

# Wireless Inductive Charging

## Tesla Unplugged

**Ari Ferro - a.ferro@utah.edu**

**Jeff Thacker - j.thacker@utah.edu**

**Zach Wilcox - z.wilcox@utah.edu**

**[http://www.eng.utah.edu/~zwilcox/senior\\_project/](http://www.eng.utah.edu/~zwilcox/senior_project/)**

**Abstract --** This paper outlines the proposed development of a wireless inductive charging system for remote controlled vehicles. The project aims to increase the viability of electric vehicles by creating a proof of concept for inductive charging while a vehicle is moving. The system will initially charge vehicles while they are stationary and record that information in a database. Eventually the project will attempt to demonstrate that a charge can be received when a vehicle is in motion over the inductive pad.

# I. Motivation

Electric vehicles are becoming more prevalent in today's society and represent a major step towards global sustainability. There are still significant obstacles to adopting electric vehicles on a large scale, including limited range and a limited number of charging stations. This project is an *attempt* to solve this problem by creating a proof of concept wireless inductive charging system that can recharge vehicles while they move. Initially this project will create stationary charging pads on which a remote control car may stop, charge, and have the charge received recorded in a database for billing purposes. Eventually the system will be adapted so that the car will be able to receive a small charge while it is moving over the charging pads.

# II. Introduction

The project is comprised of four main components as shown in Figure 1: the base station, the electric vehicles, the charging circuitry, and a computer. The base station contains the primary inductive circuit which will transfer energy to the car. The base station also communicates with a computer for authorization and data collection. The vehicle will have the secondary inductive circuit which will charge its battery.

The vehicle will inform the base station that it wants to receive a charge. This will initiate an authorization process between the base station and the computer. Data will be collected and stored in a database for analysis and visual representation.

