ChargeSpot: Wireless Power Transfer through High Resonant Frequency

Theophilus Essandoh, Ryan Johnson, and Emelio Watson

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Today there is a great push for more electronic devices to operate using some kind of wireless technology. This can be seen from RC cars to game controllers to the daily charging of cell phones. Society is becoming less and less wired and this project aims to take that phenomenon one step further with the aid of available technology. ChargeSpot is intended to demonstrate the feasibility of an autonomous electric vehicle charging system for residential use. A vast majority of today's electric vehicles are being charged with a power cable connecting the vehicle to the charging station. The goal of ChargeSpot is to have very little user interaction and no physical connection between the vehicle and the charging station.**

***Index Terms* — Wireless charging, resonators, oscillators, microcontrollers, electric vehicles, magnetic field, wireless power, autonomous.**

I. Introduction

With electric vehicles slowly becoming a more popular choice among car shoppers due to government incentives and tax breaks, a more user friendly means of charging these vehicles makes economic sense. Remembering to charge your electric vehicle every day is hard enough as it is, and if you forget to charge it overnight there is a possibility you may not make it to work the next day. An electric vehicle cannot be charged in five minutes and if you do not have a backup means of transportation owning an electric vehicle can prove to be impractical if you continually forget or neglect to recharge it. Having a ChargeSpot charging system will give you one less thing to worry about, as the system is totally autonomous and there is no need to rely on you remembering to charge it.

We chose to use high resonant frequency as our means of transmitting power wirelessly to the vehicle. Resonant power transfer works by making a coil ring with an oscillating current; this generates an oscillating magnetic field. Because the coil is highly resonant any energy placed in the coil dies away relatively slowly over many cycles. If a second coil is brought reasonably near to it, that coil can pick up most of the energy before it is lost, even if it is some distance away (See Fig. 1). This method of wireless energy transfer is better than capacitive coupling like the high electric field of the tesla coil. This method also allows for more charging current and better efficiency. [1] For this project, the concept will be implemented using a PowerWheels electric toy car. Upon successful testing of our design, it can be successfully scaled to meet the charging requirements of a full size electric vehicle.

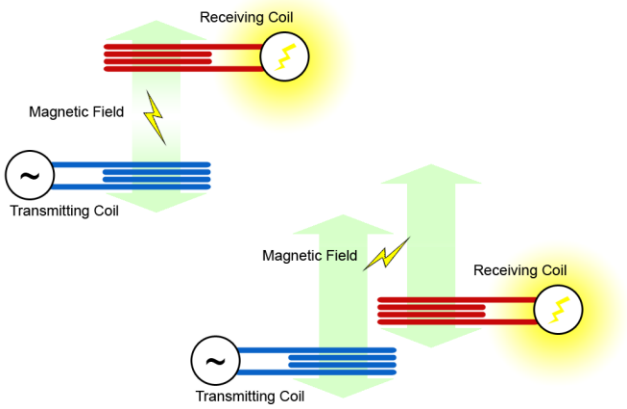

Fig. 1. High resonant frequency wirelessly transfers energy more efficiently than inductive coupling, since the receiving can be farther away and also misaligned but still transfer energy.

Efficiency plays a major part in choosing to use high resonant frequency charging. After carefully analyzing our project our main objectives include:

1) Efficiency greater than 20%.
2) Displaying battery and temperature status wirelessly on panel using LED displays.
3) Receiving coil (attached to the vehicle) must be lightweight and properly concealed as to not interfere with the normal safe operation of the vehicle.
4) Automatic shutdown when fully charged, or if temperature of battery is too hot or cold.
5) Very little user interaction, but should include a manual shutdown switch.

This charging system is designed to be purely hands free and user friendly. It charges the battery within an electric vehicle to full capacity, with a few added features. It functions as an autonomous system; everything begins when the vehicle begins to pull into a garage or designated parking area. A proximity sensor detects the presence of the vehicle as long as wireless communication has been established between the panel and the vehicle. The driver continues to proceed forward while observing the visual aid of the LED display which indicates the distance to ideal alignment of both the car coil and the ground coil.

Once in an optimal position as determined by the proximity sensor, charging mode will initiate, displaying battery and temperature status to the LED displays. If the vehicle moves, or if wireless communication is severed, the charging will automatically stop. The charging will also stop once the battery is fully charged or if the temperature is read to be too low or high for charging.

