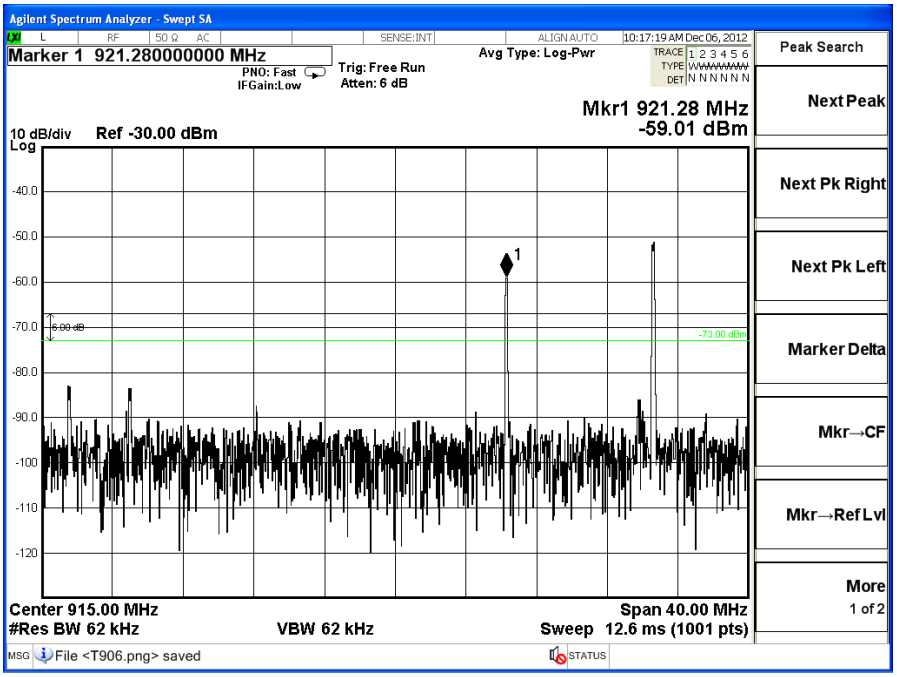

Figure 9. Single Tone Transmitted at 921.3 MHz

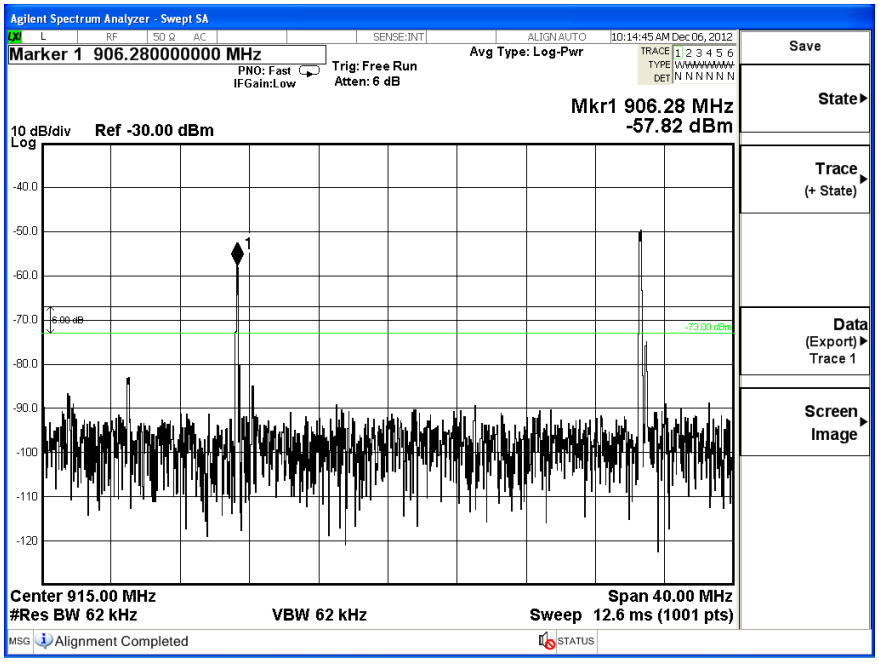

Figure 10. Single Tone Transmitted at 906.3 MHz

The first necessary circuit contains the transmitter, in addition to LICAL-ENC-MS001 encoder as can be seen below. The transmitter is a TXM-900-HP3-XXX series which contains 8 different channels to transmit along. A dip switch is used to select a channel based on the select bits: CS0, CS1, and CS2. The MODE is set to ground allowing only the 8 channels to be used for transmission. CTS will be left at floating which always allows data to be transmitted.

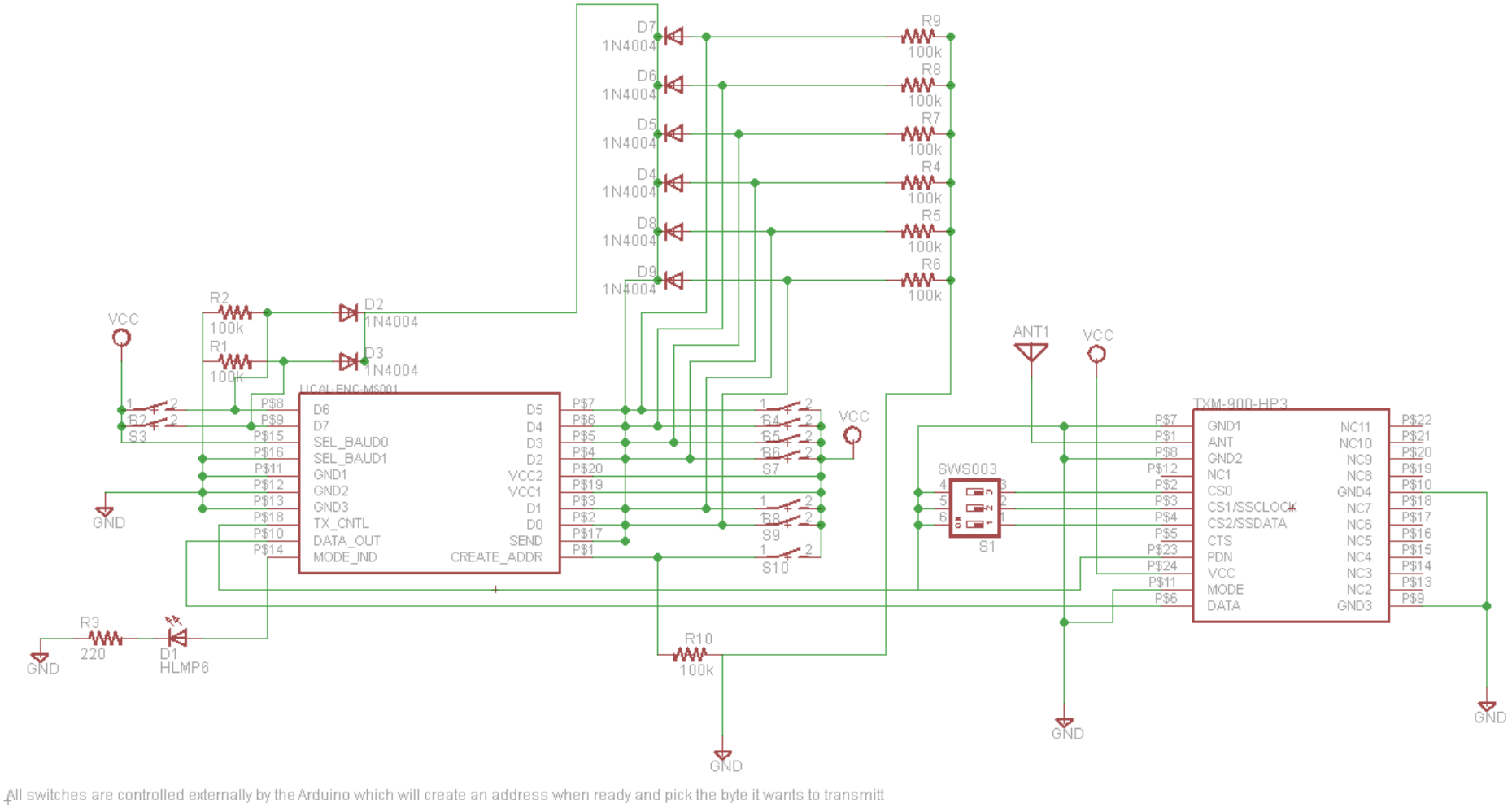

Figure 11. Transmitter Circuit with Encoder

The data bits are set using the 8 switches on the D0-D7 pin. The switches are controlled by the Arduino at the base station. The Arduino sets the switches to the correct byte it wants to transmit which then turns on the SEND pin causing the encoder to record the states of the data lines, retrieve a Code Word from the encoder, and send the information out as a data stream to the receiver. The CREATE_ADDR switch can be switched if a new Code Word is needed. For debugging purposes, an LED is set at the MODE_IND which turns on if a new Code Word is being created. The SEND mode is directly tied to the data input lines. The diodes are used to prevent voltages from activating all of the data lines at once. When SEND is high, it pulls the TX_CNTL pin high causing the TXM chip to turn on. It then sends the DATA_OUT that was stored into the transmitter.

Below is the flow chart of the basic operation of the encoder that indicates how it receives the data and sends the data packet to the transmitter.

