AN ABSTRACT OF THE THESIS OF

Allen T. Waters for the degree of Honors Baccalaureate of Science in Electrical and Computer Engineering presented on May 28, 2010.
Title: Wireless Charging System Using Inductive Coupling

Abstract approved: ______________________________
Roger Traylor

A wireless power transfer system using inductive coupling was designed and implemented as part of a three-member project team. The system uses coupled magnetic fields at a frequency of 606kHz to transfer electromagnetic energy from a charging base to the batteries of portable devices. It succeeded in transmitting power to three unique devices simultaneously, with an efficiency of 10.6%. The system tested safely for users, target devices, and neighboring electronic devices. It features a serial data connection between the base and the target device, which aids in device recognition and power conservation. This document details the design process, implementation, results, and future direction for the project. Technical details including schematics, code, and parts lists are appended.

Key Words: wireless, energy, power, transfer, induction

Corresponding e-mail address: watersal@onid.orst.edu

Wireless Charging System Using Inductive Coupling

by

Allen T. Waters

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Electrical and Computer Engineering
(Honors Scholar)

Presented May 28, 2010
Commencement June 2010

Honors Baccalaureate of Science in Electrical and Computer Engineering project of Allen T. Waters presented on May 28, 2010.

APPROVED:

Mentor, representing Electrical and Computer Engineering

Committee Member, representing Electrical and Computer Engineering

Committee Member, representing Electrical and Computer Engineering

Head, School of Electrical Engineering and Computer Science

Dean, University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

Allen T. Waters, Author

ACKNOWLEDGEMENT

I would like to thank team members Tawalin Opastrakoon and Benjamin Waters for their contributions to the project. Also, thanks to ECE44x instructor Don Heer and teaching assistant Tim Marr for their mentorship throughout the process. I would like to thank Intel for their sponsorship and financial contributions that made the project possible. Finally, I would like to thank mentor Roger Traylor and committee members Don Heer and Dr. Albrecht Jander for their contributions in the review process.

CONTRIBUTION OF CO-AUTHORS

As this paper reflects on the work of a senior design group project, some of the material contained within has been worked on by all members of the group. This includes work done by both Tawalin Opastrakoon and Benjamin Waters for the final specification document.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

LIST OF APPENDICES

LIST OF APPENDIX TABLES

1. Introduction

A. Motivation

A recent trend in power supply design is the development of charging systems that are capable of wireless power transfer. This means that the power supply is not plugged into the device being charged (though it will be in close proximity, even physical contact). Wireless power transfer technology has existed for a long time; however, recent advances have allowed it to become more practical, and recent interest in the consumer market has brought it to the center of attention [1]. The goal of this project was to create a wireless power transfer system that is capable of transmitting power with 50% efficiency (power absorbed by the inductive pickup per unit of power expended by the transmitter).

Since most consumers own several different handheld devices, such as cellular phones, pagers, PDAs, or MP3 players, it is expected that the charging system will be compatible with a variety of devices. Versatility is important for developing a marketable, competitive system. Furthermore, the system should be capable of charging multiple devices (whether identical devices or otherwise) simultaneously. To handle this requirement the charger will have several designated areas for devices to be placed on the charging surface. These "hot spots" reduce the freedom to place the devices anywhere on the surface, but eliminate the need for the charger to track the physical location of each device and transmit the correct amounts of power in the proper directions, an extremely complicated problem.

In the interest of efficiency, the handheld device will be capable of monitoring its own level of charge. It will signal back to the charger base when the battery is full, so that the charging system can cut off power transmission to avoid wasting energy. The data transfer to signal the device ID number and charge status will occur over a serial connection between microcontrollers mounted on the target device and on the charger base.

The interface should require minimal modification of the device to be charged; considering that the devices are handheld, a bulky modification would be unacceptable. Additionally, the system shall be robust and safe. Safety encompasses interactions with both humans and with other electronic devices that may be present in the environment. It shall fulfill the system requirements over 20 hours of usage during testing.

B. Course Requirements

The project was completed in fulfillment of the ECE44x "Engineering Design Project" courses, a three-term sequence of classes required for all Electrical and Computer Engineering undergraduates. As a result the project was completed according to the timeline used by those courses, described briefly below:

Fall Term: Project is designed with a top-down approach; each block is repeatedly designed, reviewed, and redesigned until satisfactory.

Winter Term: Implementation begins. By the fifth week of the term each individual functional block is expected to be fully prototyped and successfully tested. By the tenth week the entire system is expected to be assembled, and successfully tested.

Spring Term: Improvements are made to the already fully-operation system. Generally these are features that were not included in the original system requirements but are later found to be useful.

The structure of the ECE44x course sequence had many benefits, including the strict project timeline, the availability of hardware and software resources, and the support offered by the teaching staff, mentor, and customer.

C. Team Members

Other team members, working as peers, for the project included Tawalin Opastrakoon and Benjamin Waters, both seniors in Electrical and Computer Engineering. After working together to complete the top-level block diagram and interface definitions, the remainder of the project was carried out largely independently. This was achievable because as long as each member designed and implemented their blocks in accordance with the interface definitions, the fully assembled system would be functional.

The responsibilities that I undertook for the project consisted of the two microcontroller blocks (one mounted on the base of the charging system, one mounted on each of the portable devices), and the user display.

Tawalin Opastrakoon was responsible for the power transmitter and pickup blocks, as well as the power regulation circuitry on each device.

Benjamin Waters was responsible for the current-sensing circuitry mounted on each portable device, the power supply block for the charger base, and the chassis for the charger base.

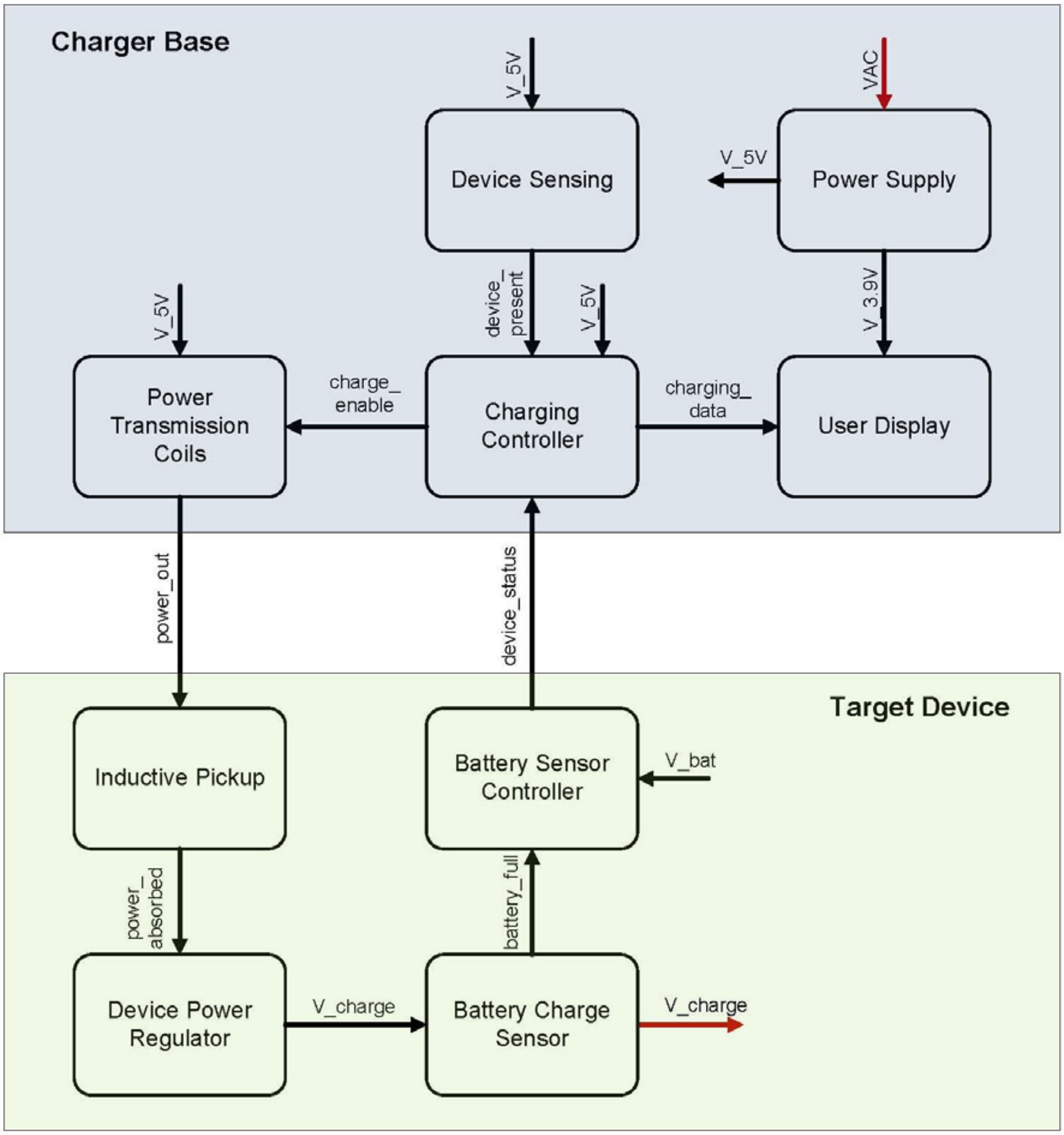
D. Sponsor Involvement

The project was generously sponsored by Intel. Though Intel did not provide mentoring or technical insight, they offered financial support of up to $1000, far beyond final cost of parts and exposition materials.