II. Overview of Ground Systems

Our ground systems consist of an MCU with XBee, the LED displays, proximity sensor, power supply, and oscillator. The microcontroller handles all the logic, and communicates with the car's microcontroller via the XBee module. Relevant data is gathered from the proximity sensor and from the car's sensors and displayed on the panel's LED bar display and 7-segment display. Power will be supplied from the outlet (AC source), rectified, and go to the oscillator and to the MCU board after being stepped down. Fig. 2 displays a block diagram for the ground systems.

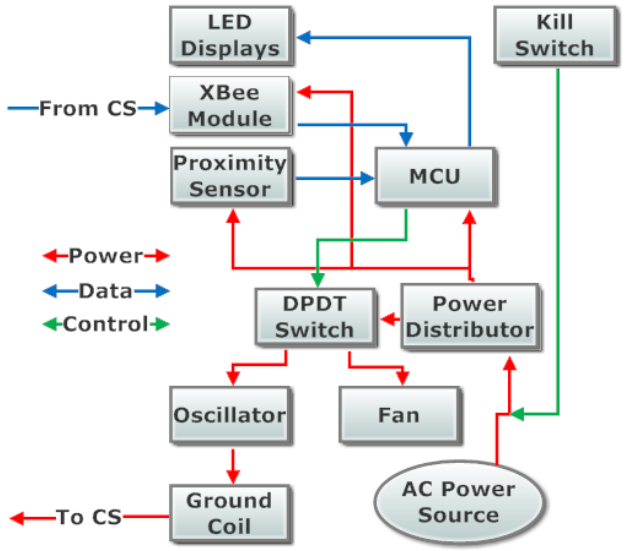

Fig. 2. Block diagram for ground systems. CS stands for Car Systems.

III. Overview of Car Systems

The car systems include an MCU with XBee, temperature sensor, charge controller, battery, and buck convertor. The microcontroller obtains data from the temperature sensor and the battery (voltage levels) and sends them to the ground's microcontroller via XBee—to prevent damage to the microcontroller, we're using a simple voltage divider circuit to read the battery's voltage. High DC voltage goes to the buck convertor from the rectifier and gets regulated to nominal voltage levels for the charge controller, which then charges the battery. The battery itself is the car's power supply for the MCU and other ICs. Fig. 3 displays a block diagram for the car systems.

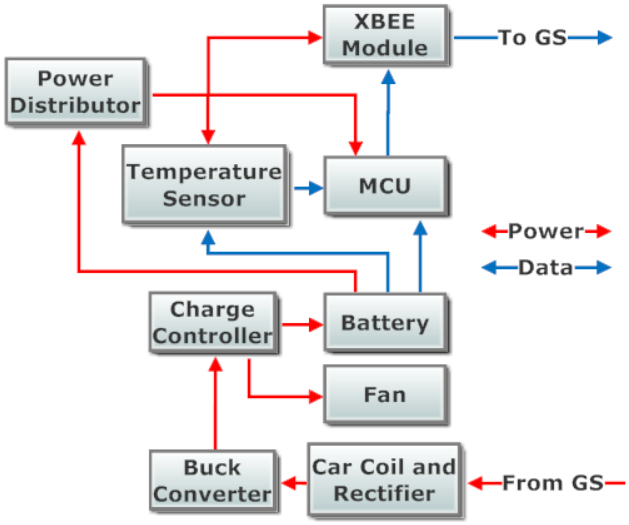

Fig. 3. Block diagram for car systems. GS stands for Ground Systems.

IV. Microcontrollers and Logic

The microcontroller being used is the Atmel ATMEGA328p with the Arduino Uno development board and Arduino IDE. The C++ coding is relatively simple, mainly relying on timing for the LED displays and XBee data transferring. The ground's microcontroller loops in an idle state until XBee communication is picked up. It then enters "mode 0" where the proximity sensor is activated and the respective data displayed. Upon reaching a pre-determined distance for ideal coil synchronization, a timer will count to a programmable value, and then enter "mode 1", also known as charging mode. The displays will be cleared and the car's microcontroller will be sending temperature and voltage data over the XBee. The LED bars will display the battery's charge level and the 7-segment will display the temperature in °F, but will switch to °C if over 99°F.

