Abstract — This document outlines the design of PowerBot, a four-wheeled robot that serves as a mobile charging station for mobile electronic devices. Mobile devices are charged via a USB cable or an inductive charging mat using PowerBot’s internal rechargeable battery. The battery can be recharged through a turret-mounted solar panel that is configured to track the sun for optimal charging efficiency. PowerBot’s movement is controlled remotely by a mobile phone application via Wi-Fi. Obstacle avoidance routines are implemented to assist the user in controlling PowerBot. Ultrasonic ranging sensors supply the obstacle avoidance routines with range data to appropriately respond to obstacles.

Index Terms — DC motors, stepper motor, ultrasonic ranging sensors, obstacle avoidance, solar panel tracking, charging, inductive charging, Wi-Fi, iOS application, microcontroller.

I. INTRODUCTION

The idea of PowerBot came about while attempting to find a novel solution to the matter of charging mobile electronic devices. Inspiration came from observing how small household robots (such as autonomous vacuums) have become more commonplace in society. A household robot that charges mobile devices seemed to be along the same vein. For the implementation to be novel, it was decided that the device would include an inductive wireless charging pad, a solar panel, and remote access and control via an iPhone application.

As a typical example of PowerBot’s utility, a user is lying comfortably on their couch after a long day. Getting up simply to charge their phone feels like a waste of energy. So the user opts to use what little power remains to boot up the PowerBot application to drive PowerBot to their location. PowerBot deftly avoids obstacles that the lazy user may not see while driving it. Upon arrival, the user can place their mobile device on the inductive charging mat or plug into one of the available charging ports, thus keeping their phone and/or device charged and nearby.

To achieve PowerBot’s functionality, two microcontrollers were selected to independently control the subsystems. One will control the motors and solar panel turret system. The other will control the ultrasonic ranging sensors, the obstacle avoidance routines, and Wi-Fi communication to the iPhone application.

The charging abilities of PowerBot will consist of using its onboard battery to charge devices through the use of an inductive charging pad located on the top of PowerBot. To use the inductive charging mat, the device would need to use an adapter to allow the device to receive power from the charging pad and to be placed on the mat. If desired, the user can charge using the USB port on PowerBot should an inductive charging sleeve not be available.

Another key feature to PowerBot is not only in how it distributes power, but how it gets its power. Fastened to the top of PowerBot will be a photovoltaic panel, allowing PowerBot to recharge its internal battery by redirecting the panel toward the direction that receives the most solar energy.

The iOS application is designed to report PowerBot’s current status and remotely control its movement. The iOS device communicates to one of PowerBot’s microcontrollers through Wi-Fi. As the robot moves, the microcontroller will aid movement through obstacle avoidance routines.

II. SYSTEM COMPONENTS

The following sections describe the components used to construct PowerBot. The component choices are discussed and their specifications presented.

A. Ultrasonic Ranging Sensors

PowerBot will utilize ultrasonic sensors to gather range information for the obstacle avoidance routines. Ultrasonic sensors can take readings fairly fast, at frequencies on the order of tens of hertz. This allows PowerBot to react to obstacles quickly while moving. An ultrasonic sensor takes readings by sending a train of high frequency (ultrasonic) pulses in a direction, then waiting for an echo of the pulses, which returns from an object. When an echo is detected, the duration between the pulse and the echo is used to calculate the distance to the object.
Figure 1 illustrates the general sonar sensor operation that will be active in our project:

The sensors chosen for PowerBot are MaxBotix LV-MaxSonar®-EZ0™. Four EZ0 sensors are mounted on the chassis of PowerBot. Two are in the front, facing outward from the corners at 30° from the front; and the other two are placed on the back corners in a similar fashion.