2. Design

A. Top-Level Block Diagram

Figure 1. Top-Level Block Diagram



B. Interface Definitions

Table 1. Top-Level Interface Names and Definitions

Signal Name	Type	Properties
VAC	Electrical	120V AC 60 Hz Standard 3-conductor power cord
V_5V	Electrical	5V DC Max 1.5A power bus
V_3.9V	Electrical	3.9V DC Max 0.5A power bus
device_present	Electrical	5V DC Low: device is present High: device is absent
charging_data	Electrical	5V DC 11-bit parallel signal 8-bit control for 7-segment displays 2-bit 7-segment select 1-bit full/empty
charge_enable	Electrical	1-bit signal High: power transmission enabled Low: power transmission disabled
power_out	Electrical	Electromagnetic radiation Frequency: 606 kHz Max 5V per charging port Max 500mA per charging port
power_absorbed	Electrical	Max 5V DC (at 100% efficiency) Max 500mA
V_charge	Electrical	Voltage regulated to device-specific DC value Max 5V DC (at 100% efficiency) Max 500mA
battery_full	Electrical	Analog DC voltage Range 0V to 4.3V High: battery depleted Low: battery full
device_status	Electrical	5V DC USI-SPI serial communication 256 kHz data rate

C. Technology Review

During the design process, comparable technology was surveyed in order to properly judge what methods and specifications were reasonable. A study of eight wireless power transfer systems is summarized in Table 2.

Table 2. Technology Review

Product Name	**Price**	**Charging Technology**	**Physical Interface**	**Number of Compatible Devices**	**Efficiency**	**Distance from Device**	**Max Power Transfer**	**Dimension**
WildCharge PowerDisc [2], [3]	Mat: \$79.99, Sleeve: \$34.99	Electrical contacts (no EM induction)	Replace battery cover with sleeve, fitted with electrical contacts	8	~100%	Physical contact	15 W	20cm x 16cm x 1.7cm
eCoupled Splashpower [4], [5], [6]	Base: \$100, Dongle \$20	Electromagnetic induction	Either manufacturer includes receiver in device, or customer attaches dongle	8 partner companies (including Motorola and Texas Instruments)	98%	2 cm maximum	1.4 kW	5" H x 7" W x 9" L
MIT WiTricity [1], [7]	N/A	Electromagnetic induction with self-resonant coils	Hardwired to receiver coil	N/A (proof of concept with lightbulb)	45%	2 m	60 W	Two 60-cm coils
Powermat [8]	Mat: \$99.99, Sleeve: \$29.99-39.99	Electromagnetic induction/RFID Handshake	Powermat receiver casing, USB tips	4 (3 wireless, 1 USB)	~100%	Physical contact	15 W	0.625"x12.25"x4.56"
U. of Tokyo Wireless Transmission Sheet [9]	~ \$100 / m^2	Electromagnetic induction, Inkjet printed organic components (printed organic transistors and Plastic MEMS switches)	Need power-harvesting coil in devices	Unlimited	62.30%	0-5 mm	29.3 W	210x210x1 mm, 50g

Product Name	**Price**	**Charging Technology**	**Physical Interface**	**Number of Compatible Devices**	**Efficiency**	**Distance from Device**	**Max Power Transfer**	**Dimension**
Duracell myGrid [10], [11]	$79.00	Electromagnetic induction	Need Duracell Power Sleeve/Clip	4	~100%	Physical contact	15 W	6.75"x8.5"x 0.75", 4.0 oz
Palm Touchstone [14], [15]	$69.99	Electromagnetic Induction	Replace back cover, magnetic charging base holds phone in place	1	???	Physical contact	2.5 W	1.9" x 3.3" x 6.1"
Tekno Wii InCharge [12], [13]	$38.97	Electromagnetic induction	Replace battery pack with lithium polymer battery of same size, and place on docking station	1	???	< 1"	5 W	2.5" x 4.25" x 7.0"

According to this review of current wireless technology, the favored power transfer method is electromagnetic induction. A base station radiates an electromagnetic field, which is received by a unit attached to the target device. However, design features varied among the different products studied; the most advantageous features stood out as possible design goals for our project.

First, a well-known research group at MIT used strongly coupled electromagnetic resonators, allowing the source resonator to transmit energy to the receiver on a very specific frequency, without interfering with other electromagnetic waves in the environment. Other products, including the Powermat and Touchstone, use magnets or grooves in the chassis to force the device into alignment with the transmitter in the docking station- this ensures more efficient charging. Finally, an RFID handshake between target devices and the charging base is a feature implemented by the Powermat to increase power efficiency. This handshake instructs the charger to source only the power needed for that specific device, cutting losses, and also when to stop a charging process (when full charge is detected).

Clearly, most products attempt to minimize the distance between the charger and the target, to maximize the efficiency of the power transfer.

D. Needs Identification

Based on the technology review and meetings with our ECE44x mentor and customer, the following system requirements were identified for the project. This list of eight requirements guided the project throughout the design process.

- **Functional**

  Though this may be taken for granted, a working prototype is necessary to satisfy the customer. It must transmit power wirelessly.

- **Safe**

  The completed system should pose no risk to either the user, the device being charged, or any other devices in the environment. Risks include electric shock, damage the device being charged, or damage to other electronic devices in the environment.

- **Low cost**

  Though our client values this less than other needs, the cost (charger base and modifications to a single target device) shouldn’t exceed $150.

- **Monitoring capabilities**

  The customer desires some mechanism for displaying what devices are detected, and whether they are fully charged.

- **Robust**

  Over the course of a 20-hour testing period, the charger should neither be physically damaged, nor should the efficiency diminish.

- **Efficient**

  The system should operate with at least 50% power efficiency, to remain competitive with wired charging systems. This efficiency is calculated as the ratio of the power absorbed by the inductive pickup block on the target device to the power consumed by the power transmission block on the charger base:

$$efficiency = \left| \frac{P(inductive\ pickup)}{P(transmission\ coils)} \right|$$

Thus the efficiency measurement only considers the wireless power transfer from transmitter to receiver, and not any power losses in regulation, control signals, the power supply, etc. Power consumed by the transmission coils is measured by hardwiring the charge_enable control signal to the 5V power bus (thus isolating the transmission coils block) and measuring the average current flow from the 5V power bus. The product of the average current with the 5V bus is the power consumption, when transmitting.

Power absorbed by the inductive pickup is calculated by measuring the output current and output voltage to the device power regulator block. It is necessary that the block still be attached to load the output. The measurements are performed at the power_absorbed interface.

- **Versatile**

  Considering that a consumer will have many handheld electronic devices, the system should be compatible with at least three different target devices.

- **Charging multiple devices simultaneously**

  Similar to the need for versatility, the system should be capable of supplying power to at least three targets (not necessarily identical) simultaneously.