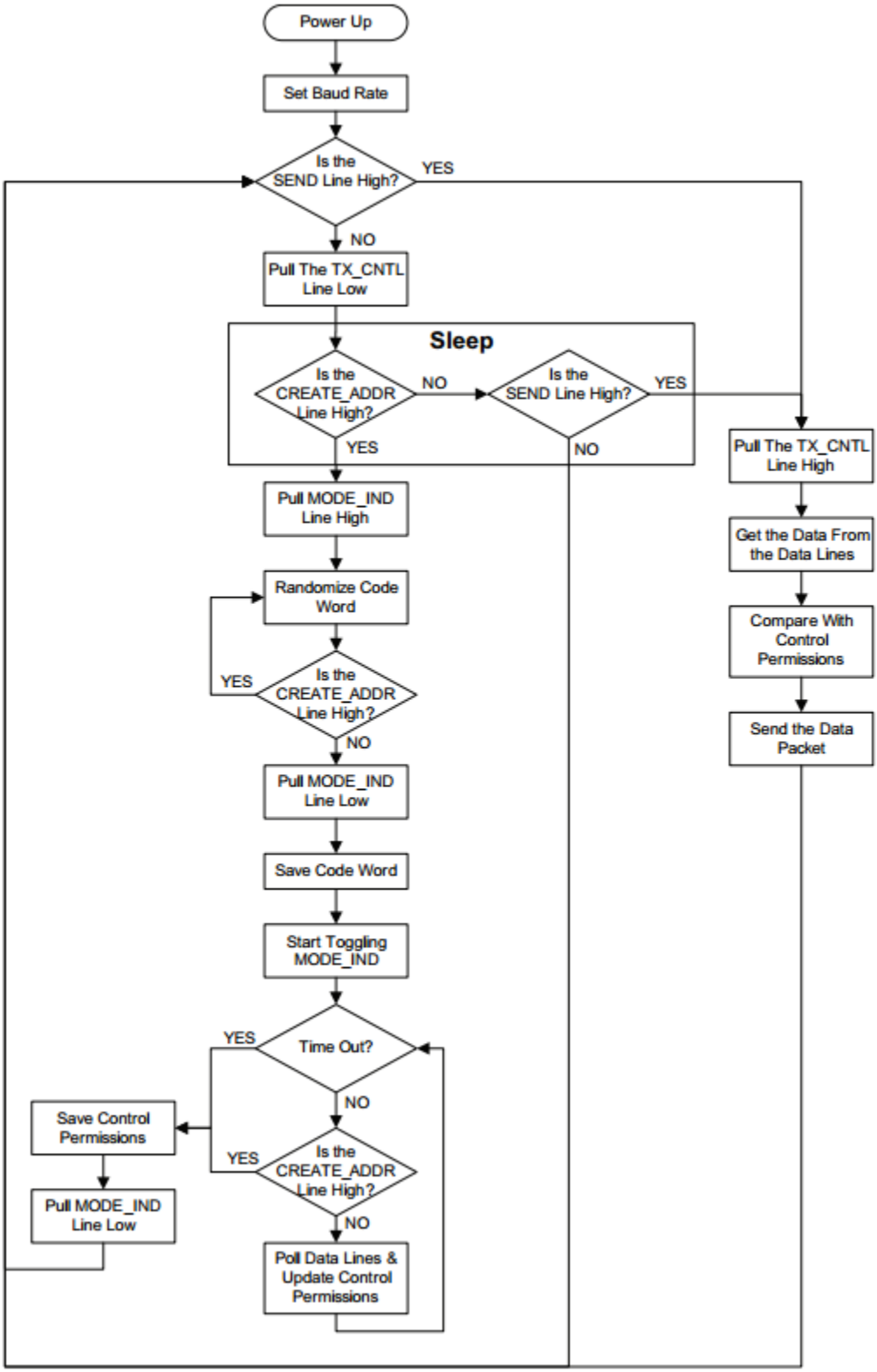

Figure 12. Encoder Flow Chart

The second circuit is a receiver with an LICAL-DEC-MS001 decoder which can be seen below. The receiver is a RXM-900-HP3-XXX series which contains 8 different channels to receive along. A dip switch is used to choose the correct channel to transmit along based on the three channel select bits, CS0, CS1, and CS2. It also contains a mode dipswitch. The MODE is set to ground allowing only the 8 channels to be used for transmission. RSSI is connected to the LEARN function on the decoder to indicate when data is received. Once the data is received, it is transmitted in a data stream through the DATA pin to the decoder's DATA_IN pin.

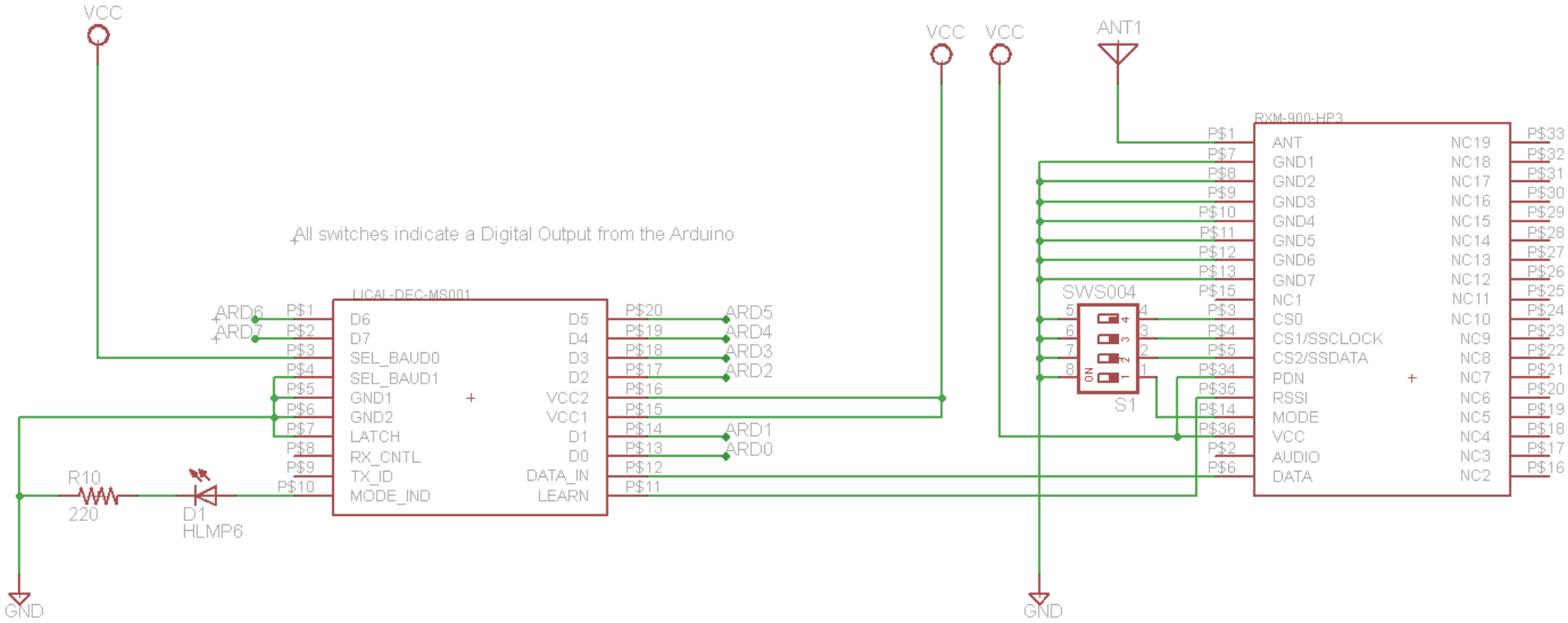

Figure 13. Receiver Circuit with Decoder

The decoder will have D0-D7 channels that are directly fed into the Arduino's digital inputs. The latch is set to low so the data outputs are unlatched. When a valid transmission is received and LEARN has been set to high, the MODE_IND turns on the LED to indicate a valid transmission if the Code Word matches. If this is condition is met, data displays on the output of the D0-D7 channels. At that point, the Arduino takes over and captures the data on to its own system.

The basic function of the LEARN feature is to accept transmissions from an encoder that was sent on the receiver side. In order to do that, the decoder must first LEARN the encoders Code Word which will be sent through the channel. This is done by taking the LEARN to high to indicate that the decoder is ready to accept a new Code Word. The decoder will look for the valid transmission and if it finds one store the Code Word.

The following equation is used to calculate the length of the antennas which are used for the transmitter and receiver. L is the length in feet of quarter-wave length and f is the frequency in MHz.

$$L = \frac{234}{f} = \frac{234}{916MHz} \cong 0.255\ ft$$

Equation 2. Length of the Antenna

Internet	
Inputs	Digital Signal from Microcontroller
Outputs	Digital Signal to Microcontroller

The internet is being used by the Arduino to gather weather data for our local area. The connection that we are using is an Ethernet cable connected between the Arduino and an internet outlet. The data that we are using is provided for free online by the National Oceanic and Atmospheric Administration (NOAA). For this location, the weather XML feed we are using is Champaign/Urbana, University of Illinois-Willard, IL (KCMI) 40.04N 88.28W.

3.3 Block Description – Sprinkler Robot

GPS	
Inputs	3.3 VDC ± 0.5% from Power Electronics
Outputs	Digital Signal to Microcontroller

A Venus GPS device will be responsible for keeping track of the position and direction of the robot. The Venus GPS will update the microcontroller of the sprinkler robot with the coordinate values, and then the microcontroller will transmit the coordinate values to the base station. This will allow the microcontroller at the base station to send the robot the next command.

The GPS will be in direct contact with the microcontroller so it will not need to send signals directly through the transmitter chips. This setup will allow 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.

A National Marine Electronics Association (NMEA) message will be 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 below.

Start of Sequence	Payload Length (PL)	Payload		Checksum (CS)	End of Sequence
		Message ID	Message Body		
0xA0, 0xA1	Two Bytes	Message ID: 1 bytes; Payload up to 65535 bytes		1 byte	0x0D, 0x0A

Syntax of the NMEA message:

<0xA0, 0xA1><PL><Message ID><Message Body><CS><0x0D, 0x0A>

- Start of Sequence: Indicates the beginning of a message.
- Payload Length (PL): Is a 16 bit value that will tell you the length of the Payload.
- Payload: contains the Message ID and the Message Body. The Message ID can take the hexadecimal values between 0x01 to 0xFF, and the value will be cross referenced with the Message ID table so the meaning of the Message ID can be determined. For example: an output of 0xB3 is the Message ID that will give the WAAS status of the GPS receiver. The Message body will contain the message.
- Checksum (CS): Is the last message before the end of the sequence. It is an 8 bit exclusive OR to go along with only the payload bytes.
- End of Sequence: Indicates the end of the message.

Venus GPS features:

- Up to 20Hz update rate
- 2.5m accuracy
- WAAS support

- Works directly with active or passive antenna
- Internal flash for optional 75K point data logging
- Supports external SPI flash memory data logging
- 2.7 - 3.3 VDC supply

Wireless Communication	
Inputs	5 VDC ± 0.5% from Power Electronics Digital Signal from Microcontroller
Outputs	Digital Signal to Microcontroller

A transmitter circuit and receiver circuit are on the robot and are responsible for sending and receiving information between the base station Arduino and the robot Arduino using serial communication. This feature is important because it allows the base station to update GPS locations and motor controls for the robot.

The receiver and transmitter chips being used operate at different frequencies because multiple channels are required for this application of wireless communication. Because of this, the Linx chips operate at a range of 902-928 MHz to allow for multiple channels. FCC regulations allow free use of this band so this works perfectly for our setup.

The circuits are the same as the ones in the Wireless Communication block diagram for the base station. Please refer to that section for the schematics, data, and flow charts.

Power Electronics	
Inputs	12 VDC ± 5% from Battery
Outputs	3.3 VDC ± 0.5% to GPS 5 VDC ± 0.5% to Wireless Communication 5 VDC ± 0.5% to Microcontroller

The power electronics are actually a collection of low-dropout regulators (LDOs) that ensure that the voltages necessary to power the system will fall in the range of 12V, 5V, and 3.3V. The motors will mainly be powered by a 12 V battery. The Venus GPS will be powered by 3.3V.The microcontroller, receiver, and transmitter will be powered by 5Vs. The two LDOs being used are the LT1529-5 and the LT1529-3.3 which both have a dropout voltage of 0.6V at I_{OUT} = 3A. The LDOs do not need protection diodes because they are contained internally within the chip. The LDOs are also capable of I_{OUT} = 5A while being powered by V_{IN} =12V before the V_{OUT} can no longer stay at 5V (based on the simulations below). The LDO powered components will never draw more than 1A at the same time ensuring that this will not be a problem. The first test circuit below shows that as the voltage increases, the V_{OUT} eventually stabilizes at 5V as the V_{IN} reaches 5.5V. The second test circuit shows that as the I_{OUT} goes past 5A, the circuit can no longer provide 5V to the output.