There are many ways for the charging to cease. Firstly, if the XBee communication is broken, the microcontroller will return to its idle state. If the vehicle moves from its original position, the proximity sensor will pick that up and return to "mode 0". If the temperature of the battery is too hot or too cold, the charging will cease until the temperature reaches an acceptable range. If the voltage of the battery reads full, the charging will cease. Finally, if

the kill switch is pressed, the charging will immediately cease (as well as any functions of the ground systems).

V. Sensors

The proximity and temperature sensors came with their own equations that plotted output voltage to the sensors respective output. The proximity sensor's equation accurately converted the sensor's output voltage to inches, and we used this to program our proximity display. However, the temperature sensor was too inaccurate and a new equation had to be created. A reference temperature was needed to plot against the sensor's output voltage, to which we used a DMM's temperature probe. We gathered data points from 15°C to over 150°C, plotted and obtained an equation in MATLAB (See Fig. 4), and implemented the equation into our code for the car microcontroller.

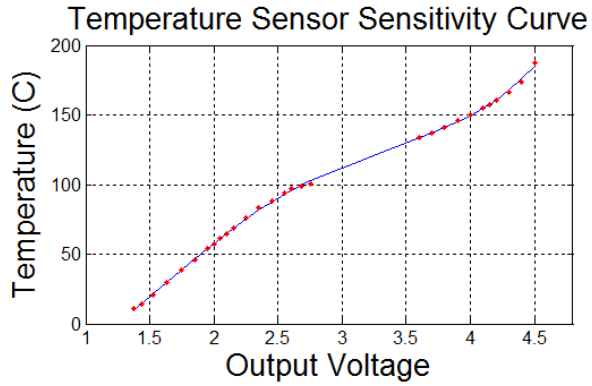

Fig. 4. Sensitivity curve of ZTP-115m temperature sensor. The red dots represent gathered-data points, and the blue curve represents the basic fitting curve obtained from MATLAB. The curve is a 4th order equation: $Ax^4 + Bx^3 + Cx^2 + Dx + E$ where the coeffs. are 4.0367, -40.878, 135.65, -107.88, and -6.6811 respectively.

We also needed to read the current battery voltage levels, and connecting the terminals straight to an analog input of the microcontroller will damage it, so we implemented a simple voltage divider (Eq. 1). Starting at the battery terminal, the circuit follows a 10kΩ resistor to point A, and from there it follows a 4.7kΩ resistor to ground. The microcontroller will be reading voltage from point A (See Fig. 5). This insures that the input voltage to the battery is less than 5V.

$$V_{in} = V_{bat} * \frac{4.7k\Omega}{4.7k\Omega + 10k\Omega} \quad (1)$$

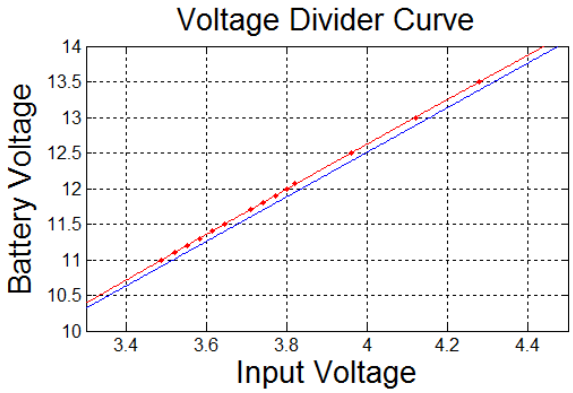

Fig. 5. Voltage divider curve using gathered-data points (red) and using the voltage divider equation (blue).

VI. Oscillator and Coils

A high frequency oscillator circuit is used to "ring" the ground system coil at a predefined resonant frequency, thus allowing wireless power transfer to the car coil. Due to part availability and design constraints the resonant frequency was determined to be 100kHz; however this design is quite sufficient for our needs. The oscillator produces a sine wave signal as seen in the Fig. 6.