3. Implementation

A. Schematics

Figure 2. Device-Side Schematic

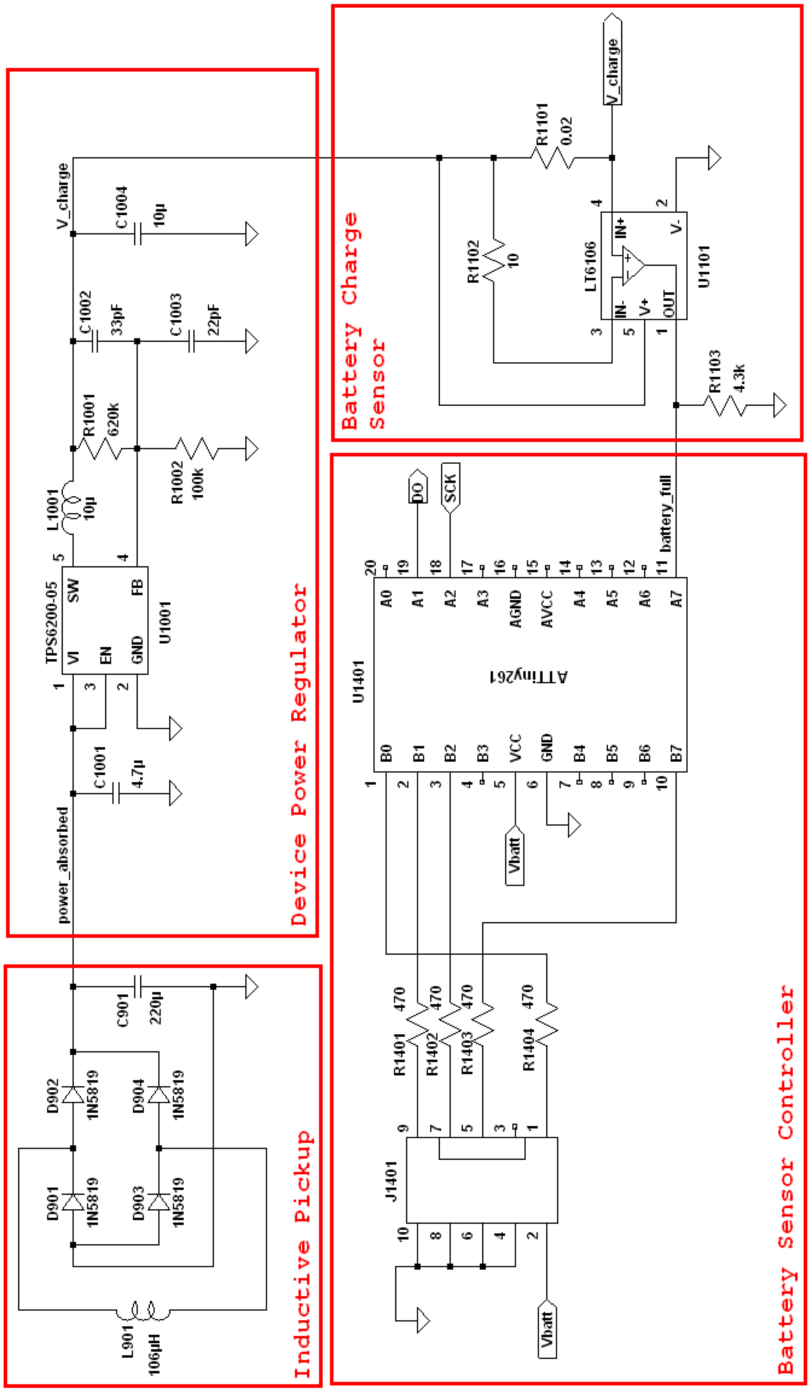
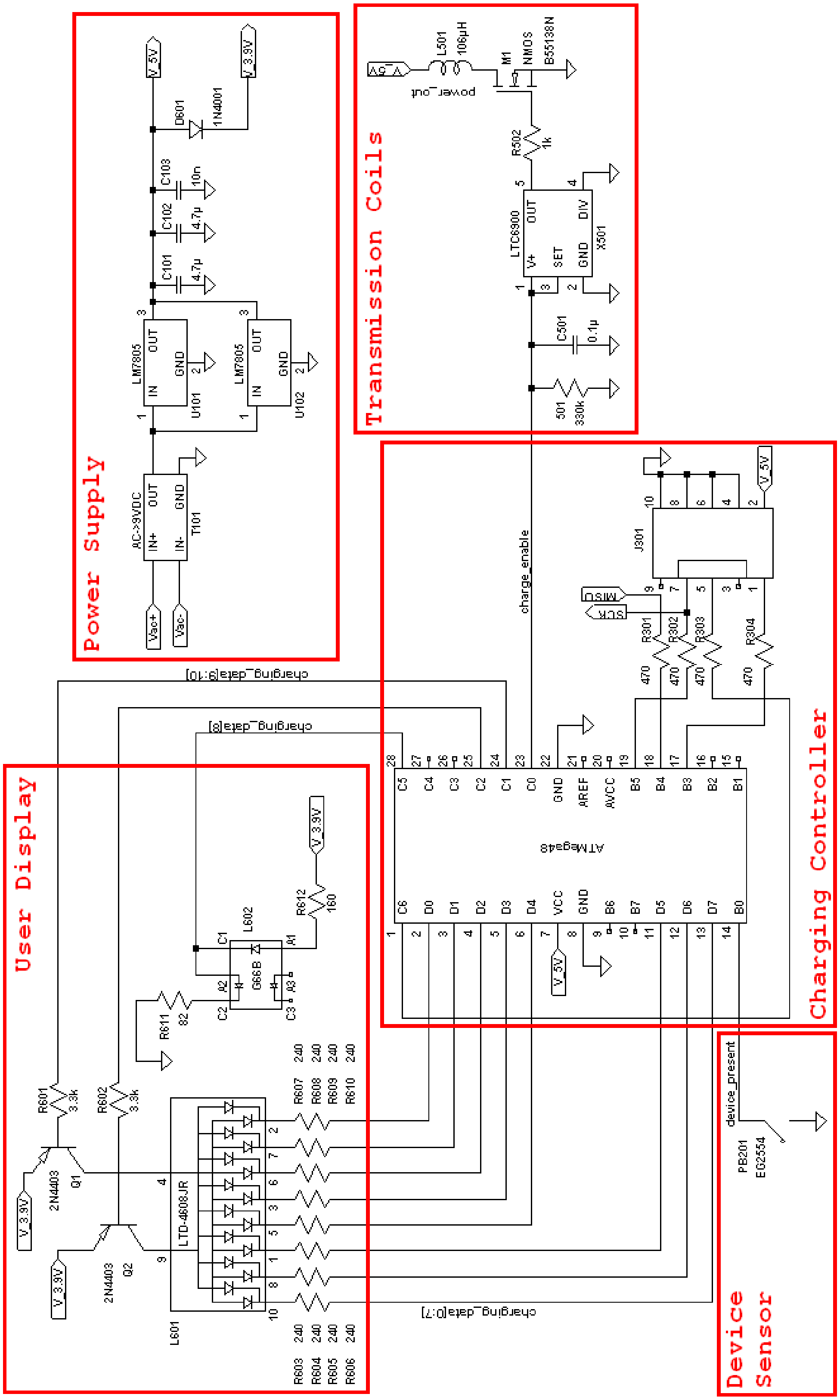


Figure 3. Charger Base Schematic



B. Sub-block Descriptions

Parts lists and budgets for each of the following blocks are included in Appendix C. This section shall simply provide a technical description of how the blocks function.

Power Supply

An AC-DC wall transformer outputs 9V at a maximum of 1.5A. This is further regulated by a pair of LM7805 voltage regulators in parallel, delivering 5V to all blocks within the charging base. It is filtered by capacitors C101-103 to reduce any noise.

Device Sensor

Devices whose batteries are completely discharged will be unable to send a valid device ID to the charger base in order to request power. Therefore an alternative device sensor is available, a simple pushbutton. When pressed the system will send power for 5 minutes without requiring a valid device ID. This should be sufficient to charge the device battery to a level at which it will begin sending USI-SPI data packets.

User Display

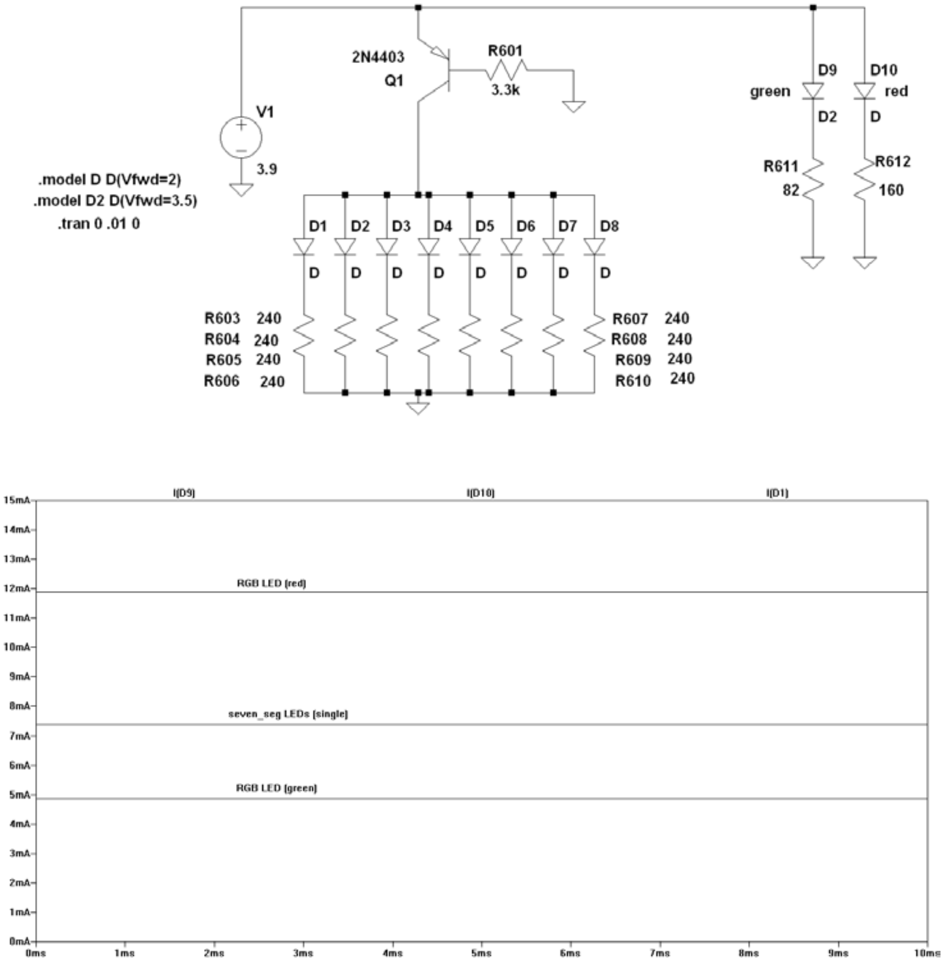
For each of the three charging ports on the base, the user display consists of a two-digit seven-segment LED display and a single red-green-blue LED. The seven-segment display, under the control of the Mega48 charging controller, is blank when no power is being transferred. When a device is being charged, it displays the device ID (or “On” when no valid

ID is available). The RGB LED is green while power is being transferred, and is red otherwise.