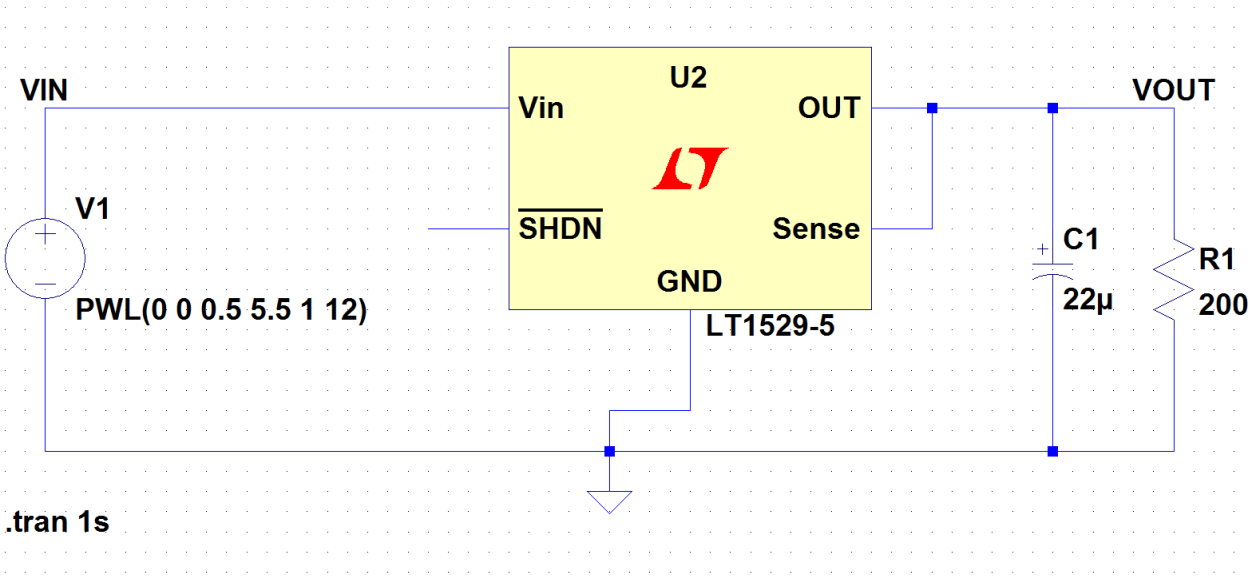

Figure 14. Test Circuit 1 for Power Distribution

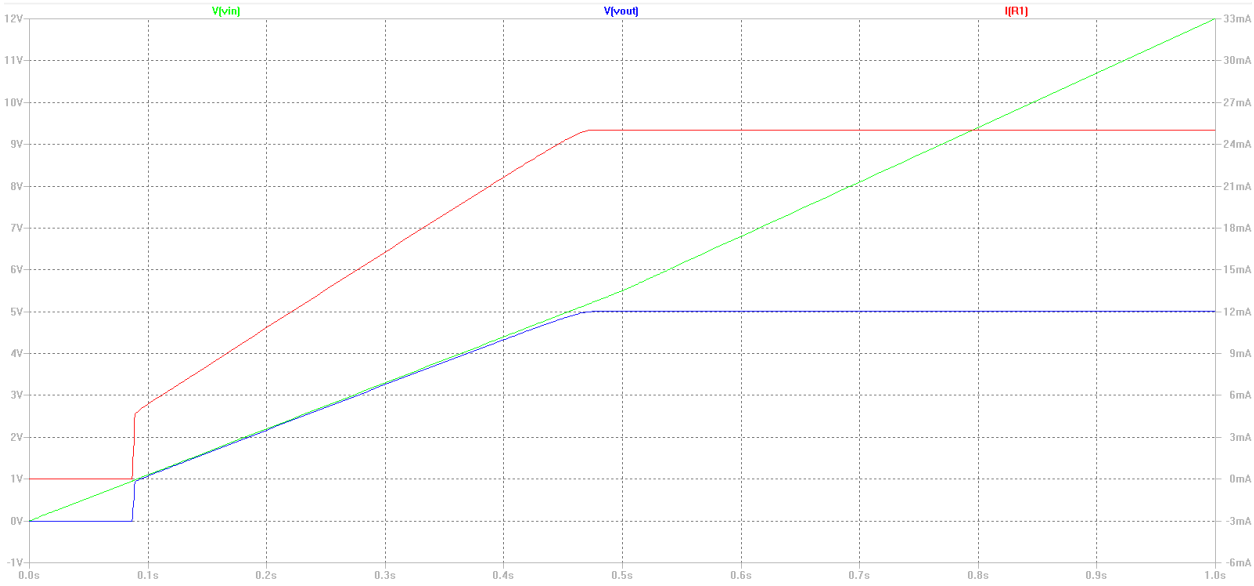

Figure 15. Simulation for Test Circuit 1

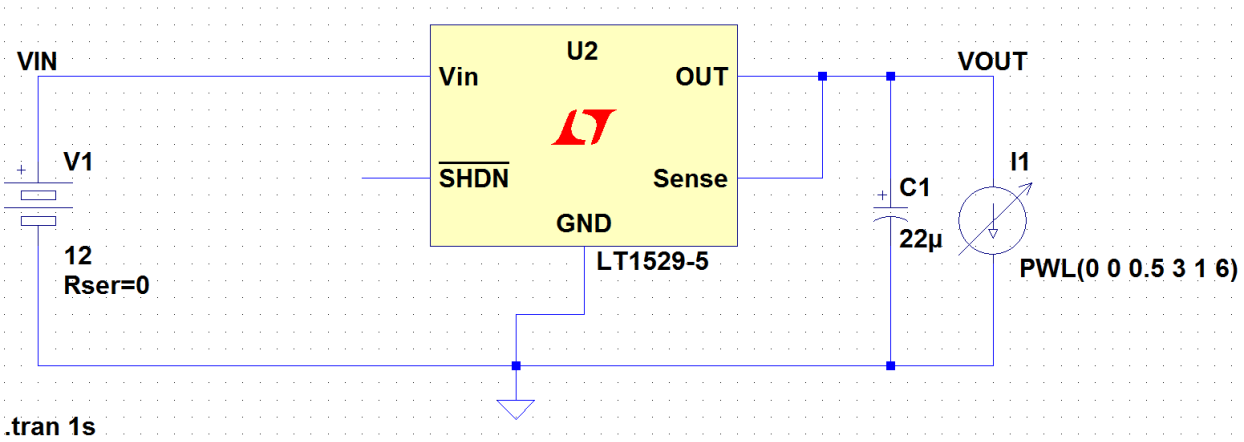

Figure 16. Test Circuit 2 for Power Distribution

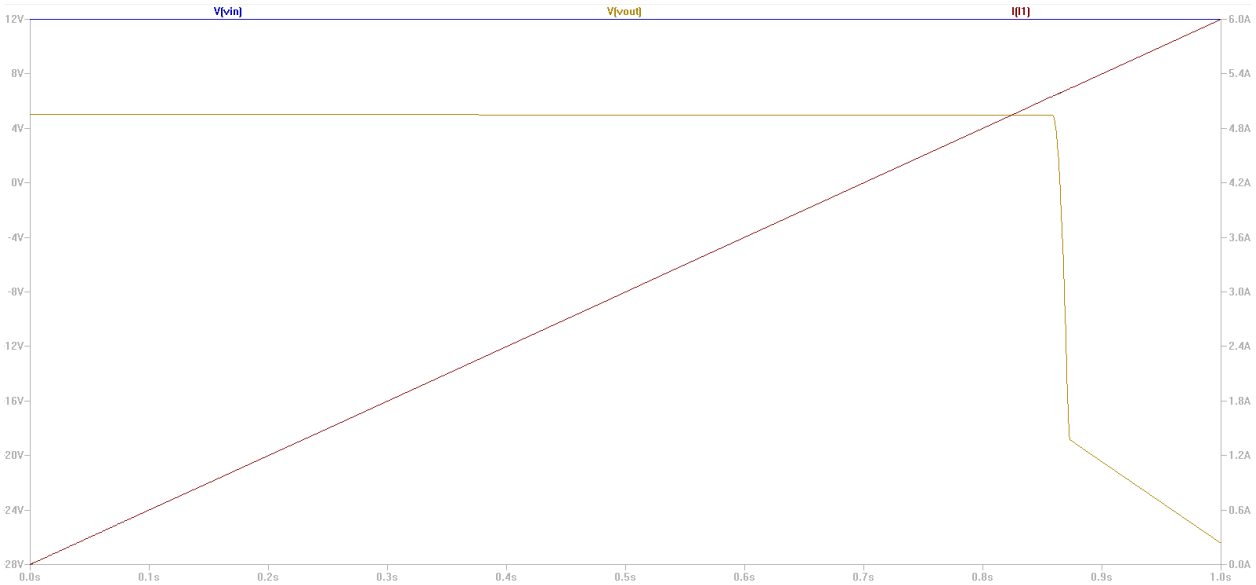

Figure 17. Simulation for Test Circuit 2

Battery	
Inputs	None
Outputs	12 VDC ± 5% to Power Electronics 12 VDC ± 5% to H-Bridge Circuits

The purpose of the batteries is to provide power to all of the components on the robot. The main power supply will come from multiple 12 V batteries. There is only one main battery to power the motors and the electronics related to the motors. This is an Xtreme Plus battery rated for 12V and 35AH. The main function of this battery is to power the two Dayton Motors and the Pittman Gear Motor. There is also another smaller SLA7-12 battery rated for 12V, 7Ah used for powering the GPS, transmitter, receiver, and Arduino. The smaller battery supply is also connected to two low-dropout regulators to ensure that once the battery is depleted past a certain point, it will stop powering the components. Below is the power distribution of the system:

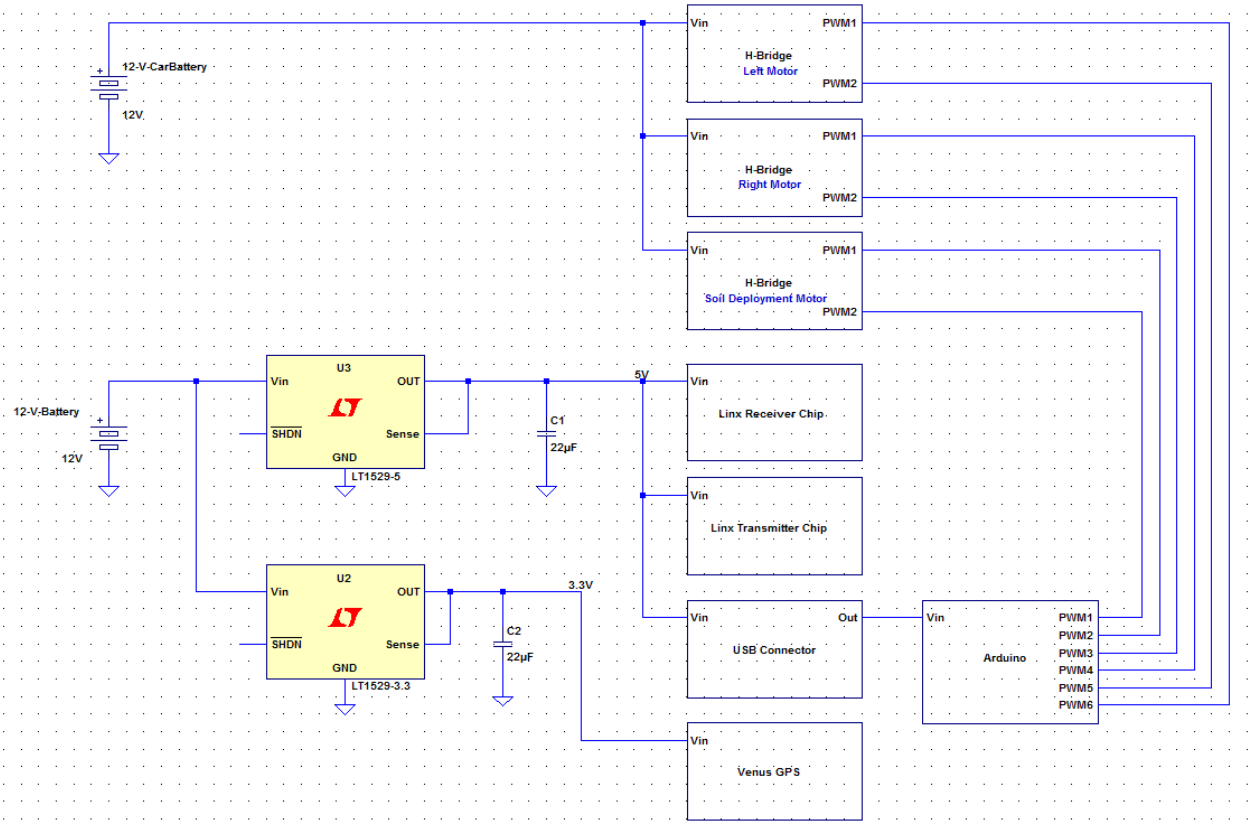

Figure 18. Power Distribution System on Robot

The most amount of current the main battery can is when the robot is driving itself to its location. This would imply that 2.1A are going through both motors and the LT1160 is drawing maximum amperage. This value is 10.3 mA. The most current being drawn at any one time would be 4.2103 A. The main battery runs at 12V, 35Ah which means that the motor could run at 12V for 8.3129 hours at maximum current draw.

Microcontroller	
Inputs	5 VDC ± 0.5% from Power Electronics Digital Signal from GPS Digital Signal from Wireless Communication Digital Signal from Soil Probe Analog Signal from Drive Motor Analog Signal from Soil Probe Motor
Outputs	Digital Signal to Wireless Communication 5 VDC ± 5% to Soil Probe 5 VDC ± 5% to H-Bridge Circuits

An Arduino Mega board will serve as the base station's microcontroller. The Arduino's purpose is to control the power electronics, wireless communication, and internet updates of the base station.

The Arduino on the robot is powered by a 12V battery that is connected to a low-dropout regulator. The regulator steps the voltage down to 5V and the 5V signal is sent to a USB connection. The USB connection is then sent to the Arduino, powering the Arduino through the USB port. This is done because the USB connector on the Arduino has built in fuses so if a short circuit ever occurs, the fuse will blow and not damage anything else on the circuit.

The Arduino is keeping track of three different potentiometers, each on the three different motors. Those will be used to track the distance travelled by each wheel with respect to the amount of turns the motor makes.

The Arduino is also in charge of measuring the soil resistance through the soil probe. The Arduino supplies 5V which drops across a reference resistor and also the soil probes. Using one of its analog inputs, the Arduino measures the voltage difference between the two resistors. From this, it then calculates the resistance of the soil and passes that information along to the wireless communication to send to the base station.

The sprinkler robot wireless communication will be transmitting signals to the base station, and the Arduino will provide the appropriate signal that will need to be sent to the base station. The sprinkler robot wireless communication will also receive signals from the base station, and the Arduino will analyze that data to perform its next task.

A Venus GPS device will be used to update the physical location of the sprinkler robot, and it will be up to the microcontroller to analyze that data. The microcontroller will send the coordinates it receives from the GPS to the base station. Then the base station will be able to determine which command to issue to the sprinkler robot.

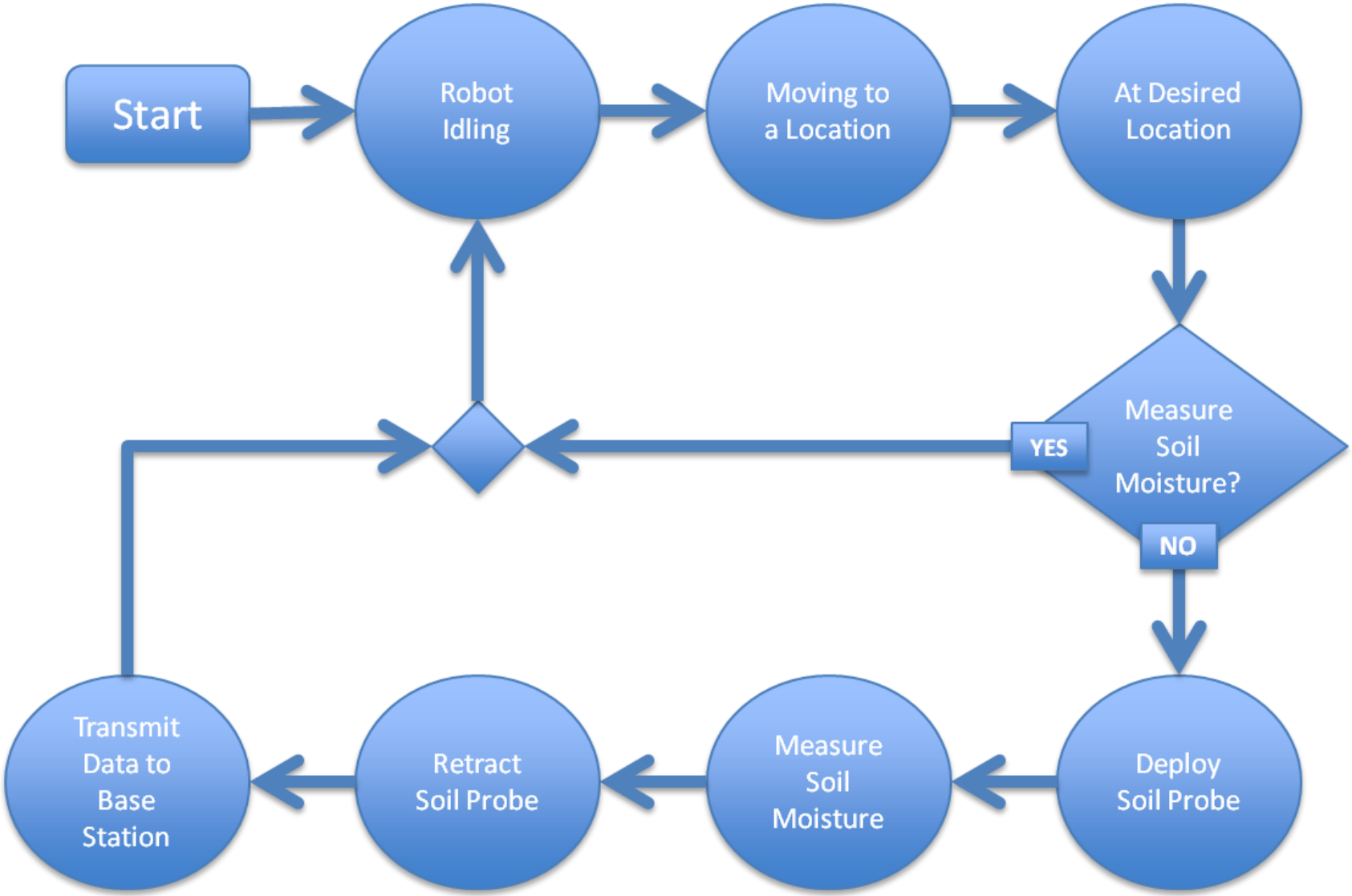

Figure 19. H-Bridge Driver Circuit

Soil Probe	
Inputs	5 VDC ± 5% from Microcontroller
Outputs	Analog Signal to Microcontroller

The soil probe is made out of two metal leads that are electrically isolated from one another. They are used to measure the resistance of the soil while they are inserted into the ground. This is accomplished by using the voltage divider rule. By constructing the circuit in figure 20, we can measure the voltage drop across a known reference resistance and compare that to the voltage drop across the soil. Then the voltage divider rule allows us to find the resistance of the soil in relation to the known reference resistor. The resistance of our reference resistor is 1MΩ so it is most accurate at measuring resistance values close to that value. The circuit is connected to the Arduino at the 5 Volts, analog in, and ground. The analog in will be what is measured to calculate the soil resistance.

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

Equation 3. Voltage Divider Rule

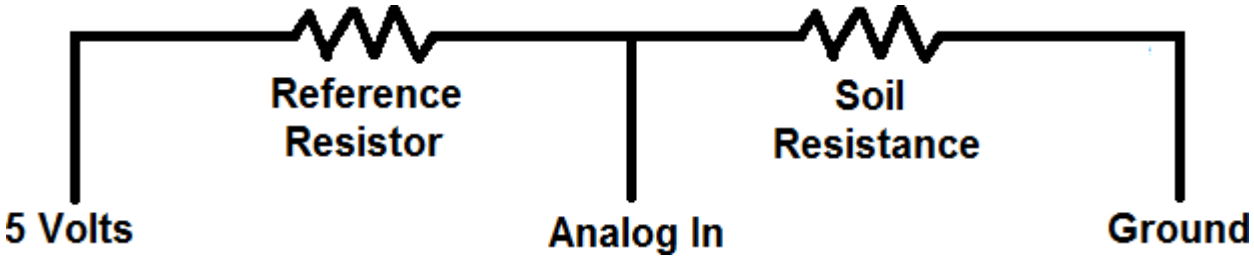

Figure 20. Soil Probe Circuit

Below is test code for an Arduino that measures resistance. This was made using the soil probe circuit. When the Arduino is connected to a PC, it displays what the measured resistance value is. This is accomplished through serial communication that this code shows how to set up. When doing verification, serial communication like this is important for displaying what the values of variables are.

```
int analogPin = 0;      // variable to measure the analog pin
int raw = 0;            // variable to store the raw input value
int Vin = 5;            // variable to store the input voltage
float Vout = 0;         // variable to store the output voltage
float R_Ref = 220;      // variable to store the reference resistance
float R_Meas = 0;       // variable to store the measured resistance
float buffer = 0;       // buffer variable for calculation

void setup()
{
  Serial.begin(9600);             // Setup serial
}

void loop()
{
  raw = analogRead(analogPin);    // Reads the Input PIN
  Vout = (5.0 / 1023.0) * raw;    // Calculates the Voltage on the Input PIN
  buffer = Vout / (Vin - Vout);   //
  R_Meas = R_Ref * buffer;        //
  Serial.print("Voltage: ");      //
  Serial.println(Vout);           // Outputs the information
  Serial.print("R_Meas: ");       //
  Serial.println(R_Meas);         //
  delay(1000);
}
```

Figure 21. Arduino Test Code for Soil Probe

Drive Motors	
Inputs	12 VDC ± 0.5% from H-Bridge Circuits
Outputs	Digital Signal to Microcontroller

The two main motors that are controlling the robot's movement will each be connected using H-Bridge circuitry. They are both 1LPV6 motors that run on 12V and 2.1 A. This set up will allow the robot to travel in forward and reverse directions depending on the switch setup. Since some of the switch combinations can cause a short circuit, a LT1160 Full-Bridge N-Channel Power MOSFET Driver was used. This gate driver has built in circuitry that prevents shoot-through which occurs when the switches turn on in a way that causes a short circuit. The driver will also take in two PWM modulated signals from the Arduino that will control the four switches the full-bridge setup. The circuit is below along with the truth table of possibilities.