The measurement range of the LV-MaxSonar-EZ0 is from 6 inches to 254 inches. Obstacles can be detected as close as zero inches. Any objects that lie within 0 to 6 inches will read as 6 inches. A power supply of 2.2 V to 5.5 V is required to operate each sensor. These voltages correspond to current draws from 2 mA to 3 mA. A higher supply voltage results in a stronger beam with a slightly larger cone. The sensor is capable of taking readings at up to 20 Hz, but is usually closer to 50 Hz. There are two modes of operation: free-run and triggered. In free-run mode, the sensor operates continuously, taking readings at 20 Hz the entire time it is active. Triggered mode allows one to take measurements at any time, whenever the signal is triggered.

On this sensor, readings can be taken through several formats. The EZ0 supports serial, analog, and pulse width formats. The serial connection on the sensor operates from 0 V to 3.3 V (Vcc), at 9600 Baud. The analog connection on the sensor gives voltage readings that have a proportion of (Vcc/512) per inch. The pulse width output has the proportion 147 μs/inch. PowerBot will use the serial format to gather its data. It is interfaced with the PIC32 microcontroller through a UART connection.

B. Wi-Fi

Wi-Fi was chosen as the optimal communication solution for PowerBot. The Wi-Fi module inside of PowerBot will consist of one module that will be purchased from the Microchip Technology, Inc., containing all of the necessary components to make connections to networks and open connections to the user’s mobile device for communication of commands. The part being used for Wi-Fi communication is the Microchip MRF24WB0MA. The specifications for the Wi-Fi module are given in Table 1 below:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage Range</td>
<td>-0.3 - 5.5 V</td>
</tr>
<tr>
<td>RSSI</td>
<td>Yes</td>
</tr>
<tr>
<td>Sleep</td>
<td>Yes</td>
</tr>
<tr>
<td>Encryption</td>
<td>AES128</td>
</tr>
<tr>
<td>Pin Count</td>
<td>36</td>
</tr>
<tr>
<td>Frequency Range</td>
<td>2.405-2.48 GHz</td>
</tr>
</tbody>
</table>

Table 1: Wi-Fi Module Specifications

The Wi-Fi module is controlled by the microcontroller via a UART connection and allows for network speeds of up to 1 Mbps when transmitting data. It also allows the user to communicate with PowerBot from a large distance away, during testing, when connected via Ad-Hoc network, the range is around 50 feet. This can also be expanded when using an existing Wi-Fi network. This expansion will allow for a 2 hop range that is equal to more than double the previous range and can be longer depending on the quality of the in home router being used. The block diagram in Figure 2 below illustrates the layout of the Wi-Fi module. It works with the Microchip’s TCP/IP stack to allow for a fully functional Wi-Fi-capable device to be created.

C. Sensor & Wi-Fi Controller: Microchip PIC32

The microcontroller chosen for operating the ultrasonic sensors and communicating via Wi-Fi is the PIC32 from Microchip Technology, Inc. The PIC32 was chosen because it is sufficiently powerful to perform the obstacle avoidance operations while communicating to the iOS application through Wi-Fi. Additionally, Microchip has a Wi-Fi Comm Demo Board using a PIC32 and the MRF24WB0MA Wi-Fi module in the same package. This packaged proved to be a convenient means for managing
the aspects of sensor and Wi-Fi control. Table 2 below outlines the specifications of the processor.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU Family</td>
<td>PIC32MX695F512H</td>
</tr>
<tr>
<td>Max Speed</td>
<td>80 MHz</td>
</tr>
<tr>
<td>Program Memory Size</td>
<td>512 kB</td>
</tr>
<tr>
<td>RAM</td>
<td>128 kB</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>2.3 V to 3.6 V</td>
</tr>
<tr>
<td>Pin Count</td>
<td>64</td>
</tr>
<tr>
<td>A/D Channels</td>
<td>16</td>
</tr>
<tr>
<td>UART Modules</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 2: PIC32 Specifications

Having six UART modules is convenient, since four are required for the four ultrasonic sensors. Four of the PIC32’s analog-to-digital (A/D) channels are used to command the MSP430 to trigger the motors to move PowerBot.