Validation:

The following schematic and simulation results verify that the user display block sources sufficient and safe amounts of current to each of the LED components.

Figure 4. User Display Subcircuit and Simulation Results



When active, each of the seven segment LEDs passes 7.4mA, below the nominal rating of the components datasheet but (as evidenced by the working design) sufficient to illuminate the LED. This is also well below the maximum current an IO pin on the microcontroller can sink (40mA according to datasheet). In total, the seven segment display could consume up to 59.2mA of current, which is safe considering that an IO port can sink up to 150mA.

The green segment of the RGB LED passes 4.9mA and the red segment passes 11.9mA. These are below the nominal 20mA (see the LATB-G66B datasheet), but by inspection of the final design it is sufficient to illuminate the LEDs.

Charging Controller

Each port on the base contains an Atmel Mega48 microcontroller. This uC, as shown in Figure 3, is the common connection between all other blocks in the charging base.

Pin B0 is the active low input from the device sensor, a simple pushbutton that must be debounced in software.

Pins B3 (MOSI), B4 (MISO), B5 (SCK) and C6 (reset_n) are used for the AVR low-voltage serial programming interface.

In connection to the user display: pins C2:1 select (active low) which seven-segment display is illuminated; pin C5 toggles the status LED (high for green, low for red); pins D7:0 turn on different portions of the seven-segment displays (active low).

The USI-SPI connection with the battery sensor controller utilizes pins B4 (MISO) and B5 (SCK). Since the Mega48 is configured as the master, it controls the clock as an output and the MISO pin is, of course, an input.

Charging Controller Code

The code written for the charging controller is available in Appendix A. In summary, the microcontroller loops indefinitely while writing the control bits for the user display and power transmission coils. Every 15 milliseconds it performs an interrupt service routine (ISR) that checks the device sensor and decides whether power should be sent or not. Every 64th time the ISR executes (i.e. every 1 second), it reads the SPI input. A valid SPI data packet contains a target device ID and a full/discharged flag, which the uC also uses to determine whether power should be transmitted.

Power Transmission Coils

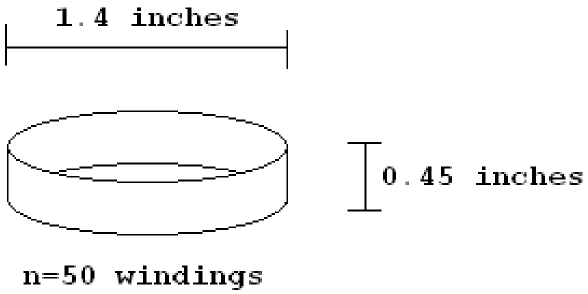
The transmission coils are driven with an AC signal, generated by a low-power LTC6900 oscillator. When the control signal from the charging controller activates the oscillator, the output signal oscillates at a frequency of 606kHz. This is determined from the datasheet as follows:

$$f_{osc} = 10MHz \bullet \left(\frac{20k}{N \bullet R_{SET}} \right) = 10MHz \bullet \left(\frac{20k}{1 \bullet 300k} \right) = 606kHz$$

R_{SET} is R501 in the power transmission coils schematic (Figure 3), and because the DIV pin on the oscillator is grounded, N=1. This output signal directly drives the hand-wound inductive coils. The concentrated flow of current around the coil produces a magnetic field directed normal to the plane of the loops.

Figure 5 illustrates the inductive coils used on both the transmitter and receiver:

Figure 5. Inductive Coil



The coils are wrapped around a hollow ferromagnetic core, in a solenoid shape. 30-gauge copper wire with an enamel covering (for insulation) is used for the windings. Note that the two coils will ideally only be separated by the thickness of the plastic chassis during transmission, a distance of 3/16”.

Discussion of solenoids

Of course, the design of the inductive coils is crucial to the efficiency of the wireless power transfer. There are several properties with which we are concerned: size, shape, wire gauge, core, and number of turns. It is necessary to discuss how these properties may be manipulated to optimize the efficiency of our power transmission, regardless of whether the implemented design was optimal.

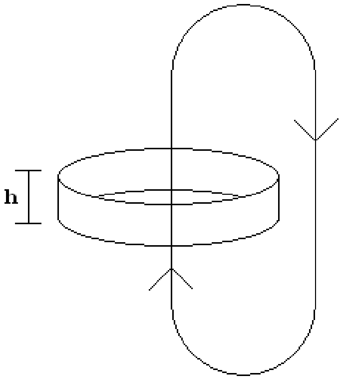
First, let it be assumed that efficiency is improved by decreasing the resistance of the inductive coils. Any resistance will only dissipate power as heat.

Second, recall Ampere’s Circuital Law:

$$\oint_C \mathbf{B} \cdot d\ell = \mu_0 I_{enc}$$

where **B** is the induced magnetic field, μ_0 is the permeability of free space, and I_{enc} is the current enclosed by closed path C. Evaluating this integral around a closed path enclosing the inductive coil:

Figure 6. Amperian Loop Around Solenoid



$$B = \mu_0 \mu_r \frac{NI}{h}$$

where N is the number of turns, h is the height of the solenoid, μ_r is the relative permeability of the core material, and B is the magnitude of the induced magnetic field. A greater magnetic field will induce more current in the receiver coil, so the ideal design of the coils should maximize B in the preceding discussion without unnecessarily increasing the coil resistance.

Clearly, height h should be minimized and μ_r should be maximized. The former is achievable by minimizing the wire gauge and by simply winding the solenoid as tightly as possible. The latter requires that a solid ferromagnetic core is used, such as nickel or ferrite (material cost must be considered here as well).

While increasing N will create a larger magnetic field, it also increases the internal resistance of the coils and more power will be lost to heat. Similarly, a thinner wire gauge will allow a smaller solenoid height but will increase the resistance per unit length of the coils. These two properties should have been experimentally tested more thoroughly in our application; it was poor engineering to select 50 turns of 30-gauge wire without more experimentation.

The circular shape of the solenoid creates a uniform, symmetrical magnetic field. Supposing that the solenoid were not circular (square, for example), the induced magnetic field would still be uniform. However, the loss of symmetry would require that the receiver coil be oriented very specifically (aligning the four corners) in order to maximize the inductive coupling between the two coils. This is an unnecessary constraint that is avoided with the circular solenoid, due to its rotational symmetry.

The diameter of the coil is constrained both by the physical limitations of the available area in the chassis, and by efficiency concerns. For efficiency, a larger coil is easier to align (improving efficiency) but increases the coil resistance. Again, more thorough experimentation would be necessary to determine what diameter is optimal.

This discussion expresses concern over the power loss through the inductive coil due to its internal resistance. Calculating this loss in the current design:

$$
\begin{aligned}
&P = I^2 R = I^2 \left(N \times \pi \times d_{coil} \times \frac{R}{length} \right) \\
&P = (49.2mA)^2 \left(50 \times \pi \times 1.4'' \times \frac{103.2\Omega}{kFt} \right) \\
&P = (49.2mA)^2 (1.89\Omega) \\
&P = 4.58mW
\end{aligned}
$$

Since this power consumption is fairly negligible compared to the total power consumed by the transmission coil block (measured: 246 mW), a final design decision would not weight this concern as high as others.

In summary, a more thorough solenoid design would minimize the solenoid height, increase relative permeability with a solid ferromagnetic core, and maintain the circular shape. It would increase the number of turns and coil diameter, and decrease the wire gauge until coil resistance became a design concern, tested experimentally.

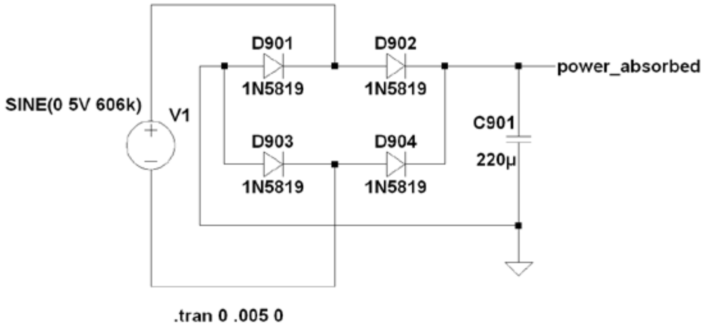
Inductive Pickup

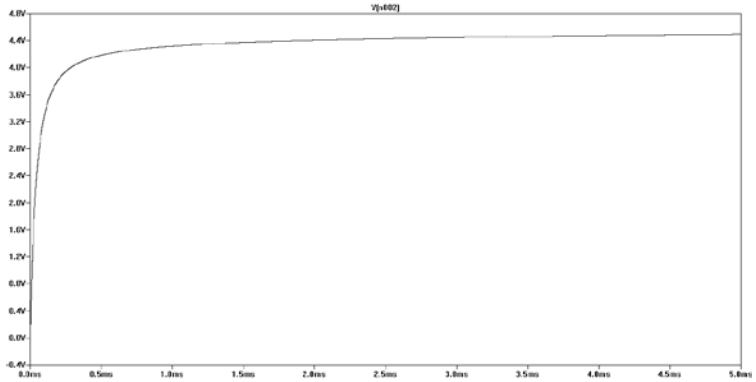
The pickup is inductively coupled to the transmission coils, meaning that the hand-wound inductive pickup coil is positioned close to and aligned with the transmission coil. Then the magnetic field produced by the transmitter passes through the dense loop of wires, inducing an electrical current. By maximizing the inductive coupling of the pickup coil in this block with the transmitting coil, the magnitude of the induced current is optimized. The four diodes in the schematic form a full-wave bridge rectifier, and the capacitor C902 filters the output voltage into a constant DC signal.