IN TOP	IN BOTTOM	T GATE DR	B GATE DR
L	L	L	L
L	H	L	H
H	L	H	L
H	H	L	L

Figure 22. Truth Table for H-Bridge Driver

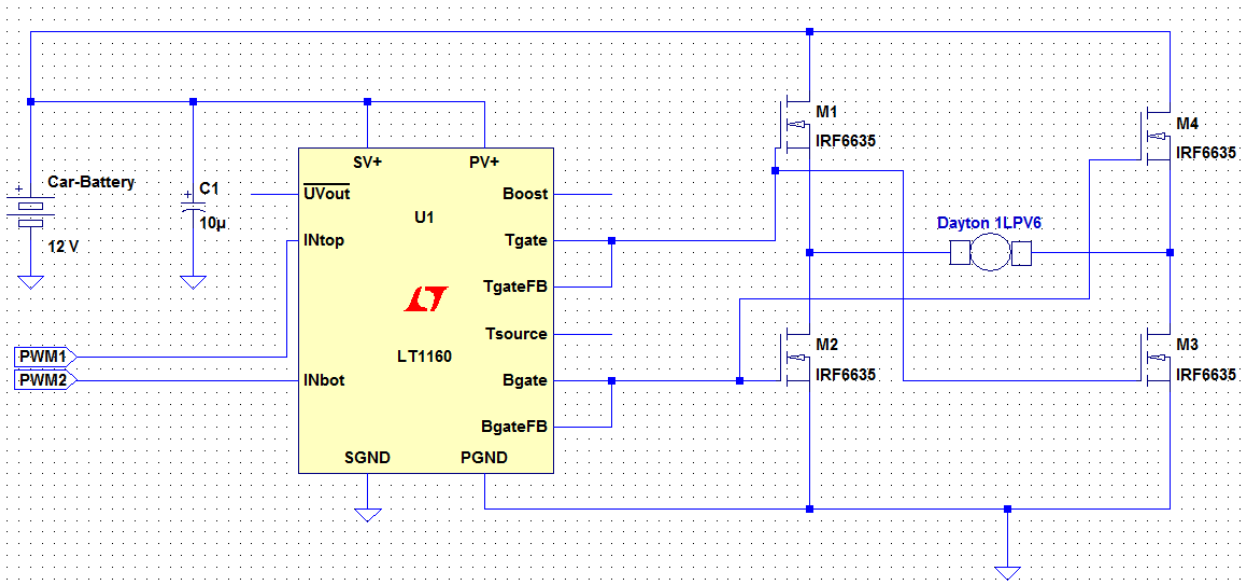

Figure 23. H-Bridge Driver Circuit

H-Bridge Circuits	
Inputs	12 VDC ± 0.5% from Battery 5 VDC ± 5% from Microcontroller
Outputs	12 VDC ± 0.5% to Drive Motor 12 VDC ± 0.5% to Soil Probe Motor

The H-Bridge circuit can be seen in the figure above it is mainly composed of the LT1160 Full-Bridge gate driver, the power MOSFETs and the motor. The MOSFETs are IRF6635s which allow up to 30V and a minimum current of about 25 A. This is more than enough to run the motors and is a good choice because it only has a $R_{DS,ON}$ of 1.3mΩ. The other MOSFET that would work would be the IRFZ44 which allows up to 60V and a I_D = 50A to pass through the circuit. The $R_{DS,ON}$ = 0.024Ω which is also fairly good.

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

Equation 4. Power Lost in MOSFET

The IRF6635 would lose about 0.005733 W and the IRFZ44 would lose about 0.10584 W. These values are not extreme and if necessary, a heat sink could be added to each MOSFET. The rest of the circuit explanation was given in the Drive Motors section of the block diagram. Please refer there for more details.

Soil Probe Motor	
Inputs	12 VDC ± 0.5% from H-Bridge Circuits
Outputs	Digital Signal to Microcontroller

The soil probe motor is a gear motor attached to a rail system that slides the soil probes into and out of the ground. The motor that we are using for this is the Pittman GM9434G807. This is a 12 VDC motor that has a gear ratio of 65.5:1. The H-bridge motor drive will be in direct control of this motor by supplying it with the positive or negative 12V to drive the rail system down or up. The MOSFETs will be IRF6635 and will be powered by an LT1160 Full-Bridge N-Channel Power MOSFET Driver. The PWMs will be supplied by the on-board microcontroller. See the H-Bridge Drive Motor Figure in the Drive Motor section for circuit.

Figure 24. Pittman GM9434G807

4 - Requirements and Verification

4.1 Testing Procedures – Base Station

Power Outlet Requirements	Verification
1) Supplies 120V ± 5% at 60 Hz ± 0.5%	• Using a Volt-meter, insert the leads into the flat holes of the power outlet. Make sure not to touch the leads while testing. Record the AC voltage of the power outlet to make sure it is within the desired range. • Using an oscilloscope, insert the leads into the flat holes of the power outlet. Make sure not to touch the leads while testing. Record the frequency of the power outlet to make sure it is within the desired range.

Hose Actuator Requirements	Verification
1) Turns the flow of water on and off to the sprinkler a) When given an input signal of 24 VAC ± 1% at 60 Hz ± 0.5% it turns on b) For an input signal less than 5 VAC ± 1% 60 Hz ± 0.5% it remains off	a) • Using a signal generator, supply the hose actuator with 24 VAC ± 1% at 60 Hz ± 0.5%. • Check to see if the solenoid in it moves to the open position. b) • Using a signal generator, supply the hose actuator with a sweep of voltages up to 5 VAC ± 1% 60 Hz ± 0.5%. • Make sure that the solenoid does not move to the open position

Power Electronics Requirements	Verification
1) Low-Drop Voltage Regulator (LDO) is outputting 5 VDC ± 0.1%	a) • Probe the input with a Volt-meter.

a) V_{IN} > 5.5VDC ± 0.5% b) I_{OUT} < 5 A ± 0.5%	Display signal on oscilloscope. b) • Probe the output with an Amp-meter. Ensure that the current draw is less than 5A. Display waveform on an oscilloscope.
2) Wall outlet transforming to correct voltage a) V_{IN} = 120VAC± 5% b) V_{OUT} = 24VAC ± 5%	a) • Using a Volt-meter, measure the input voltage. Display the output on an oscilloscope. b) • Using a Volt-meter, measure the output voltage. Display the output on an oscilloscope.
3) USB connector is receiving 5VDC ± 1% from the computer	• Using a Volt-meter, measure the USB connector voltage. Display the output on an oscilloscope.

Microcontroller Requirements	Verification
1) The Arduino is receiving the correct amount of voltage a) The USB Connector is making a good connection b) V_{IN} at Arduino is 5VDC ± 5%	a) • Use a multimeter at the USB connector to ensure a connection is made. • Check for proper grounding by measuring voltage using a Volt-meter at output from LDO. b) • Probe the USB Arduino V_{CC} port with a Volt-meter and display signal on the output.
2) There must be data transfer between the microcontroller and the wireless communication. a) The Arduino must be able to send information to the transmitter in order to transmit data to the sprinkler robot.	a) • Wire the transmitter and receiver to the Arduino. • Connect the Arduino to a PC and open serial communication • Send a binary signal that is at least 20 bits long

b) The Arduino must be able to read the information that the receiver is getting from the sprinkler robot.	• Have the Arduino print out the signal values that it is sending to the transmitter. • Have the Arduino at the sprinkler robot print out the signal that it is receiving. • Compare the transmitted signal at the base station Arduino to the signal received at the sprinkler robot Arduino. • Do this test 10 times and make sure that the same signal is sent and received all 10 times. It must pass all 10 tests. b) • Wire the transmitter and receiver to the Arduino. • Connect the Arduino to a PC and open serial communication • Send a binary signal that is at least 20 bits long • Have the Arduino print out the signal that it gets at the receiver. • Compare the printed received data to that of the sprinkler robot Arduino. • Do this test 10 times and make sure that the same signal is sent and received all 10 times. It must pass all 10 tests.
3) Gathers information about the weather to predict when to activate the sprinkler robot a) Arduino gathers data from NOAA website b) Determines when to activate the sprinkler robot	a) • Make sure that the internet requirements are satisfied. • Connect the Arduino to a PC and open serial communication. • Check what variable values are being stored in the Arduino. Makes sure that the variables being stored are the same as those from online • Do this test 5 times and have the

	weather data online be different for each test. It must pass all 5 tests. b) • Connect the Arduino to a PC and open serial communication. • Run through the program to see if the anticipated variables are the same as the actual variables. • Check if the predictions about robot activation match what is intended. • Do this test 5 times with different input variables each time. It must pass the test all 5 times.

Wireless Communication Requirements	Verification
1) The Arduino is sending the correct data type (byte) to the encoder. a) Correct data from the Arduino at each bit of the D0-D7 values. b) SEND pin is set to high c) DATA_OUT is outputting serial data stream	a) • Compare values set in Arduino code to the values measured with 8-LEDs at the D0-D7 pin. • Do this test 1000 times with different input variables each time. Record in table and look for errors. b) • Using a Volt-meter, measure the voltage at the TX_CNTL pin to ensure that the transmitter has been activated. Voltage should be set as high. c) • Using a Volt-meter, measure the voltage at the DATA_OUT pin and ensure bits are being transferred. Display output on oscilloscope. • Do this test 1000 times with different input variables each time. Record in table and look for errors.
2) Transmitter is sending data on the correct channel	• Probe the antenna voltage to ensure a signal is being transmitted by using a

	Volt-meter and oscilloscope. • Using a Spectrum Analyzer, find the signal from the range 906Mhz – 921Mhz. Ensure the dB is high enough for clean reception to be made.
3) Receiver chip is getting data from the transmitter chip a) Ensure channels are set to correct frequency for both receiver and transmitter b) DATA pin on receiver chip is outputting voltage	a) • Check dip-switches to make sure they are set to the correct values b) • Using a Volt-meter, measure the DATA pin on receiver chip. Ensure that there is a voltage being outputted. • Display signal on oscilloscope. • Do this test 1000 times with different input variables each time. Record in table and look for errors.
4) The decoder chip is receiving the data and sending it to the Arduino a) DATA_IN pin sees a rising edge b) Code Word matches the one in memory c) D0-D7 contain correct data	• • Using a Volt-meter, measure the Probe DATA_IN pin on decoder chip. Ensure that there is a voltage being inputted. • Display signal on oscilloscope. • Do this test 1000 times with different input variables each time. Record in table and look for errors. • LEARN mode accepts correct code word from encoder ensuring a transmission is possible • Do this test 1000 times with different input variables each time. Record in table and look for errors. • Display D0-D7 on an LED to ensure that they are the correct values being sent from Arduino. • Do this test 1000 times with different input variables each time. Record in table and look for errors.