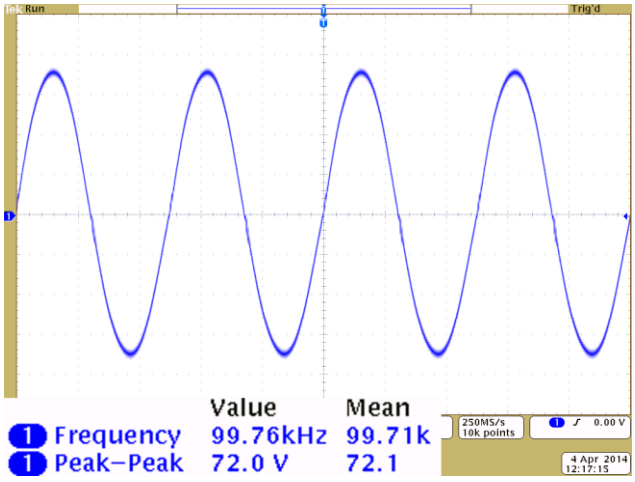

Fig. 6. Oscilloscope trace of output frequency and voltage of the transmitting coil.

A) ZVS Driver Oscillation Circuit

A modified version of a zero voltage switching (ZVS) driver circuit design was used as our oscillator. It is a very simple circuit that can oscillate a large amount of power with about 90% efficiency. When power is applied at +V current starts to flow through both sides of the primary and to the MOSFETs' drains. Simultaneously,

that voltage appears on both of the MOSFETs' gates and starts to turn them on. Because no two components are exactly alike, one MOSFET turns on slightly faster than the other and more current will then flow through the MOSFET that is on. The extra current flowing on that side of the primary steals the gate current from the other MOSFET and starts to turn it off. A condenser forms an LC tank with the primary and the voltage proceeds to rise and fall as a sinusoid. If it were not for that capacitor, the current would continue to increase until the transformer's core became saturated and the MOSFETs explode. [2]

As stated before, this oscillator involves zero-voltage switching (ZVS), meaning that the MOSFETs switch when they have zero volts across them. This is good because it allows the MOSFETs to switch when they are carrying the least power—something that, for the most part, eliminates the switching losses which generate huge amounts of heat. For this reason, only small heat sinks are needed, even when oscillating 1000 watts.

Another feature that helped us tremendously with the efficiency of our wireless design is a phenomenon known as resonant rise, which increases the voltage in the LC tank to approximately V_{CC}*π. Because of this we had to choose MOSFETs rated at 4x the input voltage being fed to the oscillator. We also had to be careful not to place them on the same heat sink as that would electrically connect both drains. For the capacitor we had to use mica or Mylar because an electrolytic capacitor in this application would explode.

The frequency of this resonance is determined by the inductance of the ground system coil and the capacitance across it. The relationship is given by Eq. 2:

$$f = \frac{1}{2\pi\sqrt{LC}} \tag{2}$$

The schematic (See Fig. 7) shows the oscillator circuit. Too much gate current can cause damage so the 470Ω resistors limit the current that charges the gates. The 11kΩ resistors are used to pull the gates down to ground to prevent the MOSFETs from latching. The12V Zener diodes prevent the gate voltage from exceeding 12volts and for maintaining constant amplitude in the oscillations. When the voltage on the opposite leg of the tank is grounded, the UF4007 diodes pull the gates down to ground; they are also capable of handling the reasonably high operating current as well as fast switching of the circuit. Finally, to improve the overall performance of the circuit, we charge the gates with V_{CC} and discharge them via the LC tank circuit using the ultrafast diodes.

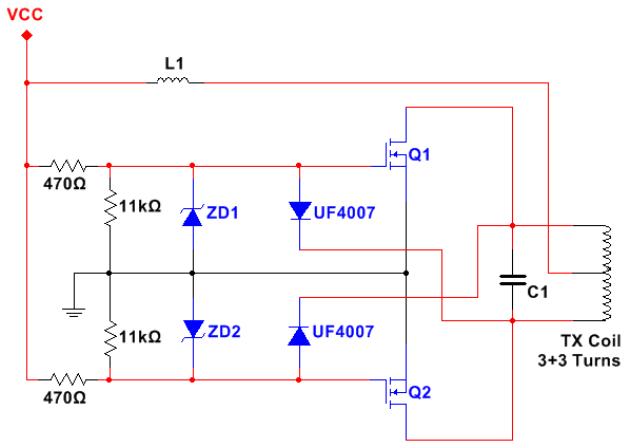

Fig. 7. ZVS Driver Oscillation Circuit schematic altered for our design.