Validation:

The following subcircuit schematic and simulation results illustrate that the inductive pickup block correctly rectifies the induced AC signal (replacing the inductive coil with an AC source).

Figure 7. Inductive Pickup Subcircuit and Simulation Results





Device Power Regulator

After receiving and rectifying the transmitted power, it must be regulated to a voltage level appropriate for the device battery that is being charged. This block uses the TPS62200 switching voltage regulator to step-down the voltage level, adjustable to the device-specific needs. Many of the component values are specified in the typical application of the IC, found in its datasheet. All components are fixed except R1001 and C1002 (referred to as R1 and C1 in the datasheet). This resistance is varied to match the output voltage specification, defined by:

$$V_{out} = 0.5V \times \left(1 + \frac{R1}{R2}\right)$$

where R1002=R2 is fixed at 100kΩ. The capacitance C1002 is set according to the datasheet by:

$$C1 = \frac{1}{2 \times \pi \times 10kHz \times R1}$$

By correctly calibrating these values, the proper battery charging voltage is supplied to each device.

Validation:

The regulator uses the typical application of the TPS62200 switching voltage regulator; details available in the datasheet.

Battery Charge Sensor

In order to detect when the device battery is fully charged, the battery charge sensor converts the input current into an analog voltage and sends it as an input to the battery sensor controller. The block is effectively just an op-amp, the LT6106 current sensing IC. The input current creates a small voltage drop across the 0.02Ω resistor, which is amplified by a factor of 430 (the ratio between the other two resistors). For small input currents, the output voltage will approach 0V; for higher input currents, it will approach V_{CC}.

Note that while the battery charge sensor circuit detects charge completion, it does not perform any regulation function to protect the device battery. We must differentiate between battery protection and charge completion detection here- the former is left to the target device itself, not the modifications for our project. It is assumed that the target device will protect its own battery from over-current or over-voltage. If battery protection became a concern, it is possible for it to be performed by the battery charge sensor, but there was no effort to add that functionality here.

Validation:

The sensor block uses the typical application of the LT6106 current sensing IC; details available in the datasheet.

Battery Sensor Controller

Each modified target device contains an Atmel Tiny261 microcontroller. This uC, as shown in Figure 2, provides a data connection back to the Mega48 charging controller on the base.

Pin A7 is the analog input from the battery current sensor; it is configured as an Analog-to-Digital Converter (ADC) in the software.

Pins B0 (MOSI), B1 (MISO), B2 (SCK) and B7 (reset_n) are used for the AVR low-voltage serial programming interface.

The USI-SPI connection with the charging controller utilizes pins AI (DO) and B5 (SCK). Since the Tiny261 is configured as the slave, it reads the clock as an input and the MISO pin is, of course, an output.

Battery Sensor Controller Code

The code written for the battery sensor controller is available in Appendix A. The 10-bit ADC input is configured to run in single-shot mode at 512kHz, with V_{CC} as a reference. The result is left-adjusted; reading only the high byte effectively makes it an 8-bit ADC.

After initialization, the microcontroller enters an infinite, empty loop. Twice per second it executes an interrupt service routine (ISR) that reads the ADC and computes a running average of the current sensor readings.

In the running average, the new ADC reading is weighted at 1/16 and the previous average is weighted at 15/16. Therefore only a long series of low readings will drop the running average. If the running average is below a defined threshold, then a “battery full” flag is set.

The ISR then creates a data packet containing the device ID (which will be unique to the target device) the lowest 7 bits and the “battery full” flag in the MSB. It sends this out the USI port, to be received by the charging controller.

Chassis

The chassis is an 8” x 11” x ¾” (W x L x D) black plastic case. On one side of the chassis the power switch is attached flush to the case, and the three device sensor pushbuttons are exposed for use. On the opposite side, the USI-SPI connector, a 3x1 female header, is exposed to be connected to the target devices. A thin power cord runs out of the chassis to the AC-DC wall transformer. The top surface of the chassis is flat and blank; this is the surface on which the user places a target device that is intended to be charged.

4 Results

A. Measuring Efficiency

This efficiency is calculated as the ratio of the power absorbed by the inductive pickup block on the target device to the power consumed by the power transmission block on the charger base:

$$efficiency = \left| \frac{P(inductive\ pickup)}{P(transmission\ coils)} \right|$$

Power consumed by the transmission coils is measured by hardwiring the charge_enable control signal to the 5V power bus (thus isolating the transmission coils block) and measuring the average current flow from the 5V power bus. The product of the average current with the 5V bus is the power consumption, when transmitting.

$$\begin{aligned} &I_{ave}(transmission\ coils) = 49.2mA \\ &V(transmission\ coils) = 5.00V \\ &\Rightarrow P(transmission\ coils) = 246mW \end{aligned}$$

Power absorbed by the inductive pickup is calculated by measuring the output current and output voltage to the device power regulator block. It is necessary that the block still be attached to load the output.

$$\begin{aligned} &I(inductive\ pickup) = 6.03mA \\ &V(inductive\ pickup) = 4.33V \\ &\Rightarrow P(inductive\ pickup) = 26.1mW \end{aligned}$$

With each individual power calculation, efficiency follows:

$$efficiency = \left| \frac{P(inductive\ pickup)}{P(transmission\ coils)} \right| = \frac{26.1mW}{246mW} = 10.6\%$$

B. Testing Results

Complete testing procedures are listed in Appendix B. In summary:

Table 3. System Requirements and Test Results

Requirement	**Pass/Fail**
Wireless	Pass
Charging multiple devices simultaneously	Pass
Safe for user	Pass
Safe for surrounding devices	Pass
Robust	Pass
Versatile	Pass
Efficiency	Fail
Low cost	Pass
Monitoring charge status	Pass
Monitoring device ID	Pass
Portable (enhancement)	Pass
Usable (enhancement)	Pass

The completed system only transmits power with approximately 10% efficiency, failing to reach the required 50%. Otherwise all system tests passed; the monitoring requirements were not met for the Winter term project deadline, but were completed during Spring term; the enhancement requirements (Portable, Usable, Aesthetic) were added and successfully achieved during Spring term.

C. Design Compromises

Two major, disappointing design compromises had to be made during the course of the project. Early in the year, the design process called for passive RFID tags in each target device, to be used for the monitoring capabilities. The advantage of the passive RFID tags is that (in addition to communicating wirelessly), the tag does not rely on power from the device to operate correctly; even with a fully discharged device battery the tag could complete the necessary handshake to send data to the charger base.

However, the ECE44x teaching staff discouraged our team from pursuing the RFID solution, explaining that it would be too difficult to implement in addition to the rest of the design work for the wireless power transfer. In response, we changed the design to instead use infrared emitters and sensors to pulse out serial data from the battery sensor controller to the charging controller. This was less desirable because the reliability of the sensors was poor, and would require the target device to be aligned properly on the charger base such that the sensor and emitter would align.

Again, the ECE44x teaching staff discouraged this design, because there would be too much interference from the environment for the sensors to function reliably. With few feasible options and the design process coming to an end, we resorted to an undesirable solution: a wired USI-SPI connection between the charging base and target device.

So the first unfavorable design compromise is that the wireless power transfer system has a wired data connection. In defense of this compromise, the wireless power transfer can function without the data connection; a device may be charged without transmitting the device ID and status back to the base, and the software supports this. Furthermore, the purpose of the project was to be a proof-of-concept for a wireless charging system, and that was a success despite the deficiencies in the data communication.

The second compromise stems from the lack of RFID tags as well. The device sensing block must be able to sense a target whether the device battery is fully charged, partially charged, or depleted. When the battery is depleted it is unable to power any device-side circuitry to indicate its presence to the charger. Possible solutions included a pressure sensor or reflective IR sensor on the surface of the charging base; unfortunately neither of these could discern between a target device and a brick. Then the charger would mistake a brick for a device with a dead battery, and waste energy trying to recharge its battery.

Therefore, the device sensor is instead just a pushbutton. When the user puts a device on the charger that has a fully depleted battery (unable to start charging by sending its device ID via the USI-SPI port), the user presses the pushbutton. This forces the charger to send power for 5 minutes without requiring a valid device ID. After 5 minutes it is assumed that the device battery will be full enough to start sending USI-SPI packets.