	• Verify BAUD rate is correct by tracking bits per second on Arudino. Compare rate to that of Encoder and Decoder rate.

Internet Requirements	Verification
1) Connects to the NOAA website for weather information	• Plug an Ethernet cable into the internet outlet and then into a PC • Check that you can access the NOAA website at www.noaa.gov

4.2 Testing Procedures – Robot Sprinkler

GPS Requirements	Verification
1) Must update coordinates from global positioning satellites utilizing WAAP capabilities. a) In a stationary position with an accuracy of less than one meter. b) During movement with an accuracy of 2.5m ± 0.2m.	a) • Set up the Venus GPS device in the laboratory for stationary testing. • Connect the output of the Venus GPS device to the Arduino. • Enable WAAP on the Venus GPS device. • Connect one of the two GPS antennas on the roof of Everitt Laboratory to the GPS device. • Cross verify results with the actual coordinates of the antennas by printing out the results on the Arduino. • Cross verify coordinate data from the Venus GPS with the actual location of the antenna. • Fix the coding with the Arduino if the coordinate reading does not match the actual location within the acceptable error. • Repeat previous steps for the other GPS antenna on the roof of Everitt Lab. • Repeat steps using the GPS antenna that will be used on the Sprinkler Robot. • Once an accuracy of less than one meter has been achieved for each antenna, and is consistent within the acceptable error for multiple trials, the stationary position testing will be complete. b) • Install the Venus GPS device with its antenna on the Sprinkler Robot. • Move the robot around Everitt Lab's

	perimeter while sending the coordinate updates to the Base Station. • Print out the data at the base station to display the coordinate updates. • Cross verify the coordinate results with the actual coordinates. • Continue with cross verifying coordinates for movement around Everitt Lab until the proper readings show up. • Once an accuracy of 2.5m ± 0.2m has been achieved, and is consistent within the acceptable error for multiple trials, the movement testing will be complete.

Wireless Communication Requirements	Verification
1) The Arduino is sending the correct data type (byte) to the encoder. a) Correct data from the Arduino at each bit of the D0-D7 values. b) SEND pin is set to high c) DATA_OUT is outputting serial data stream	a) • Compare values set in Arduino code to the values measured with 8-LEDs at the D0-D7 pin. • Do this test 1000 times with different input variables each time. Record in table and look for errors. b) • Using a Volt-meter, measure the TX_CNTL pin to ensure that the transmitter has been activated. Voltage should be set as high. c) • Using a Volt-meter, measure the DATA_OUT pin and ensure bits are being transferred. • Display output on oscilloscope. • Do this test 1000 times with different input variables each time. Record in table and look for errors.

2) Transmitter is sending data on its channel a) Sending on the correct channel.	a) • Using a Volt-meter, measure the antenna voltage to ensure a signal is being transmitted • Display output on oscilloscpe • Using a Spectrum Analyzer, find the signal from the range 906Mhz – 921Mhz. Ensure the dB is high enough for clean reception to be made.
3) Receiver chip is getting data from the transmitter chip a) Ensure channels are set to correct frequency for both receiver and transmitter b) DATA pin on receiver chip is outputting voltage	a) • Check dip-switches to make sure they are set to the correct values b) • Using a Volt-meter, measure the DATA pin on receiver chip. Ensure that there is a voltage being outputted. • Display signal on oscilloscope. • Do this test 1000 times with different input variables each time. Record in table and look for errors.
4) The decoder chip is receiving the data and sending it to the Arduino a) DATA_IN pin sees a rising edge b) Code Word matches the one in memory c) D0-D7 contain correct data	a) • Using a Volt-meter, measure the Probe DATA_IN pin on decoder chip. Ensure that there is a voltage being inputted. • Display signal on oscilloscope. • Do this test 1000 times with different input variables each time. Record in table and look for errors. b) • LEARN mode accepts correct code word from encoder ensuring a transmission is possible • Do this test 1000 times with different input variables each time. Record in table and look for errors. c)

	• Display D0-D7 on an LED to ensure that they are the correct values being sent from Arduino. • Do this test 1000 times with different input variables each time. Record in table and look for errors. • Verify BAUD rate is correct by tracking bits per second on Arudino. Compare rate to that of Encoder and Decoder rate.

Power Electronics Requirements	Verification
1) First low-drop voltage regulator (LDO) is outputting 5 VDC and the second LDO is outputting 3.3 VDC. a) $V_{IN} > 5.5VDC \pm 0.1\%$ b) $I_{OUT} < 5\ A \pm 0.5\%$	a) • Probe the output with a Volt-meter and read the voltage. Display signal on oscilloscope. • Probe the input with a Volt-meter. Display signal on oscilloscope. b) • Probe the output with an Amp-meter. Ensure that the current draw is less than 5A. Display waveform on oscilloscope.

Power Electronics Requirements	Verification
1) Both the 12 VDC, 7 AH and 12 VDC 35 Ah batteries are fully charged	• Using a Volt-meter, measure the output voltage of both batteries • If the voltage is less than 12 VDC± 0.5%, use a battery recharger

Microcontroller Requirements	Verification
1) The Arduino is receiving the correct amount of voltage a) The USB Connector is making a good connection b) V_{IN} at Arduino is 5VDC ± 5%	a) • Measure using a multimeter at the USB connector to ensure it is not a floating value. • Check for proper grounding by

	measuring the voltage using a Volt-meter at output from LDO. b) • Probe the USB Arduino V_{CC} port with a Volt-meter and display signal on an oscilliscope.
1) There must be data transfer between the microcontroller and the wireless communication. a) The Arduino must be able to send information to the transmitter in order to transmit data to the base station. b) The Arduino must be able to read the information that the receiver is getting from the base station.	a) • Wire the transmitter and receiver to the Arduino. • Connect the Arduino to a PC and open serial communication. • Send a binary signal that is at least 20 bits long • Have the Arduino print out the signal values that it is sending to the transmitter. • Have the Arduino at the base station print out the signal that it is receiving. • Compare the transmitted signal at the sprinkler robot Arduino to the signal received at the base station Arduino. • Do this test 10 times and make sure that the same signal is sent and received all 10 times. It must pass all 10 tests. b) • Connect the Arduino to a PC and open serial communication. • Send a binary signal that is at least 20 bits long • Have the sprinkler robot Arduino print out the signal that it gets at the receiver. • Compare the printed received data to that of the base station Arduino. • Do this test 10 times and make sure that the same signal is sent and received all 10 times. It must pass all

	10 tests.
2) The microcontroller must receive the sprinkler robot's coordinates from the GPS. a) The Arduino must be able to read the information that the GPS is providing.	a) • Wire the Venus GPS device to the Arduino. • Connect the Arduino to a PC and open serial communication. • Make the Arduino print out the coordinate values that it receives from the Venus GPS device. • Compare the data being printed out with the actual data that the GPS device is sending to the Arduino. • Do this test 5 times and make sure that it passes all 5 times.
3) Sends the measured soil resistance value to the wireless transmitter a) Measures the soil resistance b) Sends the soil resistance to the wireless communication	a) • Verify that all soil probe requirements are satisfied b) • Connect the Arduino to a PC and open serial communication • Record the values of the variables that are being sent to the wireless communication • Using an oscilloscope, see if the variable values that are being sent match up with the measured output • Do this test 5 times and make sure that it passes all 5 times. Each time use different resistance values

Soil Probe Requirements	Verification
1) Measure soil resistance with up to 5% error a) Resistance between probes must be greater than 100MΩ b) Circuit reference resister must be the same as that used in the	a) • Make sure that the soil probes are not contacting any other material • Using an Ohm-meter, measure the resistance across the two soil probes.

program c) Resistance value measured by Arduino is accurate within 5%	Verify the resistance is greater than 100MΩ b) • Using an Ohm-meter, measure the resistance of the reference resister • Open up the program for the Arduino • Check that the variable value in the Arduino program is the same as the reference resister value. c) • Connect the Arduino to a PC and open serial communication • Put across the soil probes a test resistance value • Compare the stored variable value for measured resistance to that of the test resistance value • Do this test 5 times and make sure that it passes all 5 times. Use a different reference resistance value each time

Drive Motors Requirements	Verification
1) 144:1 turns ratio at 2.1A ± 1% and 12 VDC ± 1%	• Use the potentiometer to track how many turns the motor makes every minute. • Record the data and check if the motor is running properly.
2) Adjustable Speed	• Have the Arduino vary PWM signals and see if the speed can be adjusted. • Record values in a table for different PWM signals and compare the speed.

H-Bridge Circuit Requirements	Verification
1) Gate Driver being powered correctly a) SV+ and PV+ getting correct voltages b) UVout floating	a) • Probe the voltages at SV+ and PV+ to ensure they are 12VDC ± 1%. Display voltage signal on oscilliscope. b) • Probe the UVout pin with a multimeter and ensure that it is floating
2) PWM signals correct a) T_{gate} and B_{gate} receive the correct signals at the correct frequencies b) Power MOSFETs turning on	a) • Compare signals coming from the Arduino to those that are going into the gate driver to make sure they match • Ensure voltage is strong enough to power signal by using a Volt-meter and measuring the INtop pin and INbottom pin. • Compare the frequencies at INtop to T_{gate} using the oscilloscope and see if they match. • Compare the frequencies at INbot to B_{gate} using the oscilloscope and see if they match. b) • Probe the power MOSFET with a current source and ensure that current is flowing through it. Display signal on oscilloscope. • Probe the V_{GS} voltage and ensure it is high enough to turn the gate on based off of the data sheets.

Soil Probe Motor Requirements	Verification
1) Is able to insert and remove the soil probe into the ground with ½" accuracy a) Power electronics supplies ±12 VDC ± 0.5% b) Measures position within ½" accuracy	a) • Using a Volt-meter, check that voltage input into the motor • For forward slide movement, the voltage should be +12 VDC ± 0.5% • For reverse slide movement, the voltage should be -12 VDC ± 0.5% b) • Connect the Arduino to a PC and open serial communications • Move the slide to a location and check that the Arduino variables are storing the anticipated values for that location with ½" accuracy

4.3 Tolerance Analysis

The most important part of the sprinkler robot is ensuring the system is powered correctly with the batteries. This means that we need to ensure that the two batteries on the robot are always outputting 12 VDC at all times. In addition, a test to measure how long each battery lasts under maximum current load and the voltage/current drops that occur during that time is vital to record.