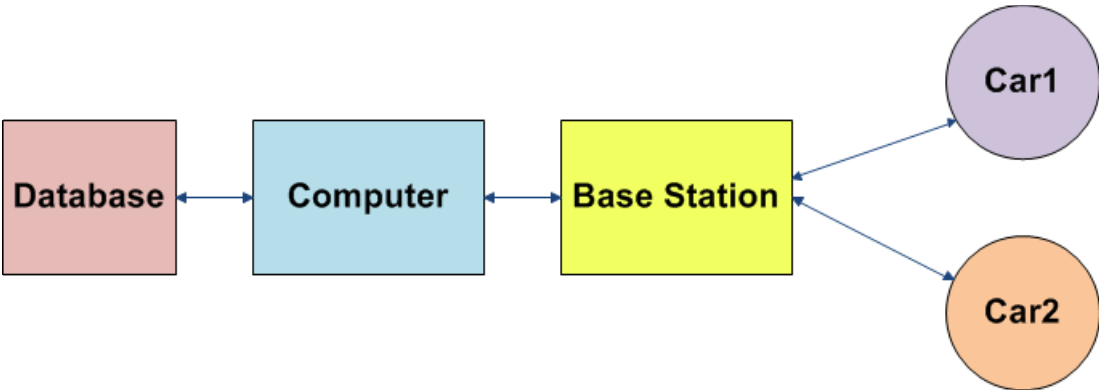


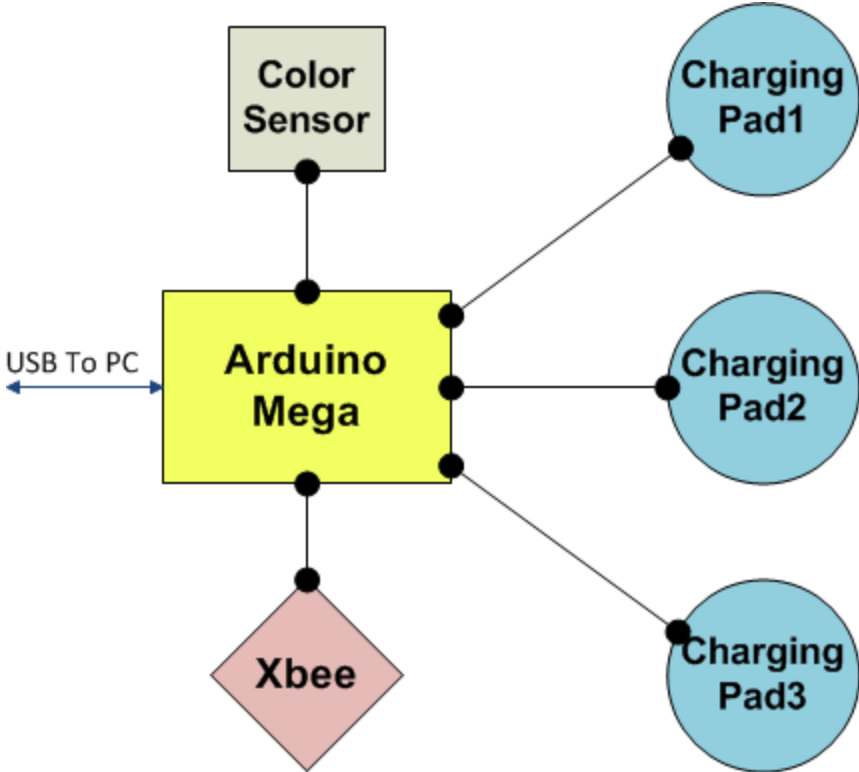
Figure 1: Block diagram of the Project

# III. Implementation Details

## III-A. Base Station

The purpose of the base station is to enable and disable the inductive charging pads as they are being used, to record information from the vehicles when they charge, and to log all recorded information in an external database. The base station will check a car's unique identifier with the database; if the car is authorized to charge, the base station will enable a charging pad when the car drives over it.

The base station will use an Arduino Mega development board as the main microcontroller. An Arduino shield with an XBee 802.15.4 wireless radio will be used to communicate with the radio control (RC) cars [1] [2]. The base station will be connected to three different inductive charging circuits that will be turned on with a single enable line from the Arduino. Power transferred will be monitored using the analog to digital converters (ADCs) that are built into the Arduino Mega. Each charging pad will use a colored light sensor to detect cars. Finally, the Arduino will send information recorded from cars to a computer via a serial connection.



Base Station

Figure 2: Base Station Block Diagram

## Arduino Mega

The development board we selected for this project is the Arduino Mega because it features enough digital input/output (IO) lines for our purposes [1]. The IO requirements are: three I2C interfaces for the light sensors, a serial communication pair for the radio, a serial connection to the computer, six ADC inputs to measure power transferred, and three digital outputs for the charging pads.

Arduino Mega Pinout	
Digital Pins:	Analog Pins:
Pin 0: Rx XBee Radio	Pin A0: Current Monitor Pad 1
Pin 1: Tx XBee Radio	Pin A1: Current Monitor Pad 2
Pin 16: SDA Light Sensor 1	Pin A2: Current Monitor Pad 3
Pin 17: SCL Light Sensor 1	Pin A3: Voltage Monitor Pad 1
Pin 18: SDA Light Sensor 2	Pin A4: Voltage Monitor Pad 2
Pin 19: SCL Light Sensor 2	Pin A5: Voltage Monitor Pad 3
Pin 20: SDA Light Sensor 3	
Pin 21: SCL Light Sensor 3	
Pin 22: Charge Pad 1 Enable	
Pin 23: Charge Pad 2 Enable	
Pin 24: Charge Pad 3 Enable	

**Table 1:** Arduino Mega Pin Usage

## XBee 802.15.4 Radio

The radio selected for this project was an XBee Series 1. This particular radio was selected because it supports a very easy to use network topology [2]. The 802.15.4 network that the XBee radios use works well for our purposes because it uses a broadcast protocol. This works well because the radio will not have to establish a 'connection' before it can be used. The base station will simply be able to communicate with any car in range.

## XBee Shield:

The XBee radio connects to an Arduino using the XBee Shield. The shield has all of the necessary hardware required to interface easily with the XBee radio. It snaps onto the Arduino using separately purchased stackable headers [3] and uses serial communication pins 0 and 1 on the Arduino by default [4]. The shield covers several other pins on the Arduino but does not use them. The pins may still be used by replacing the headers with right angle headers. The shield also has a small prototyping area that will be used for the on-car monitoring circuits.

## Light Sensors:

The Avago ADJD-S311-CR999 color light sensors selected for this project can read red, green and blue (RGB) color intensities from the ambient light they are exposed to [5]. The sensor uses I2C, commonly referred to as two wire interface. Every light sensor acts as a slave device on the same address, which means that each sensor must be on its own I2C bus.