B) Transmitting and Receiving Coils

We chose to go with a flat spiral design for both the transmitting and receiving coils. This is a space saving design and works out well if the vehicle does not have much ground clearance. The receiving coil was made slightly smaller than the transmitting coil to compensate for misalignment.

The transmitting coil is wound with a center tap and consists of 3+3 turns while the receiving coil consists of 2.5+2.5 turns. After winding the coils the inductance was determined using the LCR-816 tester in the senior design lab. In order to have maximum power transfer between the transmitting coil and the receiving coil, both tank circuit coils must be tuned to the same frequency. Our resonant frequency is designed to be 100 kHz (See Fig. 6).

The two main coils used for this design are both air core inductors wound with 10 American Wire Gauge (AWG) bare copper wires. Air core coils are relatively free of iron losses which pose a problem for coils with ferromagnetic cores. Other advantages of air core coils include the fact that its inductance is unaffected by the current it carries and this helps a great deal in our application since charging current can vary depending on battery charging needs. In addition, since we have to maintain magnetic resonance in both coils, fluctuations in inductance due to current capacity would greatly affect the efficiency of our system. The production of harmonics can also affect the sensitive electronics in our design and cause unwanted behavior. Using an air core coil prevents the production of harmonics and you also obtain a better Q-factor, greater efficiency, greater power handling, and less distortion.

There are definitely some disadvantages in using the air core coil, however, but they are overshadowed by the benefits and most of them can be tolerated without greatly affecting performance. Without a high permeability core, more turns and/or a larger coil is needed to achieve a given inductance value. In designing our coils we have to be mindful of the fact that more turns means larger coils, lower self-resonance due to higher inter-winding capacitance, and higher copper loss. Since our application uses a relatively high frequency voltage signal which generally doesn't need high inductance, this would not be a problem. The air core coil also provides the benefit of providing an external radiated field at our resonance frequency, which we want to be strong enough to be picked up by the receiving coil. Fig. 8 shows an example of the flat coil design we are using.

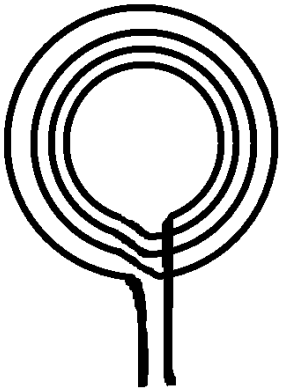

Fig. 8. Simple flat coil design; this example has 4 turns. The reason for the "X+X" terminology for the transmitting coil is due to the fact that the oscillator circuit we are using connects the inductor from the LC tank to the middle of the coils (center-tapped).

VII. Charge Controller and Buck Convertor

We used a 12V Low Dropout Voltage (LDV) charge controller to charge the vehicle's battery. The reason for using the charge controller is to monitor the amount of current and voltage coming from the receiving coil and buck convertor. This feature can help maintain battery life and performance and also eliminate some safety issues. The charge controller uses a series of P-channel MOSFET linear regulators and differential amplifier. It can handle a power supply rating at 50W (4A, 12V nominal) with an open circuit voltage of 18V to 20V. It has an adjustable output voltage and max power dissipation of 16W, with a dropout voltage of 1.25 V @ 4A. A red LED is included to indicate the power to the circuit and a green LED to indicate a fully charged or close-to-fully-charged battery. It also has reverse current protection to protect it and other systems. [3]

As for the buck convertor, an LM2596 Simple Switcher Power Convertor circuit from Texas Instruments was used to control the voltage and current provided by the receiving coil on the car system to the charge controller. It is used to step down the voltage and increase the amount of current going into the charge controller. The LM2596 series are monolithic integrated circuits that provide the entire active functions for a step-down (buck) switching regulator; it is also capable of driving a 3A load with excellent line and load regulation. The adjustable output LM2596 series operates at a switching frequency of 150 kHz which allows for smaller sized filter components than what would be needed with lower frequency switching regulators. [4]

The following (Eq. 3, 4, and 5) were used to find the output voltage and the correct sizes of components needed. [4] To find the output voltage and resistor values:

$$V_{out} = V_{ref}\left(1 + \frac{R_2}{R_1}\right); V_{ref} = 1.23V, R_1 = 1k\Omega \quad (3)$$