Previously, it has been determined that the maximum amount of current in the main battery ever being drawn at one time is 4.2103 A which occurs while running the two Dayton motors and powering the gate driver. To test the tolerance of the batteries, a test needs to be done where the battery draws a constant current of 4.2103 A on a load. The battery voltage needs to be recorded in the form of data points that indicate the remaining voltage. This is done using a voltage and current sensor on the microcontroller that records the values over time. These points will then be plotted in Excel so they can be compared to the expected results. A current graph also displays how the current is affected over time in the load. The main battery runs at 12V, 35Ah which means that the motor could run at 12V for 8.3129 hours at maximum current draw.

The secondary battery runs on 12V, 7Ah and has a maximum current draw of 677.56 mA when all components are running on full power. This mean the battery would last for around 10 hours at its 12V rating. Again, the same test as the main battery would need to be done to ensure the battery would function properly over time.

The voltage regulation in the system was put in place to ensure that even if the voltage dropped under 12V, most of the system could still be powered due to the low-drop out regulators. The LDOs will also need to be tested with a voltage sweep and current sweep to ensure that they can handle certain ranges of voltages and currents.

A subtest is necessary on each block component that is battery powered to see the amount of power consumption each takes over a specific time interval when drawing an average amount of current, idle current, and the maximum current which occurs when operating the most processes or the most demanding process.

5 - Cost Analysis and Schedule

5.1 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

5.2 Parts Cost

Name	Part Number	Quantity	Unit Price	Combined Price	Personal Cost
Arduino	Arduino Mega	1	$56.08	$56.08	$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
Decoder	LICAL-DEC-MS001-ND	2	$3.98	$7.96	$7.96
Encoder	LICAL-ENC-MS001-ND	2	$3.98	$7.96	$7.96
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		1	$100.00	$100.00	$100.00
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	IRFZ44RPBF-ND	12	$2.04	$24.48	$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 200 µF	EGPA101ELL201ML20S	2	$2.89	$5.78	$0.00
Diodes	1N4004	8	$0.03	$0.24	$0.00
LED		2	$0.03	$0.06	$0.00
Resistors 100kΩ		8	$0.01	$0.08	$0.00
Dipswitch	SDA03H0SBR	1	$1.20	$1.20	$0.00
Dipswitch	SDA04H0SBR	1	$1.24	$1.24	$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, 35Ah Battery	Xtreme Plus	1	$60.00	$60.00	$0.00
TOTAL				$1,272.91	$470.07

5.3 Grand Total

Grand Total:

$1,272.91 + $50,625.00 = $51,897.91

5.4 Schedule

Week	Task Description	Group Members
2/4	**Proposals Due, Call Lincoln Labs**	
	Design Transmitter and Receiver EAGLE-Cad	Denis
	Research Soil Moisture Measuring	Kevin
	Research Differential GPS	Jose
2/11	**Mock Design Reviews**	
	Design Power Distribution System for Robot Car	Denis
	Collect Soil Resistance Data	Kevin
	Construct GPS Circuit with WAAS	Jose
2/18	**Design Review Closes, Call Lincoln Labs**	
	Simulations for H-Bridge Circuit and Buck Converter	Denis
	Construct Arduino Soil Probe	Kevin
	Interface GPS with Arduino	Jose
2/25	**Design Review**	
	Build and Test H-Bridge Circuit and Buck Converter	Denis
	Build System for Soil Probe Deployment	Kevin
	Testing GPS Component with Everitt GPS Antennas	Jose
3/4	**Call Lincoln Labs**	
	Build Receiver and Transmitter for Base Station	Denis
	Construct Arduino Controlled Robot Movement	Kevin
	Install GPS in Robot Car	Jose
3/11	**Individual Progress Reports Due**	
	Establish Communication Between Base Station and Robot	Denis
	Build Arduino Controlled Sprinkler System	Kevin
	Build Robot GPS Movement System	Jose
3/18	**Spring Break**	
3/25	**Mock-up Demos and Mock Presentation Sign-up Closes, Call Lincoln Labs**	
	Mock Presentation - Test Slides	Denis
	Mock Presentation - Control Slides	Kevin
	Mock Presentation - Design Slides	Jose
4/1	**Last Day to Request First Revision PCB Fabrication**	

	Document Tolerance Analysis Verification	Denis
	Build Weather Collection System	Kevin
	Test the Reliability of Robot Moving to Location	Jose
4/8	**Last Day to Request Final Revision PCB Fabrication, Call Lincoln Labs**	
	Final Tests and Verification - Wireless Communication	Denis
	Final Tests and Verification - Base Station	Kevin
	Final Tests and Verification - Robot	Jose
4/15	**Demo and Presentation Sign-up Closes**	
	Final Touches on Project	Denis
	Presentation – PowerPoint	Kevin
	Demo Preparation	Jose
4/22	**Demos and Presentations, Call Lincoln Labs**	
	Final Paper - Introduction and Design	Denis
	Final Paper - Design Verification	Kevin
	Final Paper - Costs and Conclusion	Jose
4/29	**Presentations, Final Paper, Lab Notebooks, Checkout**	
	Final Paper	Denis
	Lab Notebooks	Kevin
	Checkout and Parts Return	Jose

6 - Ethics and Safety

6.1 Ethics

The Smart Sprinkler Robot System is a product that will be able to do all of the details listed in the benefits and features. In order to realistically make this robot marketable, it will have 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*

6.2 Safety

In order to make a worthwhile sprinkler robot, we will need to be able to be able to guarantee the consumer that the sprinkler robot will be safe for the consumer to own. When introducing water to electrical devices, the engineer must keep in mind that the electrical devices must be properly sealed to make sure that water does not get onto the electrical devices. This will comply with the safety of the humans around and of the device.

The battery used for the sprinkler robot contains battery acid. With that said, we want to make sure that the battery is stored on inside the robot properly. That will ensure that if there is a battery acid leak, the rest of the robot, humans, or the grass around is not damaged.

In order to guarantee circuit and human safety, the sprinkler robot will use proper grounding. The grounding will serve as a means for safety because it will prevent shocks to the user in case the electrical insulation fails.

In order to be sure that the Smart Sprinkler Robot System is safe of hazardous materials, all the parts used in this system are compliant with the Restriction of Hazardous Substances (RoHS) Directive. This means that all components are free of lead, mercury, cadmium, hexavalent chromium, Polybrominated biphenyls, and Polybrominated diphenyl ether.

7 - Appendix

7.1 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] Linear Technology [Online]. Available: http://www.linear.com/

[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

[13] Spark Fun Electronics [Online]. Available: 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