The light sensor has twenty-two different registers that are used to configure sensitivity and

store the current sensor values. The light sensors have a special serial interface allowing each register to be read from or set individually. Because of the way the interface works, a full read of the current sensor value takes approximately 1.5ms, which is fast enough for our purposes. The pins on the color light sensors are extremely small. In order to facilitate soldering, we are purchasing the sensors on the SEN-10701 evaluation board from Sparkfun [6].

In this implementation the light sensors will be used to detect when a car is above a charging pad. Cars will be identified by a colored light on their undersides. The light sensors on each pad will be able to recognize when each car is above the pad by the color of the light.

### III-B. Car

An overall block diagram for the cars is shown in Figure 3. The cars used in this project are going to be *New Bright R/C F-150's* [7]. They are small, low cost remote control cars from Radio Shack. We will not be modifying the remote control functionality of the vehicle. We will remove the decorative plastic and add our circuitry to the existing vehicle. This includes adding our inductive charging system to the car, replacing the battery charging circuitry, adding an Arduino Uno with an XBee radio shield, and attaching several colored LEDs to the underside of the vehicle.

Cars will listen for a message broadcasted from the base station indicating its presence. All vehicles in range will receive this message. When a car receives the message from the base station, it will respond by indicating the vehicle's ID number. Upon receiving an authorized ID number, the base station will send an authorization to the car and the car will turn on its LED and drive over a charge pad. The car will use the radio to transmit data identified by its unique ID back to the base station both on a set time interval and when it has finished charging.

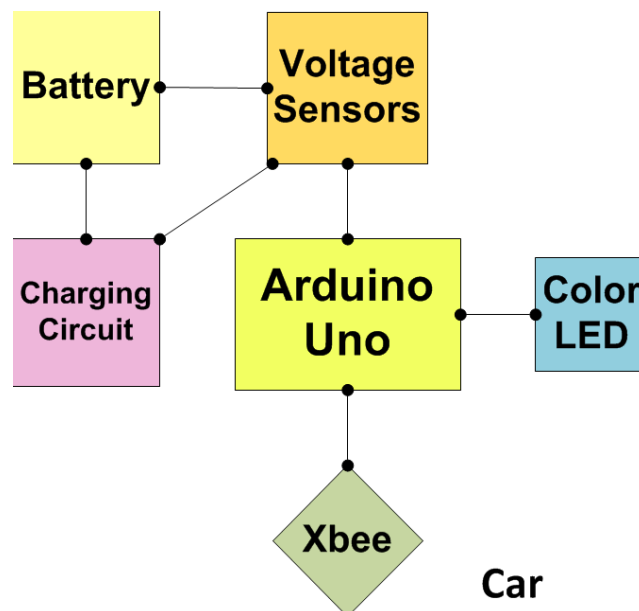


Figure 3: Car Circuitry Block Diagram

## Arduino Uno

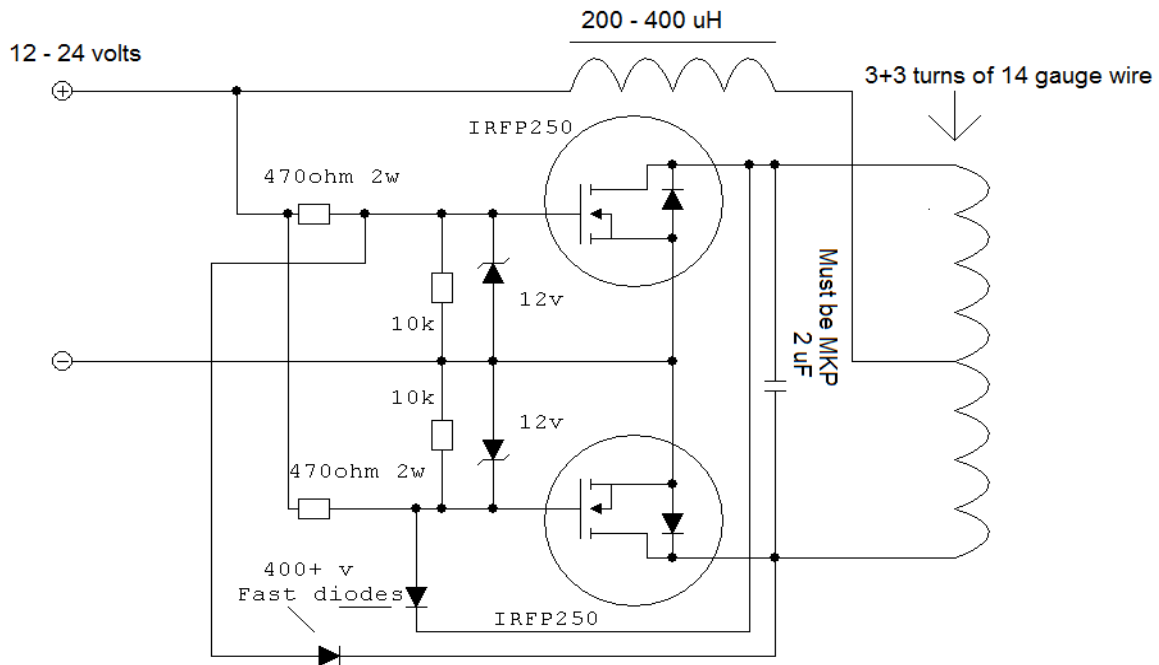
The Arduino Uno is small and low cost with enough digital and analog capability for our project [8]. It features 14 digital IO lines and five analog IO lines. It is compatible with the same XBee shield and radio used for the base station. The IO requirements include: digital output to turn on LEDs, a serial connection with the XBee radio, and three analog inputs to monitor charging characteristics. Table 2 indicates the pins that will be used on the Arduino Uno.