To find the correct inductor sizes and E.T curve provided on the data sheet:

$$E.T. = (V_{in} - V_{out} - V_{sat})\frac{V_{out} + V_d}{V_{in} - V_{sat} + V_d}\frac{1000V\mu s}{150kHz} \quad (4)$$

where V_{sat} = 1.16V and V_d = 0.5V. The last bit of that equation is a given constant for E.T (V * µs). For the feed-forward capacitor (C_{FF}) value:

$$C_{FF} = \frac{1}{31 * 1000 * R_2} \quad (5)$$

VIII. Prototype Construction

ChargeSpot's small-scale prototype involves charging a 12V PowerWheels toy vehicle. Our only small enhancement to the vehicle is a replacement of the battery to 18AH. The interior of the vehicle houses our car systems on one circuit board. The receiving coil is mounted onto the bottom of the vehicle, leaving enough ground clearance for normal operation.

The garage is mimicked for our small-scale using a wooden stand that houses the panel and transmitting coil. Within the panel lies the power supply, kill switch, cooling fan, LED displays, proximity sensor, and the

ground systems' circuit board. Protected wires run down the length of the wooden frame to where the transmitting coil is mounted. Right next to that coil lies the oscillator system and its circuit board. As soon as the vehicle comes into range, it can be driven easily over the wooden frame and transmitting coil, then ChargeSpot will begin its wireless charging.

Printed circuit boards were designed and built for two of our three major systems. The first board is for the ground systems' circuitry, involving the power distribution, microcontroller, XBee module, and LED drivers. Since we're not allowed to use production development boards in our final design, constructing the "barebones" circuit for the ATMEGA328p was completed without much difficulty, although we are still using the development boards from programming. [5]

Ribbon cables connect to the LED displays from the board, and other wires connect to a fan, the oscillator system, and input from the transformer. Our LED bar display is going to be made from scratch, using perforated boards, ribbon cable, red and green LEDs, resistors, and bottle caps. Our initial design involved an ice-cube tray, but due to light bleed-over into the next bar, we decided to use water bottle caps to seal the light in their individual bars. A major problem though is the brightness of the LEDs, specifically the red LEDs. Lowering the resistance values on each LED didn't solve the problem. Since we didn't have enough time, we couldn't switch them out with brighter red LEDs. Also, the LEDs' lighting cone is very narrow relative to how close we mounted the lights to the panel.

The second board is for the oscillator system, which is located away from the ground systems near the transmitting coil. There are two inputs— 24V and ground—and three outputs that go to the coil. The remaining car systems are placed onto a perforated board fitted for the interior of the vehicle. This board includes inputs from the receiving coil and battery, and flows through the rectifier, fan, buck convertor, charger controller, and MCU circuitry. An output is needed for the temperature sensor that is mounted near the battery's positive terminal.

IX. Power Efficiency Testing

At the time of writing this we have only been able to perform a limited amount of tests. However, over the course of constructing our prototype, we did manage to continually test our transmitting and receiving coils. Initially we were not doing so well, but once we calculated the optimal capacitance on the receiving coil, we were able to power a simple test load of three computer fans and a light bulb, all 12V peripherals. Our efficiency at the time was close to 30%. Although the wireless power transfer was working, connecting it to the charge controller showed that even though the voltage was high (>24V), the current was too low, thus the inclusion of the buck converter. We also learned that the efficiency of the coils relies heavily on the load, whether it's inductive, capacitive, or resistive. After tuning the buck converter, we were able to feed 16V @ 0.7A to the charge controller and successfully power the car systems and charge the battery (short-lived success since the charge controller's MOSFET died). Our input at the time was 24V @ 1.5A, giving us roughly 30% efficiency.

Because our test loads vary, we can show you the coils' optimal position based on the maximum power the coils can transfer, using a 12V computer fan as our test load. As you can see in Fig. 9 the energy transfer is maximized between 0 and 6 inches from the coils' origins. The power efficiency at 100% was about 15% power efficiency and the operating height is 2in.