Though this pushbutton design works, it is undesirable because it requires additional input from the user, it isn’t automated.

5 Conclusion

An inductive charging system was successfully designed and implemented, capable of charging as many as three unique target devices simultaneously. The basic functionality of the project is complete, and it implements a communication system allowing the charging base to monitor the level of charge of devices.

Issues with the design are primarily in the efficiency of the power transfer, which at 10.6% is significantly lower than the target 50%. Also, though it meets the customer and engineering requirements, the wired data communication is a disappointment and trivializes the project, because more efficient power transfer could be achieved with fewer wires than are required for the USI-SPI connection. Finally, the user display is awkwardly positioned outside of the chassis.

The most needed improvements are to the inductive coils (to improve the efficiency) and implementing RFID (for wireless data transfer, and automatic device sensing). See section 3B for a more thorough discussion of how the efficiency of the inductive coils could be optimized in a future design. Ideally the driving frequency would be increased; a research group at MIT demonstrated that many important parameters for the wireless power transfer efficiency are optimized at 13.6 MHz (compared to the 606kHz that we used). Second, adding passive RFID tags to each target device would allow for automated device sensing and recognition, even if the device battery is depleted. Surely there is an off-the-shelf RFID inventory system that would drastically improve the marketability of this project.

BIBLIOGRAPHY

[1] A. Kurs, A. Karalis, R. Moffatt, J.D. Joannopoulos, P. Fisher and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, pp. 83-86, 6 July 2007.

[2] "WildCharge: The charger of the smart brigade," June 2008. [Online]. Available: http://www.firebox.com/product/2128/WildCharge#reviews_h. [Accessed: 18 October 2009].

[3] L. Ulanoff, "WildCharger Launches Wireless Power Pad," *PC Magazine*, 25 January 2007.

[4] H. Malik, "Splashpower Charger Lets You Charge Wirelessly, Lose That Wired Mess," 8 January 2008. [Online]. Available: http://gizmodo.com/342384/splashpower-charger-lets-you-charge-wirelessly-lose-that-wired-mess. [Accessed: 18 October 2009]

[5] "Splashpower powers mobile phones without chargers," 21 September 2006. [Online]. Available: http://www.esato.com/archive/t.php/t-131342,1.html. [Accessed: 18 October 2009].

[6] S. Gingichashvili, "eCoupled's wireless power," 29 January 2008. [Online]. Available: http://thefutureofthings.com/news/1099/ecoupleds-wireless-power.html. [Accessed: 18 October 2009]

[7] I. Gennuth, "Wireless Power Demonstrated," 3 May 2007. [Online]. Available: http://thefutureofthings.com/pod/250/wireless-power-demonstrated.html. [Accessed: 18 October 2009].

[8] Powermat, "Powermat Product Line," [Online]. Available: http://www.powermat.com/us/products. [Accessed: 17 October 2009].

[9] Sekitani, T.; Takamiya, M.; Noguchi, Y.; Nakano, S.; Kato, Y.; Hizu, K.; Kawaguchi, H.; Sakurai, T.; Someya, T., "A large-area flexible wireless power transmission sheet using printed plastic MEMS switches and organic field-effect transistors," Electron Devices Meeting, 2006. IEDM '06. International , vol., no., pp.1-4, 11-13 Dec. 2006. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4154183&isnumber=4139311. [Accessed: 17 October 2009].

[10] Duracell, "Products myGrid," [Online]. Available: http://www.duracell.com/us/mygrid/default.asp. [Accessed: 17 October 2009].

[11] Duracell, "Duracell myGrid Operation Manual," [Online]. Available: http://www.duracell.com/us/mygrid/myGrid-instructions.pdf. [Accessed: 17 October 2009].

[12] C. Herold, "Tekno Creations' Wii InCharge Dual Charge Station - Accessory Review", [Online]. Available: http://nintendo.about.com/od/accessories/fr/inchargerevu.htm. [Accessed: 19 October 2009].

[13] "Wii InCharge Dual Charge Station", 2 September 2008, [Online]. Available: http://www.amazon.com/Wii-InCharge-Dual-Charge-Station-Nintendo/dp/B0017KCUEQ. [Accessed: 19 October 2009].

[14] "Touchstone Charging Kit", 7 June 2009, [Online]. Available: http://store.palm.com/product/index.jsp?productId=3671707. [Accessed: 19 October 2009].

[15] "Palm Touchstone Kit for Palm Pre", [Online]. Available: http://www.amazon.com/dp/B002CMEIWU?tag=bestbuycellphone-20&camp=213381&creative=390973&linkCode=as4&creativeASIN=B002CMEIWU&adid=1Z4BF7DAXW8W2N2BNM7J&. [Accessed: 19 October 2009].

[16] Atmel, “ATmega48/V Datasheet”, [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf. [Accessed: 17 May 2010].

[17] Atmel, “ATtiny261/V Datasheet”, [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2588.pdf. [Accessed: 17 May 2010].

[18] Lite-On Electronics Inc, "LTD-4608JR Datasheet". [Online]. Available: http://media.digikey.com/pdf/Data%20Sheets/Lite-On%20PDFs/LTD-4608JR.pdf. [Accessed: 29 November 2009].

[19] OSRAM Opto Semiconductors, "LATB-G66B Datasheet", 5 October 2007. [Online]. Available: http://media.digikey.com/pdf/Data%20Sheets/Osram%20PDFs/LATB_G66B.pdf. [Accessed: 29 November 2009].

[20] Linear Technologies, "LT6106 Datasheet", 2007. [Online]. Available: http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1154,C1009,C1077,P37671,D24992. [Accessed: 30 November 2009].

[21] Texas Instrument, "tps62200 Datasheet", September 2008. [Online]. Available: http://focus.ti.com/lit/ds/symlink/tps62200.pdf. [Accessed: 20 February 2010].