Arduino Uno Pinout	
Digital Pins:	Analog Pins:
Pin 0: Rx XBee Radio	Pin A0: Current Monitor Charger
Pin 1: Tx XBee Radio	Pin A1: Voltage Monitor Charger
Pin 2: LED Enable	Pin A2: Voltage Monitor Battery
Pin 4: Charge Enable Switch	

**Table 2: Arduino Uno Pin Usage**

## III-C. Charging System

Inductive charging uses an electromagnetic field to transfer energy between two inductive coils of wire [9]. Energy is sent through an inductive coupling to a battery charger. The inductive charging circuit will be split into two parts: the primary circuit which will be connected to the base station and the secondary circuit which will be part of the vehicle. The secondary circuit will be connected to the car's lithium polymer (LiPo) battery charger.



**Figure 4: Primary Circuit [10]**

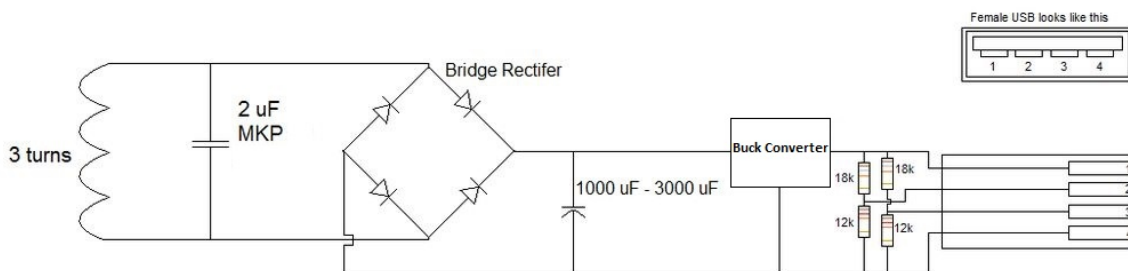
## Inductive Circuit

Inductive circuits work by inducing a magnetic field with an alternating circuit, AC, flowing through coiled wires. Next, another coil of wires induces a current when the magnetic field flows orthogonal to it.

The primary circuit takes an direct current, DC, source and transforms it to an AC signal. From there, the signal reaches the primary circuit coils and is transformed into a magnetic field. The coil on the secondary circuit induces an AC signal. That signal is rectified to DC. It is then reduced to 5 Volts through the buck converter.

There are many guides on the internet which detail inductive charging circuits. The inductive circuit in this project will be based on the guide “Wireless Power Charger!” [10] found on *Instructables.com*.

The schematic for the primary circuit can be found in Figure 4. The secondary circuit diagram can be found in Figure 5. In addition to the schematic, the primary circuit will also have a heatsink and fan to cool the mosfets. The coil in both the primary and secondary inductive circuits will be comprised of seven to nine feet of fourteen gauge wire. Experimentation will be required to determine the diameter of the loops for the desired effect.