[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

7.2 Extra Figures

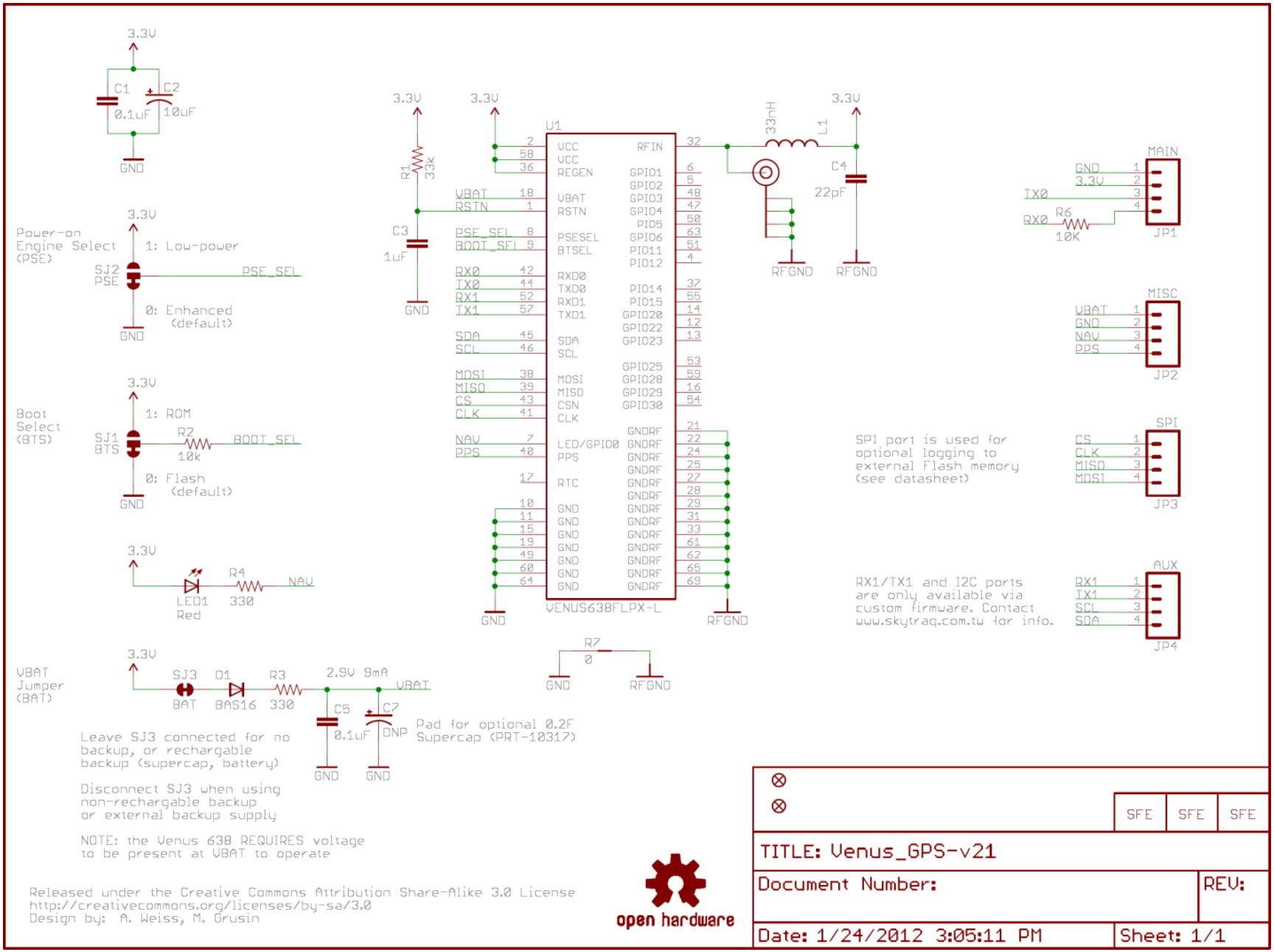

Figure A1. Venus GPS v21

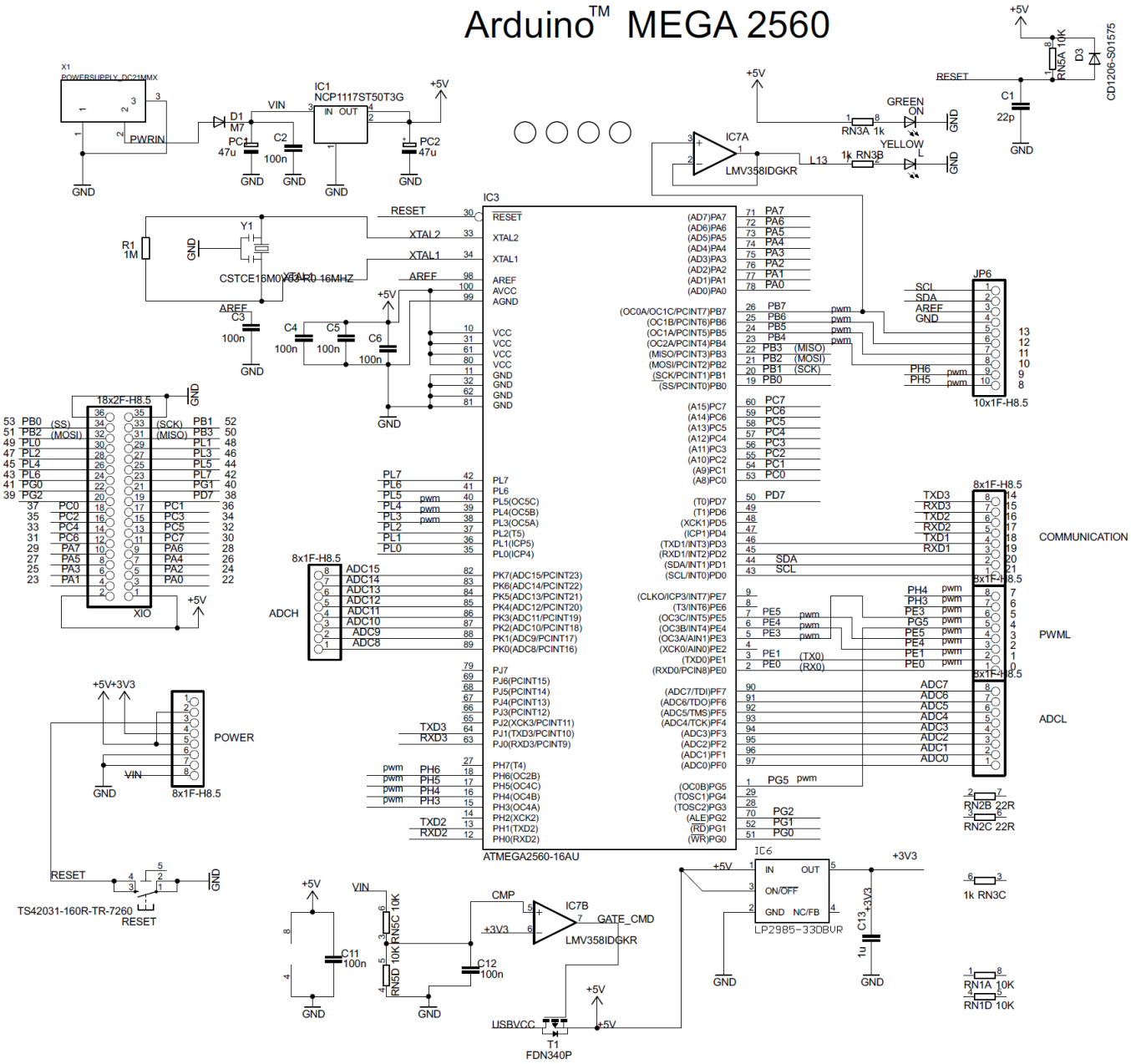

Figure A2. Arduino MEGA 2560 Diagram Part 1

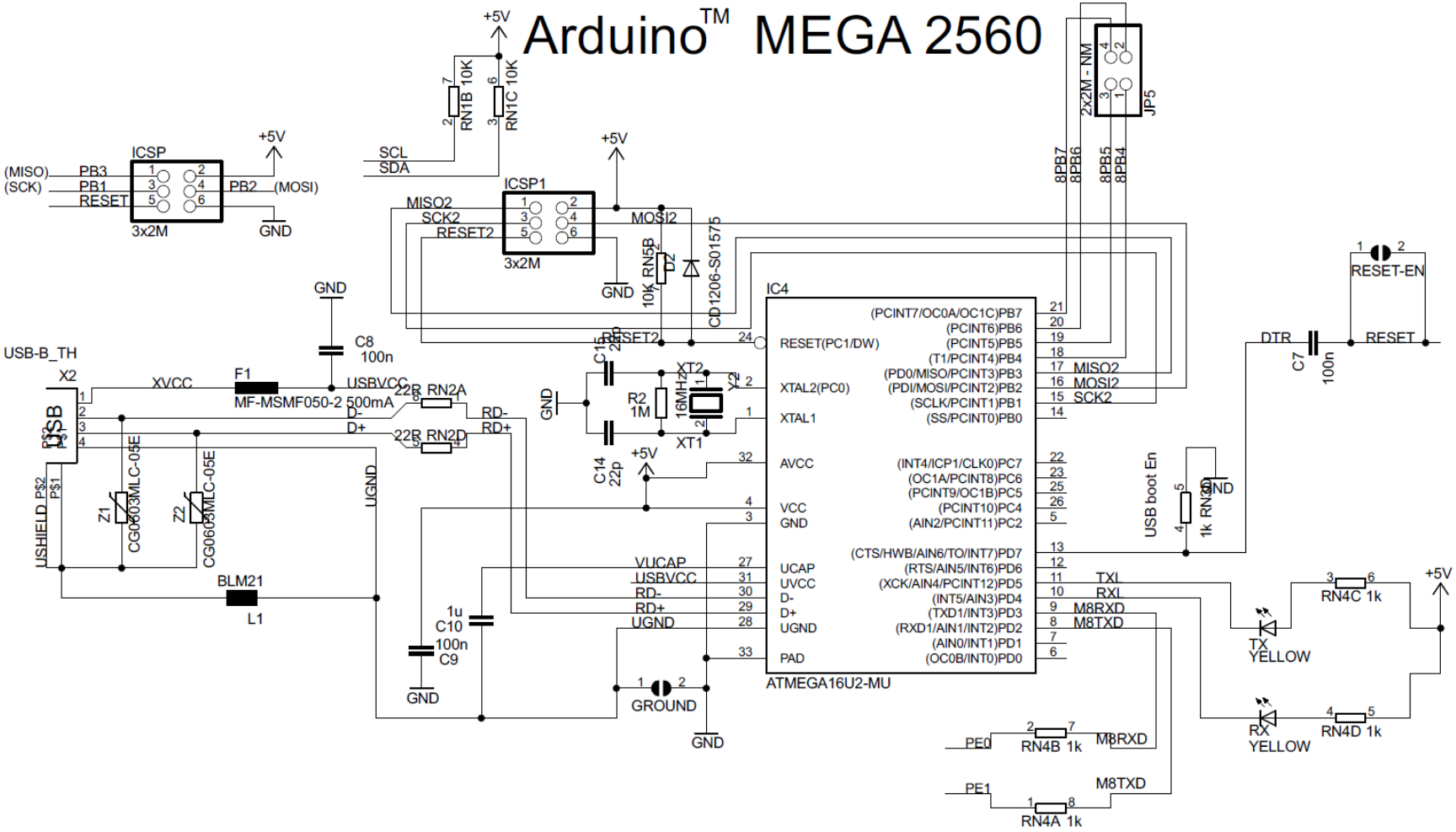

Figure A3. Arduino MEGA 2560 Diagram Part 2

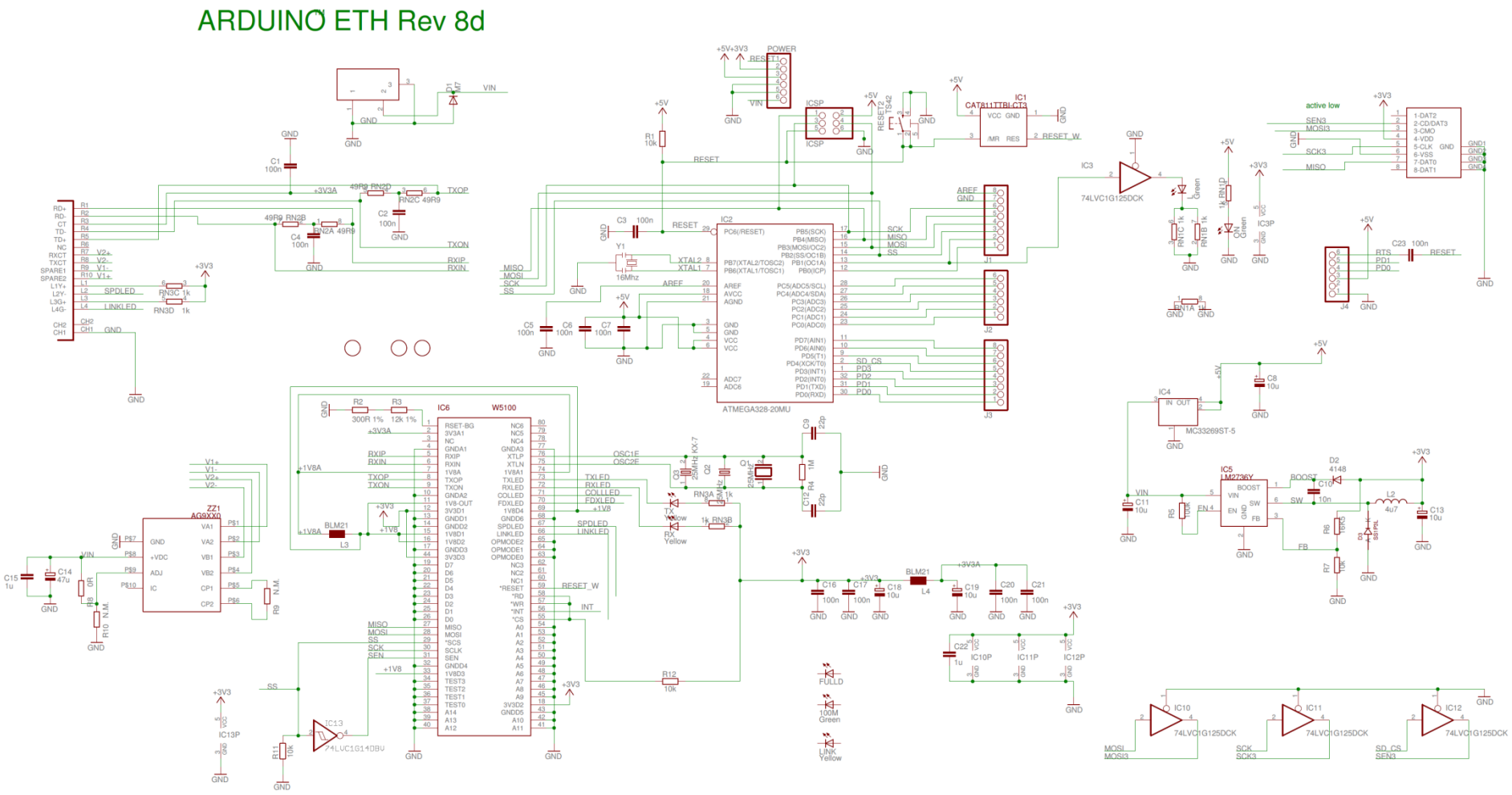

Figure A4. Arduino Ethernet