Appendix A. Microcontroller Code

1. Charging Controller

```
/* mega48_code.c
   Allen Waters
   5.13.10
       For ECE 44x Project 12.
       Mega48 watches pushbutton input.  When the button is pushed, the
       controller sends power to the transmission coils for 5 minutes.
       During this time the user display turns on the green status LED,
       and displays the device ID.  If no valid ID is available, then
       it displays "On" instead of the device ID.

       After 5 minutes the controller continues sending power as long as
       valid device IDs are read from SPI.  If an invalid ID is read, it
       turns off power, and goes back to the red status LED with the
       display blank.
*/

/*****************************************************************************/
#define F_CPU 1000000

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

/*****************************************************************************/
//  Hardware setup:
/*****************************************************************************/
/* -- Inputs --
       Pin  B0    : input from device sensor (button)
       Pins B2-5  : SPI connection from Tiny26
                    B2: slave select (output, not used)
                    B3: MOSI (i.e. output, not used)
                    B4: MISO (i.e. input)
                    B5: SCK (output, at 1/4 speed)
   -- Outputs --
       Pin  C0    : enable power transmission to coils
       Pins C1-C2 : outputs to segment select
       Pin  C5    : output to status LED
       Pins D0-D7 : outputs to 7-segment LED control signals
*/

/*****************************************************************************/
// USI->SPI PACKETS
/*****************************************************************************/
/* (MSB)  b7  b6  b5  b4  b3  b2  b1  b0  (LSB)
          |   <------- device ID ------>
          |
      charge status
  (1->done, 0->charging)
*/

/*****************************************************************************/
// GLOBAL VARIABLES
/*****************************************************************************/
// holds IDs to display. Volatile so it won't be optimized out of the ISR.
volatile uint8_t seg_data[2];

// holds the ID number (two digits in decimal) to display
volatile uint8_t id_num = 0;

// hold remaining charge time in seconds
volatile uint16_t c_time = 0;

// 1 -> send power; 0 -> don't
volatile uint8_t send_power = 0;
```

```
// decimal to 7-segment LED display encodings, logic "0" turns on segment
uint8_t dec_to_7seg[11] = { 0b00100010,   //digit 0
                            0b11101011,   //digit 1
                            0b00110001,   //digit 2
                            0b01100001,   //digit 3
                            0b11101000,   //digit 4
                            0b01100100,   //digit 5
                            0b00100100,   //digit 6
                            0b11100011,   //digit 7
                            0b00100000,   //digit 8
                            0b11100000,   //digit 9
                            0b11111111 }; //blank

// holds state of debouncing for pushbutton input
uint8_t sw_state = 0;

//***************************************************************************
/* Function: chk_button
    Checks the state of the pushbutton.  The function passes in ones until the
    button is pushed, then passes in zeros while the button is pushed.  Returns
    a 1 only once per debounced button push so a debounce and a toggle function
    can be implemented at the same time.

    Source: Ganssel's "Guide to Debouncing".  Expects active low pushbutton on
    PIN B0.  Debounce time is external loop delay times 7.
*/
uint8_t chk_button(void){
    sw_state = (sw_state << 1) | (!bit_is_clear(PINB, 0));
    if (sw_state == 0x80) return 1;
    return 0;
}

//***************************************************************************
/* Function: validate_id
    Determine whether ID number maps to a known device.
*/
uint8_t validate_id(uint8_t num){
    if(num == 72) return 1;
    if(num == 35) return 1;
    if(num == 81) return 1;
    return 0;
}

//***************************************************************************
/* Function: seg_values
    Determine decimal digits to display for ID number.
*/
void seg_values(uint8_t sum){
    seg_data[1] = dec_to_7seg[ sum % 10 ];
    seg_data[0] = dec_to_7seg[ (sum - (sum%10)) / 10];
}

//***************************************************************************
/* Function: spi_init
    Initializes SPI communication.  Master mode, MSB first, sampling the
    input from the Tiny26 on the rising edge of SCLK, clock low when idle.
    Running at system clock speed / 4 = 256 kHz.
*/
void spi_init(void){
    // Pins B5:2 direction already configured in main()

    //leaving SPI2X = SPR1 = SPR0 = 0 makes SCK frequency 1/4 the oscillator
    SPCR |= (1<<SPE)|(1<<MSTR);
}

//***************************************************************************
/* Function: tcnt0_init
    Initializes timer/counter 1.  TCNT0 is in normal mode, running with
    a 1/64 prescale on the I/O clock.

    TCCR0A is initialized to correct values; and TCNT0, OCR0A, and OCR0B
```

```
    do not need to be initialized.
*/
void tcnt0_init(void){
    TCCR0B |= (1<<CS01)|(1<<CS00);
    TIMSK0 |= (1<<TOIE0);
}

//******************************************************************************
/* Function: timer/counter0 ISR
    When the TCNT overflow interrupt occurs, update device status values.

     1MHz/64         = 16.384 kHz
     1/16kHz         = 61.04 us
    (1/16kHz)*256    = 15.625 ms
    (1/16kHz)*256*64 = 1 s
*/
ISR(TIMER0_OVF_vect){
    static uint8_t id_num = 0, finished = 0;
    static uint16_t ms15_count = 0;

    //if button was pushed: add 5 min to timer
    if(chk_button()) c_time = 300;

    //increment counter of 15-millisecond intervals
    ms15_count++;

    //if 64 intervals have passed (1 second), update counter
    if(!(ms15_count % 64)) {
        //decrement charge time counter
        if(c_time > 0) c_time--;

        //write garbage to SPI, then read input byte
        SPDR = id_num;
        id_num = SPDR;
    }

    //if packet has more than just 7-bit ID num, it must be
    //    because MSB is on to indicate charging is complete.
    if(id_num > 99) finished = 1;

    //decide what to display
    if(validate_id(id_num) && (finished == 0)){
        //update 7-segs with new values
        seg_values(id_num);
        send_power = 1;
    } else if(validate_id(id_num) && (finished == 1)){
        //update 7-segs with "--" to indicate completion
        seg_data[0] = 0b11111101; //"-"
        seg_data[1] = 0b11111101; ///"-"
        send_power = 0;
    } else if(c_time > 0){
        //update 7-segs with "On"
        seg_data[0] = 0b00100010; //"O"
        seg_data[1] = 0b10101101; //"n"
        send_power = 1;
    } else {
        seg_data[0] = dec_to_7seg[10]; //blank
        seg_data[1] = dec_to_7seg[10]; //blank
        send_power = 0;
    }
}

//******************************************************************************
int main()
{
    //make PB0, PB4 inputs with pull-up resistor
    //make PB2, 3, 5 output
    DDRB  = 0x2C;
    PORTB = 0xFF;

    DDRC = 0xFF;    //bits C0-C7 outputs to user_display board
```

```
    PORTC = 0x06;   //  initially turn 7-segs off
    DDRD = 0xFF;    //outputs, for LED segments
    PORTD = 0xFF;   //  initially output high (blank)

    tcnt0_init();   //prepare timer/counter 0
    spi_init();     //prepare SPI port
    sei();          //enable interrupts before entering loop

    while(1){
        PORTC |= 0x06;
        PORTD = seg_data[0]; //write data out
        PORTC &= 0xFB;       //select second 7-seg
        _delay_us(200);

        PORTC |= 0x06;
        PORTD = seg_data[1]; //write data out
        PORTC &= 0xFD;       //select first 7-seg
        _delay_us(200);

        PORTC |= 0x06;

        //if sending power, update PORTC.  Otherwise shut off.
        if(send_power){
            PORTC |= (1<<5)|(1<<0);   //send power; green status LED
        } else {
            PORTC &= 0xDE;            //no power; red status LED
        }
    }
}
```

2. Battery Sensor Controller

```
/* tiny26_code.c
   Allen Waters
   5.12.10
       For ECE 44x Project 12.
       The Tiny26 has an input from the current sensor on the device
       battery.  It reads the value from this sensor using the ADC,
       and decides whether the device is done charging or not.  It
       continuously sends data to the charging controller (Mega48)
       containing the unique device ID and the charging status.
*/

/******************************************************************************/
#define F_CPU 1000000

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

/******************************************************************************/
// HARDWARE CONNECTIONS
/******************************************************************************/
/* --Inputs--
      Pin A7 : input from device sensor (using ADC)
               NOTE: PA7 is called ADC6, index is weird.
   --Outputs--
      Pin A1 : DO (data out from USI interface)
      Pin A2 : SCL (clock from USI interface)
*/

/******************************************************************************/
// USI->SPI PACKETS
/******************************************************************************/
/* (MSB)  b7  b6  b5  b4  b3  b2  b1  b0  (LSB)
          |   <------- device ID ------>
          |
      charge status
  (1->done, 0->charging)
```

```
*/

/*****************************************************************************/
// GLOBAL VARIABLES
/*****************************************************************************/
#define _DEVICE_ID_ 81

//holds weighted average of current sensor values.  Initialize to 64, which
//    is about 1V with a 5V source.
volatile uint16_t weighted = 0x40;

/*****************************************************************************/
/* Function: find_ave
    Calculates weighted average of current sensor readings.  Previous average
    is weighted at 15/16, new value weighted at 1/16.
*/
void find_ave(uint8_t new_val){
    weighted *= 15;
    weighted >>= 4;
    weighted += new_val;
}

/*****************************************************************************/
/* Function: tcnt0_init
    Initalizes 8-bit timer/counter0 (TCNT0). TCNT0 is running in normal mode,
    using internal I/O clock with 1/64 prescaling.  Interrupts occur at
    overflow, 0xFF.

    TCCR0A, TCNT0x, OCR0x, and TIFR need no initialization.
*/
void tcnt0_init(void){
    TCCR0B |= (1<<CS01)|(1<<CS00);
    TIMSK  |= (1<<TOIE0);
}

/*****************************************************************************/
/* Function: adc_init
    Initalizes 10-bit analog-to-digital converter to read the current sensor.
    ADC6 (reading pin A7) is set up for:
        -single-shot mode
        -Vcc as reference
        -left-adjusted
        -system clock is prescaled by 2
        -unipolar mode
        -no gain
        -digital input buffer disabled

    ADCH, ADCL, ADCSRB, and DIDR1 need no initialization.
*/
void adc_init(void){
    ADMUX |= (1<<ADLAR)|(1<<MUX2)|(1<<MUX1);
    DIDR0 |= (1<<ADC6D);
    ADCSRA |= (1<<ADEN);
}

/*****************************************************************************/
/* Function: usi_init
    Initalizes the USI port on the tiny26. Three-wire mode, using SCLK from
    Mega48, which is running at 256kHz.  Slave mode, using PORTA2:0 rather
    than PORTB2:0.

    USIDR, USIBR, USISR don't need to be initialized.
*/
void usi_init(void){
    DDRA |=   0x02;     //Turn DO to output

    USICR |= (1<<USIWM0)|(1<<USICS1);
    USIPP |= (1<<USIPOS);
}

/*****************************************************************************/
```

```
/* Function: timer/counter0 ISR
    When TCNT0 overflow occurs:
        -Increment counter of 15 ms increments
        -Every half second, fill data packet and send to Mega48

     1MHz/64         = 16.384 kHz
    (1/16kHz)        = 61.04 us
    (1/16kHz)*256    = 15.625 ms
    (1/16kHz)*256*32 = 0.5 s
*/
ISR(TIMER0_OVF_vect){
    static uint16_t ms15_count = 0;
    uint8_t adc_input, byte_out = _DEVICE_ID_;

    //increment the number of 15ms intervals
    ms15_count++;

    //if 32 intervals have passed (1/2 second), send a USI packet
    if(!(ms15_count % 32)) {
        //read the ADC data
        ADCSRA |= (1<<ADSC);

        //wait for ADC (single-shot) to finish conversion before reading
        while((ADCSRA & (1<<ADSC)) != 0) {}
        //read the value into a variable
        adc_input = ADCH;

        //find new running average
        find_ave(adc_input);

        //if running average is less than 5 (very small current), then
        //    device is done charging.  Flip MSB in data byte to be a 1
        //    to tell base to cut off power.
        if(weighted < 5){
            byte_out |= 0x80;
        }

        //send data packet
        USIDR = byte_out;
    }
}

/****************************************************************************/
int main()
{
    //make A7 an input without pullup resistor
    DDRA = 0x00;
    PORTA = 0x7F;

    tcnt0_init();       //initialize counter timer zero
    adc_init();         //initialize the ADC
    usi_init();         //initialize USI (~SPI) interface

    sei();              //enable interrupts before entering loop

    //spin indefinitely... everything handled in ISR
    while(1) { }
}
```

Appendix B. Testing Procedures

1. Test Name: Wireless

Minimum Requirement Tested: Wireless

Test Description:
1. Completely drain device battery.
2. Measure initial battery voltage.
3. Place device on the charging station.
4. Allow to charge for 1 hour.
5. Remove device battery.
6. Measure battery voltage using voltmeter.