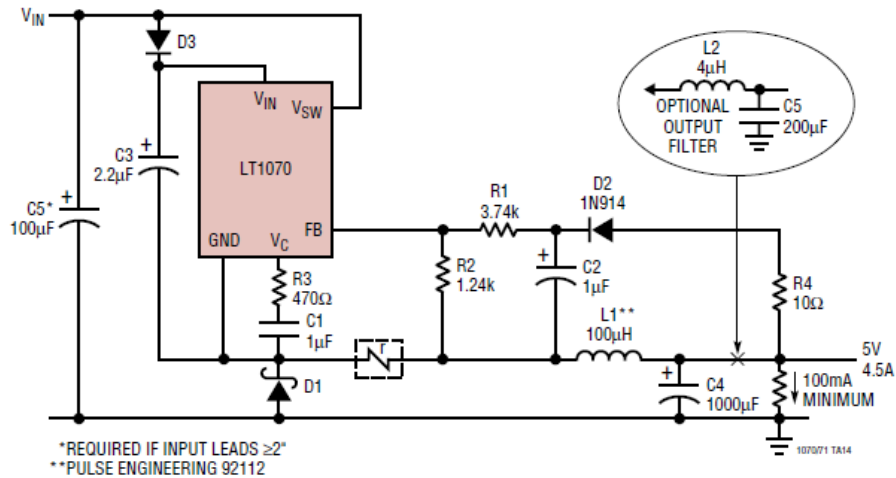


**Figure 5:** Secondary Circuit [10]

## Design Changes

The primary circuit will be powered by two AA batteries. This is slightly different than the guide’s design, which used six AA batteries. It used a LM7805 voltage regulator in the secondary circuit to limit the voltage going to the USB port. This is very inefficient because it wastes energy by dissipating excess power as heat [11]. It also limits the input voltage to the transmitter to 15 V [10]. This limitation is due to resonance rise. The resonance rise will charge the voltage regulator over what it is capable of handling. Instead we will use a boost converter. The boost converter will be used in a positive buck converter. A LT1070 boost converter will be used in the design of the buck converter circuit. Buck converters are a step-down DC to DC converter and are able to reduce the voltage with 95% efficiency [11]. The schematic for the buck converter can be found in Figure 6.

The last deviation will be to the USB connection in the secondary circuit. It was designed to charge an iPhone. This phone requires a certain amount of voltage on the data line in order to charge. The data pins for USB connectors are pin two and pin three. Sending a voltage to the data pins are unnecessary. The secondary circuit will only supply voltage to the Vcc and ground pins, pin 1 and pin 4 respectively.



**Figure 6:** Buck Converter [10]

### Battery Charger

To avoid needing to design a battery charging circuit, a LiPo battery and charger were selected. One LiPo cell supplies 3.7 V. The selected cars for the project use 5 AA batteries in series; each battery supplies 1.5 V for a total of input voltage of 7.5 V. The car has been connected up to a DC voltage supply to test the lower voltage limits. It appeared to lose all functionality below 2 V. The vehicle is able to function at 3.7 V. It, however, is unable to obtain full speed. The speed it is able to achieve with this voltage supply will be sufficient for the project's needs.

The MCP73831T battery charger is capable of supplying the battery with 3.7 V at a rate of 500 mA per hour [13]. The battery charger for the LiPo batteries connects to a USB port. This implies the secondary inductive circuit will not have to be altered for another connection. The battery charger is capable of supplying 500 mA to the battery [13].

The charger also has a 'SYS OUT' pin. According to the vendor's website, the SYS OUT allows connection from the charging circuit directly to the load without disconnecting the charger [8]. This allows the charging circuit to connect to the vehicle voltage input. Experimentation will be needed to determine if this pin will allow us to charge the battery while the battery is in use. If this is not the case, the wires from the battery to the charger will need to be spliced and connected to the vehicles' power.