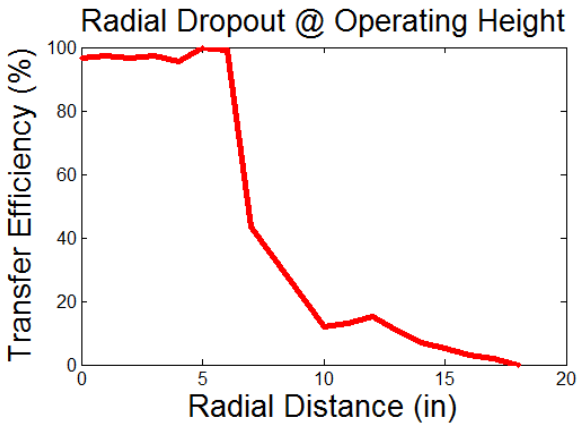

Fig. 9. This Radial Dropout curve shows at what point from the origin (x-y displacement) of the transmitting coil will the receiving coil start receiving less than maximum throughput. A computer fan load was used for this test.

Our final design will have the receiving coil roughly two to four inches above the transmitting coil, and as with the radial dropout, we tested height dropout as well (See Fig. 10). As with the Radial Dropout, the coils begin to lose efficiency around 6in from the origin, but height-wise the efficiency degrades more slowly than x-y displacement. The operating origin is where both the coils' origins are in the same spot.

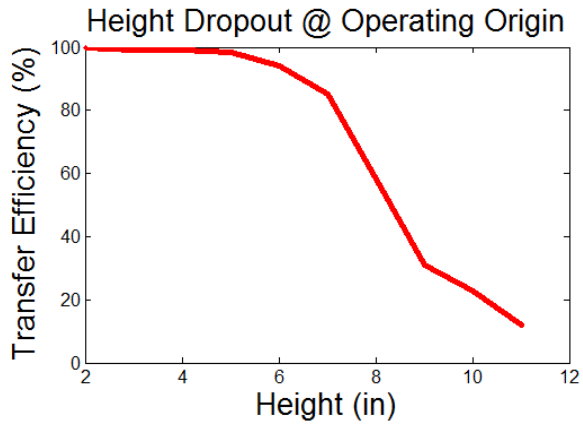

Fig. 10. This Height Dropout curve shows at what point from the origin (z displacement) of the transmitting coil will the receiving coil start receiving less than maximum throughput. A computer fan load was used for this test.

X. Project Difficulties

We have worked on this project for two semesters and have run into quite a few problems along the way. Of course, when one problem is solved, you eventually run into two more down the road. Our largest issue so far has been the charge controller circuitry, followed by some XBee communication problems, heat management, an unfinished PCB, and general miscellaneous problems.

We've had countless problems with our charge controller from the very start. However, problems designing a discrete charge controller are to be expected. We noticed that our design is very sensitive. Our first run through on a breadboard resulted in a short-lived success. The circuit was adjusted for our battery, but an accidentally short circuit caused the MOSFET to cease to function. Our second iteration worked, and so we decided to test all of our systems together. This resulted in another dead MOSFET. Our third iteration was tested with a newly-learned buck convertor circuit, also on a breadboard at the time, and everything worked. We moved our prototypes onto a perforated board, with all the other car systems connected. Everything worked perfectly, but while testing with probes, the ground probe came undone and the MOSFET died again. We replaced the MOSFET, but the charge controller still didn't work, so we moved it onto a breadboard and tested it. It worked, so we moved it back to the perforated board, and tested all the systems again. The buck convertor wasn't receiving the correct values, so we altered them, and the MOSFET ceased to work. Other than the fact that the voltage reading on the output of the MOSFET is incorrect, the MOSFET would get too hot, despite having a heat sink, and that's how we could immediately tell the MOSFET was dead.

Another major problem we faced was wireless communication with the XBee. After a lot of research, we realized we should use transparent mode on our XBee, as opposed to API mode which uses the digital I/O pins on the XBee module itself instead of just sending data in and out like we want it to. Setting the right mode and a private channel so they only communicate with each other, the XBee modules communicate flawlessly using simple serial functions. [6] The problems lie in the delays between the modules. The ground system's microcontroller requires precise timing for the LED displays, so a delay has to exist for the car system's microcontroller for sending XBee data. We're making the sending rate slightly less than the receiving (reading) rate. This way the read data doesn't fill a buffer, which creates a lag between sending the data and displaying it on the LED displays. Originally we had the sending rate higher than the receiving rate, because we used that buffer as a means to tell the ground microcontroller that XBee communication exist; however the 4+ second lag was an issue. Our fix to the problem is to use internal timers to sense an absence in XBee reception—if the timer reaches a certain value, it discontinues functions that require XBee communication. With this method, the goal is to get the sending rate very close to the receiving rate so that our timer limit isn't set too high to limit lag to a few milliseconds.