PASS: Final measured battery voltage is greater than initial battery voltage.
FAIL: Charging battery for 1 hour does not increase battery voltage.

2. Test Name: Charging Multiple Devices Simultaneously

Minimum Requirement Tested: Charging Multiple Devices Simultaneously

Test Description:
1. Drain batteries of three target devices.
2. Measure initial battery voltage of each device.
3. Place all three devices on charging system.
4. Allow devices to charge for 1 hour.
5. Remove device batteries from each device.
6. Measure final battery voltage of each device.

PASS: All three device batteries show an increase in voltage.
FAIL: Any one of the three device battery voltages does not increase.

3. Test Name: Leakage power to user

Minimum Requirement Tested: Safe

Test Description:
1. Turn on charging station.
2. Connect ammeter from surface of charging station to ground.
3. Read leakage current from charging surface.

PASS: Measured current is below 1mA.
FAIL: Measured current is greater than 1mA.

4. Test Name: Safe for surrounding devices

Minimum Requirement Tested: Safe

Test Description:
1. Turn on charging system.
2. Store test data file on USB storage device.
3. Place USB storage device on charger.
4. Place target device on station that will receive power.

4. Wait 1 hour.
5. Remove USB storage device.
6. Compare test data file contents to original test data file.
PASS: Test data file is undamaged.
FAIL: Test data file is changed.

5. Test Name: Robust

Minimum Requirement Tested: Robust

Test Description:
1. Turn on charging system.
2. Place target device on the charger.
3. Wait 20 hours while switching target devices every 4 hours.
4. Place new device on charging system.
5. Verify that, after 20 hours, charger will still transfer power to new device.
PASS: Device in Step 4 receives power, and structure (receiver and transmitter) is intact.
FAIL: Station does not charge new device after 20 hours, or structure (receiver or transmitter) is damaged.

6. Test Name: Versatile

Minimum Requirement Tested: Versatile

Test Description:
1. Repeat "Wireless" system test for total of 3 different target devices.
PASS: Each device passes "Wireless" system test.
FAIL: Any one of three devices fails "Wireless" system test.

7. Test Name: Efficiency

Minimum Requirements Tested: Efficiency

Test Description:
1. Acquire two identical target devices, and one wired charging adapter.
2. Turn on charging system.
3. Charge target device on wireless system until device reports full battery. Record time needed to charge battery.
4. Charge target device with wired adapter until device reports full battery. Record time needed to charge battery.
5. Compare times from Steps 3 and 4.
PASS: Wireless charging time is no more than 2x the wired charging time.
FAIL: Otherwise.

8. Test Name: Low cost

Minimum Requirement Tested: Low cost

Test Description:

1. Calculate total cost from Bill of Materials for charging system base.
PASS: Total cost does not exceed $150 per unit.
FAIL: Total cost greater than $150 per unit.

9. Test Name: Monitoring capabilities for charging status
Minimum Requirement Tested: Monitoring capabilities

Test Description:
1. Turn charging system on.
2. Partially drain device battery.
2. Place target device on charger.
3. Read charging status from user display.
4. Wait for charging status to change to fully charged.
PASS: Status LED is initially red (charging enabled) and later changes to green (charging complete).
FAIL: 1. Display LED is off.
2. Display LED initially green.
3. Display LED never changes state.

10. Test Name: Monitoring capabilities for device ID
Minimum Requirement Tested: Monitoring capabilities

Test Description:
1. Turn charging system on.
2. Place target device on charger.
3. Read device ID from user display.
PASS: Device ID on user display matches assigned ID number programmed onto the device.
FAIL: Device ID mismatch.

Minimum Requirement Tested: Portable

11. Test Name: Portable
Minimum Requirement Tested: Portable

Test Description:
1. Examine the package and circuitry of each device used with the charging system.
PASS: A PCB is used with each device, and this modification does not increase the size of the device by more than one inch in any dimension.
FAIL: Any device does not use a PCB, or the device geometry is increased by more than one inch in any dimension.

12. Test Name: Usable

Minimum Requirement Tested: Usable

Test Description:

1. Locate power switch on the charging base chassis.
2. Turn switch to OFF position and observe charging status of a handheld device placed on the charging base.
3. Turn switch to ON position and observe charging status of a handheld device placed on the charging base.

PASS: Charging base chassis has a power switch. When in OFF position, devices placed on charging base do not receive power from base. When in ON position, devices placed on charging base may receive power.

FAIL: Otherwise.

Appendix C. Parts List and Budget

The following budget assumes a single charging base and a single target device.

Table C1. Power Supply Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
U101-102	Digikey	LM7805CT-ND	2	0.60	1.20
C101-102	Digikey	445-2854-ND	2	0.262	0.524
C103	Digikey	BC2361-ND	1	0.248	0.248
T101	Radioshack	273-356	1	30.79	30.79
				Total	32.77

Table C2. Device Sensor Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
PB201-PB203	Digikey	EG2554-ND	3	2.18	6.54
				Total	6.54

Table C3. User Display Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
L601	Digikey	160-1538-5-ND	3	1.20	3.60
L602	Digikey	475-2813-1-ND	3	1.57	4.71
Q601-602	Digikey	2N4403-ND	6	0.13	0.78
R601-602	Digikey	CF1/43.3KJRCT-ND	6	0.08	0.48
R603-610	Digikey	CF1/8240JRCT-ND	24	0.09	2.16
R611	Digikey	CF1/482JRCT-ND	3	0.08	0.24
R612	Digikey	CF1/4160JRCT-ND	3	0.08	0.24
D601	Digikey	1N4001FSCT-ND	1	0.27	0.27
				Total	12.48

Table C4. Charging Controller Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
U301	Digikey	ATMEGA48P-20PU-ND	3	2.58	7.74
R301-R304	Digikey	P470BACT-ND	12	0.08	0.96
J301	Mouser	3M - 30310-5002HB	3	0.34	1.02
				Total	9.72

Table C5. Power Transmission Coils Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
L501	-	-	3	2.00	6.00
C501	Digikey	490-5401-ND	3	0.11	0.33
M501	Digikey	497-6730-5-ND	3	1.43	4.29
R501	Digikey	100H-ND	3	0.29	0.87
R502	Digikey	100KH-ND	3	0.29	0.87
X501	Digikey	XC244-ND	3	2.13	6.39
				Total	18.75

Table C6. Inductive Pickup Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
L901	Amidon	FT-140-61	1	3.75	3.75
C901	Digikey	490-3363-1-ND	1	0.23	0.23
C902	Digikey	495-1537-1-ND	1	1.85	1.85
D901-4	Digikey	1N5819HW-FDICT-ND	4	0.302	1.21
				Total	7.04

Table C7. Device Power Regulator Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
C1001	Digikey	587-1782-1-ND	1	0.33	0.33
U1001	Digikey	296-12716-1-ND	1	2.49	2.49
L1001	Digikey	490-4029-1-ND	1	0.151	0.151
R1001	Digikey	P620KCCT-ND	1	0.091	0.091
R1002	Digikey	P100KDACT-ND	1	0.204	0.204
C1002	Digikey	490-1592-1-ND	1	0.229	0.229
C1003	Digikey	490-1592-1-ND	1	0.229	0.229
C1004	Digikey	587-1353-1-ND	1	0.253	0.253
				Total	4.01

Table C8. Battery Charge Sensor Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
U1101	Digikey	LT6106CS5#TRMPBFCT-ND	1	1.96	1.96
R1101	Digikey	615HR020-ND	1	0.46	0.46
R1102	Digikey	CF1/410JRCT-ND	1	0.08	0.08
R1103	Digikey	4.3KW-1-ND	1	0.16	0.16
				Total	2.66

Table C9. Battery Sensor Controller Block Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
U1401	Digikey	ATTINY261-20PU-ND	1	2.13	2.13
R1401	Digikey	P470BACT-ND	4	0.08	0.32
J1401	Mouser	3M - 30310-5002HB	1	0.34	0.34
				Total	2.79

Table C10. Chassis Budget

Ref. Des.	Vendor	Vendor #	Quantity	Price (onesies)	Extended Price
	Digikey	HM169-ND	1	25.68	25.68
				Total	25.68

Table C11. Budget Summary

Functional Block	Block Cost
Power Supply	32.77
Device Sensor	6.54
User Display	12.48
Charging Controller	9.72
Power Transmission Coils	18.75
Inductive Pickup	7.04
Device Power Regulator	4.01
Battery Charge Sensor	2.66
Battery Sensor Controller	2.79
Chassis	25.68
Total	$122.34