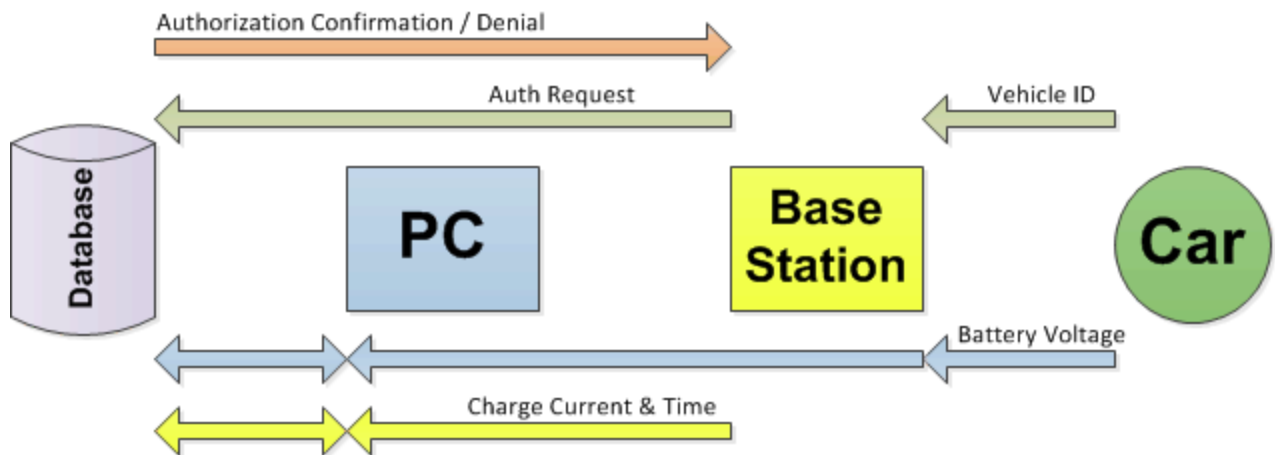
## III-D. PC Software

### Database

A database will be hosted on a local computer connected to the system. It will be used to keep track of monitored aspects of the charging system. Recorded data will include vehicle identification numbers and charging history, the amount of current provided to charge the vehicle, the length of charge time, and the battery voltage level of the vehicles. The database will



be the means of authentication whereby the system can determine whether or not to charge a vehicle. The data flow is illustrated in Figure 7. MySQL will be used to create and host the database. This database is sufficient for the basic database functionality needed for this project and the price tag of “free” is ideal.

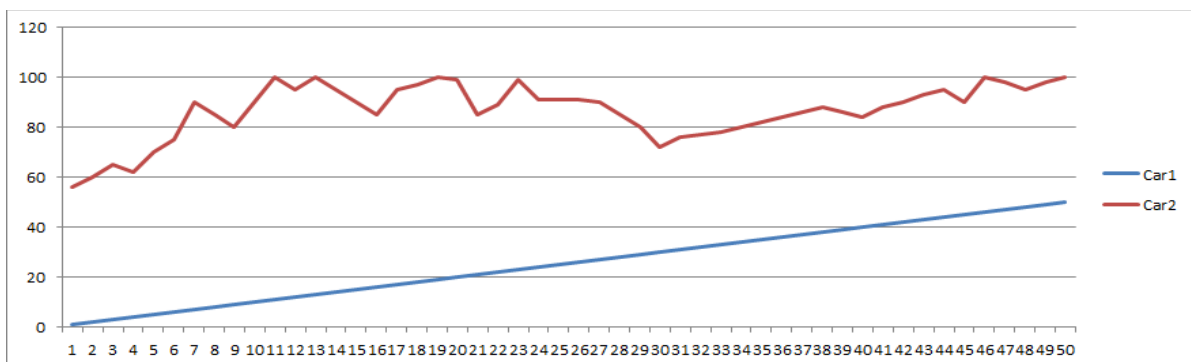


**Figure 7: Data Flow Diagram**

### Software

A software program on the host PC will manipulate the data in the database using Standard Query Language (SQL). It will be the means of entering and managing vehicle data into the database through the use of Windows forms. It will collect monitoring data from the base station via a USB serial interface. It will also receive authentication requests from the base station in the form of a vehicle identification number, search the database to determine if the vehicle is allowed to charge, and send authentication confirmation or denial back to the base station.

The software will also provide visual representations of the data recorded in the database. An example of one such visual representation is shown in Figure 8.



**Figure 8: Approximation of data visualization**

## IV. Interface Specifications

Table 3 specifies the major interfaces between different parts of the system with their respective protocols and connections:

Interface	Protocol	Device #1 Connection	Device #2 Connection
PC ← → Base Station	Serial over USB	USB port	USB port
Base Station → Charging Pad (x3)	Hard Wired	Pins 22 - 24	Enable Pin
Car → Charging Pad	Light	LED	Colored light sensor
Car ← → Base Station	XBee Wireless	(See Radio Below)	(See Radio Below)
Arduino ← → XBee Radio	UART	Pins 0-1	Rx/Tx
Base Station ← → Light Sensor (x3)	I2C	Arduino Pins 16 - 17 Arduino Pins 18 - 19 Arduino Pins 20 - 21	Pins SDA - SCL
PC ← → Database	SQL	Software	Software

**Table 3:** Interface Specifications

## V. Initial Tasking

The project has been broken up into specific tasks. The following list shows the tasks, their assigned lead and the dates and estimated times for completion: A visual illustration of this is shown in Figure 9.