Since this is a project involving power and charging, a good amount of heat is generated throughout the systems. Our first major heat issue is on the ground systems' board, coming from the rectifier and the voltage regulators. Heat sinks are placed on the components, but a fan is necessary so one is mounted in the panel. The oscillator is also getting a little warm, even with heat sinks, so a fan may be installed for prolong usage. The car systems' board has heat sinks on key components, and a fan directly on the rectifier that converts AC from the receiving coils to DC for the other systems. As stated before, the MOSFET on the charge controller has a heat sink because it can get hot from charging at medium to high current (1A to 3A). The main issue has been getting the power for a fan if the component required it and the heat sink wasn't enough.

Another issue, although minor, is an unfinished PCB shipped to us from Advanced Circuitry. A red flag was brought up during the printing process, to which we called and discussed the issue. The representative looked at the gerber files given and we both deemed it to be correct and a non-issue. Upon arrival we noticed that the oscillator PCB had the top copper layer undone, and a small silk screen error. This issue is minor because we easily jumped

the connections and the board works fine. The other PCB from them was delivered completed. Also, our development board stopped being able to communicate with our computer for programming, so another had to be purchased. Other small issues have been with how we should place certain components, how to mount them, how to build them, etc. There haven't been any other problems we've face, other than obvious time and budget constraints.

XI. Conclusion

All in all this ChargeSpot project has been a long adventurous two-semester-long test. Each of us has learned how to communicate and work together as a team to achieve our milestones to get this project completed in time. We learned how to optimize a budget and write professional reports. One of the most valuable skills we learned was designing our own systems using new programs we had to learn. Prior classes have prepared us with the basics and fundamentals, but we had to use our own wits (and the internet) to apply them. The greatest achievement we can take away from this is knowing what we've been through—all those sleepless nights and busy weeks—and everything we've learned, including all our mistakes.

Acknowledgement

ChargeSpot would like to acknowledge the assistance and guidance of Dr. Samuel Richie and Senior Design TA Mike Tullbane. We also acknowledge the help of David Douglas and the UCF Machine Laboratory for their expertise and use of their resources. Thanks you UCF for giving us this opportunity to use our knowledge for practical engineering applications and show off our best work.

Project Engineers

Theophilus Essandoh is a graduating student of the electrical engineering department at University of Central Florida. He is going to pursue a working career in the computer engineering profession and continue studies for a Master's degree in Power Electronics.

Ryan Johnson obtained his BS in Electrical Engineering Spring 2014 from UCF. He works toward entering the engineering field, specifically in relation to designing and testing embedded systems. He later plans to work towards a Master's degree in computer science while learning more about programming and applications.

Emelio Watson is a graduating senior at the University of Central Florida and will receive his Bachelor of Science in Electrical Engineering in May 2014. Emelio currently works as an intern at Siemens Energy and plans to continue his career in the Power Generation Industry. His primary interests include power electronics, instrumentation, and audio/video apparatus.

References

[1] Wilson, Tracy V.. "How Wireless Power Works" 12 January 2007. HowStuffWorks.com. <http://electronics.howstuffworks.com/everyday-tech/wireless-power.htm> 03 April 2014.

[2] Munich, Adam. "ZVS (Mazilli) Driver" 26 December 2010. Adam's Science Website. <http://adammunich.com/zvs-driver/> 03 April 2014.

[3] Keith, Jim. "12V LDO Solar Charge Controller" 19 August 2012. ElectroSchematics.com. <http://www.electroschematics.com/6899/12v-lod-solar-charge-control/> 03 April 2014.

[4] Texas Instruments Incorporated. (1999-2013) LM2596 Datasheet. <http://www.ti.com/lit/ds/symlink/lm2596.pdf> 03 April 2014.

[5] Dattilo, Joseph. "Bareduino328: Building your Barebones Microcontroller" 29 February 2012. VirtuaBotix. <https://www.virtuabotix.com/product-bareduino328-barebones-microcontroller-guide/> 03 April 2014.

[6] Thiebaut, Dominique. "Tutorial: Arduino and XBee communication" 9 April 2012. Smith College. <http://cs.smith.edu/dftwiki/index.php/Tutorial:_Arduino_and_XBee_Communication> 03 April 2014.