#	Hardware Task	Lead	Duration	Start Date	End Date
1	Parts Acquisition	All	4 weeks	1-May-13	31-May-13
2	Charging System	Zach	4 weeks	1-Jun-13	30-Jun-13
3	Base Station Control of Charging Circuit	Zach	3 weeks	1-Jul-13	22-July-13
4	Car Battery Monitor	Ari	2 weeks	11-Aug-13	25-Aug-13
5	Car to Base LED / Color Sense	Ari	2 weeks	1-Sep-13	15-Sep-13
6	Car to Base Communication (XBee)	Ari	3 weeks	21-Jul-13	14-Aug-13
7	Base Station to PC Communication	Zach	3 weeks	1-Sep-13	22-Sep-13
#	Software Task	Lead	Duration	Start Date	End Date
8	Database	Jeff	2 weeks	1-Jun-13	15-Jun-13
9	GUI	Jeff	4 weeks	15-Jun-13	16-Jul-13
10	PC Software to Store Data	Jeff	2 weeks	23-Sep-13	30-Sep-13

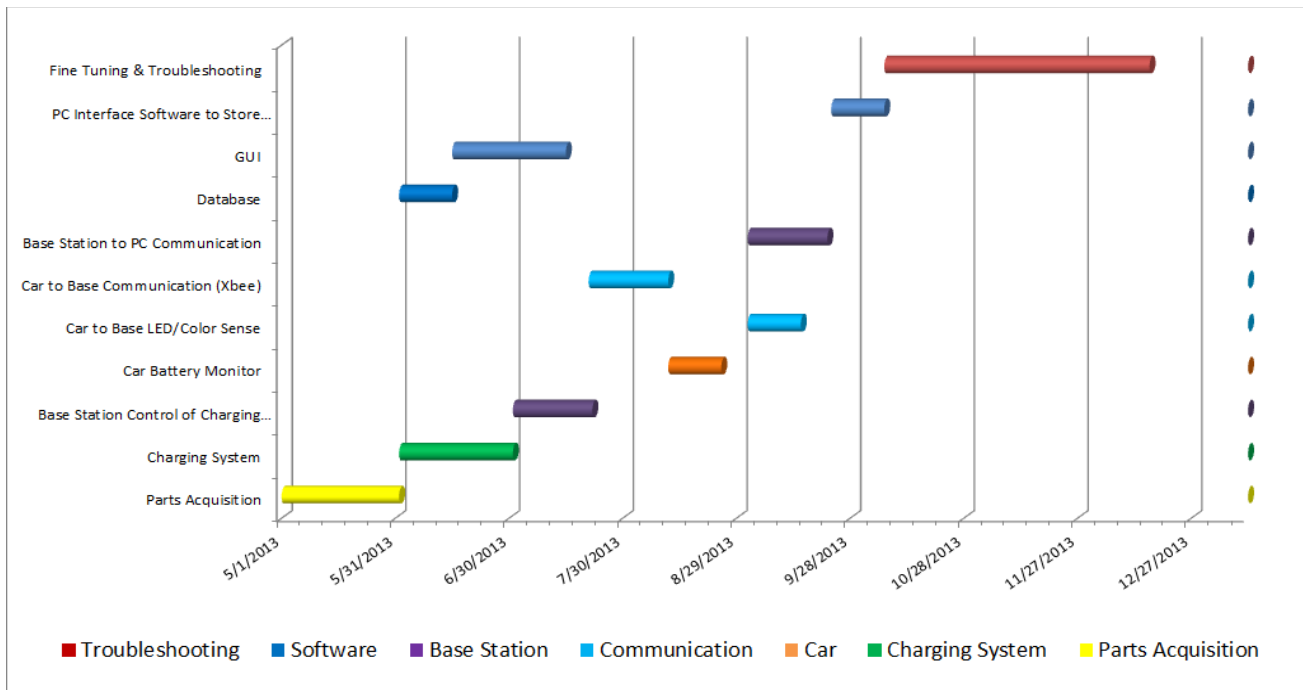


Figure 9: Tentative Task Schedule

## **VI. Risks**

### **High Risk**

The largest risk will be with the inductive circuit. The whole project revolves around being able to charge a vehicle, parked or moving, with the primary inductive circuit. We chose to rely on a design from an outside source for the analog circuits. To mitigate this risk, an inductive circuit which has a high success rate has been selected. This circuit has been demonstrated to work on the Instructable.com for an iPhone. No major modifications will be done from that circuit.

### **Medium Risk**

Being able to charge the battery while the battery is being used by the vehicle is a risk. Not all battery chargers allow this function. A battery charger that has been picked should have this feature. It is mentioned on the vendor's website that it has this ability however is not mentioned on the datasheet. A relay for switching the battery from use mode to charging mode may be added if this is found to be an issue.

### **Low Risk**

The lowest risk is the electric noise that will be produced by the inductive circuit. This may interfere with our radio communication as well as create undesirable behavior with the electric car's motor. Some simple shielding should mitigate this risk.

Using XBee wireless for communication from the vehicles is another risk. We do not have experience using XBee. Online documentation is plentiful and there are user forums to support this, but it is still mostly an unknown. Wifi can be used as an alternate method. Even if wireless communication from the vehicle becomes unfeasible, it will not be a major detriment to the project.

## VII. Bill of Materials

Item	Price	Qty	Vendor	Web Address
100 Feet 14-Gauge Wire	\$50.57	1	Home Depot	<a href="http://www.homedepot.com/p/Cerrowire-100-ft-14-3-NM-B-Wire-147-1403CR/202206557#.UVJbhWqZqU">http://www.homedepot.com/p/Cerrowire-100-ft-14-3-NM-B-Wire-147-1403CR/202206557#.UVJbhWqZqU</a>
Heatsink & Fan	\$22.99	3	Newegg	<a href="http://www.newegg.com/Product/Product.aspx?Item=N82E16835119151">http://www.newegg.com/Product/Product.aspx?Item=N82E16835119151</a>
Lithium Polymer USB Charger and Battery	\$24.95	3	Sparkfun	<a href="https://www.sparkfun.com/products/9876">https://www.sparkfun.com/products/9876</a>
Polymer Lithium Ion Battery - 850mAh	\$8.95	2	Sparkfun	<a href="https://www.sparkfun.com/products/341">https://www.sparkfun.com/products/341</a>
LT107	Free	5	Linear Technology	<a href="http://www.linear.com/product/LT1070#orderinfo">http://www.linear.com/product/LT1070#orderinfo</a>
Arduino Mega	Already Owned	1	N/A	N/A
Arduino Uno	\$29.95	2	Sparkfun	<a href="https://www.sparkfun.com/products/11021">https://www.sparkfun.com/products/11021</a>
XBee Shield	\$24.95	3	Sparkfun	<a href="https://www.sparkfun.com/products/10854">https://www.sparkfun.com/products/10854</a>
XBee Radio	\$22.95	3	Sparkfun	<a href="https://www.sparkfun.com/products/11215">https://www.sparkfun.com/products/11215</a>
8-Pin Stackable Headers	\$0.50	6	Sparkfun	<a href="https://www.sparkfun.com/products/9279">https://www.sparkfun.com/products/9279</a>
Light Sensor	\$14.95	3	Sparkfun	<a href="https://www.sparkfun.com/products/10701">https://www.sparkfun.com/products/10701</a>
MySQL Community Server 5.6.10	Free	1	MySQL	<a href="http://www.mysql.com/downloads/mysql/">http://www.mysql.com/downloads/mysql/</a>
<b>Total:</b>	<b>\$513.74</b>			

**Table 4:** Bill of Materials

## VIII. References

- [1] <http://arduino.cc/en/Main/ArduinoBoardMega>
- [2] <https://www.sparkfun.com/products/11215>
- [3] <https://www.sparkfun.com/products/9279>
- [4] <https://www.sparkfun.com/products/10854>
- [5] <https://www.sparkfun.com/products/10656>
- [6] <https://www.sparkfun.com/products/10701>
- [7] <http://www.newbright.com/product/view/id/24>
- [8] <http://arduino.cc/en/Main/ArduinoBoardUno>
- [9] [http://en.wikipedia.org/wiki/Inductive\\_charging](http://en.wikipedia.org/wiki/Inductive_charging)
- [10] <http://www.instructables.com/id/Wireless-Ipod-Charger/>
- [11] [http://en.wikipedia.org/wiki/Buck\\_converter](http://en.wikipedia.org/wiki/Buck_converter)
- [12] <http://cds.linear.com/docs/Datasheet/10701fe.pdf>
- [13] <https://www.sparkfun.com/products/9876>