D. Motor & Solar Tracking Controller: TI MSP430

The Texas Instruments MSP430G2553 is an ultra-low power device which consists of three low power modes which are optimized for prolonged battery life for portable devices. The chip features a relatively powerful 16-bit reduced instruction set computing (RISC) central processing unit (CPU), 16-bit registers, and constant generators which contribute to a certain level of code efficiency. The chip also contains a digitally-controlled oscillator (DCO) which provides an ultra-fast wake-up from standby mode in less than one microsecond (1 μs). This series of microcontrollers consists of two 16-bit timers, three (3) configurable operational amplifiers (op-amps), twenty-four (24) touch-sense-enabled input/output (I/O) pins, an on-chip comparator for analog signal compare functions or slope analog-to-digital (A/D) conversion, one universal synchronous/asynchronous communication interface (USART), and one high performance 10-bit analog-to-digital converter (ADC). The processor has a low supply voltage range between 1.8 V to 3.6 V.

The MSP430G2553 microcontroller was used to control the two DC motors and one stepper motor being used in PowerBot. It was also able to handle the solar tracking system that was implemented through the use of photoresistors and a simple application of voltage division. The low power consumption properties of 230 μA at 1 MHz and 2.2 V made this processor the ideal candidate when it came to power conservation and programmability for PowerBot.

E. Solar Panel

In order to generate, store, and distribute the power for the robot, two major components are needed. The first component, a photovoltaic panel, is an energy source that captures the light. The second component is a battery that is used to store the power that has been captured which will then be regulated by the charge controller.

Although there are many options for solar panel technologies, it is quite clear that the main candidate for this project is the mono-crystalline panel. Monocrystalline photovoltaic electric solar energy panels have been around for many years. They are among the oldest, most efficient, and most dependable ways to produce electricity from the sun.

Polycrystalline silicon panels are not far behind in efficiency but they are not much cheaper than Monocrystalline panels. The small savings in initial cost does not cover the savings in energy cost through the lifetime use of the panel. A thin film, or amorphous silicon, module was ruled out for its inferior efficiency.

Table 3 below outlines the general specifications regarding the solar panel chosen for the project:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Panel Type</td>
<td>Monocrystalline</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Instapark</td>
</tr>
<tr>
<td>Power</td>
<td>5 W</td>
</tr>
<tr>
<td>Maximum Voltage Power</td>
<td>17.5</td>
</tr>
<tr>
<td>Maximum Current Power</td>
<td>0.57 A</td>
</tr>
<tr>
<td>Open Circuit Voltage</td>
<td>21.95 V</td>
</tr>
<tr>
<td>Cost</td>
<td>$39.95</td>
</tr>
</tbody>
</table>

Table 3: Mono-crystalline Panel Specifications

This panel was chosen because the power output exceeds the required to charge the battery. Since the optimal output from the panel is greater, the circuitry will operate primarily in buck mode, which is more efficient.

F. Battery

Many different types of batteries were researched for the system including lead-acid, nickel-cadmium, lithium-ion, and nickel-metal-hydride. As a result, Ni-Cd and Ni-MH would be eliminated because their specifications fall far behind the ones for Li-ion and Li-Po batteries. Consequently, our final decision will be made out the Li-ion and Li-Po batteries. By inspecting the internal resistance for both batteries we first noticed that the Li-Po has higher internal resistance, which means less output.
current. However, research shows that the capacity of the Li-Po is higher as well, which means that our conclusion, based on the internal resistance, was wrong. Also by inspecting the energy density, we definitely noticed that the Li-Po batteries were lighter. Moreover, Li-Po batteries are more efficient and last longer. As a result of this comparison, a Li-Po battery was selected to be the battery that powers PowerBot.

The specific battery chosen for the system is a Li-ion 18650 Battery from Batteryspace.com. This 14.8 V battery has a nominal capacity of 2.2 Ah which will provide ample power for effective operation of the system. More details of the battery are shown in Table 4 below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>• 14.8 V (working)</td>
</tr>
<tr>
<td></td>
<td>• 16.8 V (peak)</td>
</tr>
<tr>
<td></td>
<td>• 11 V (cut-off)</td>
</tr>
<tr>
<td>Protection</td>
<td>• PCB installed with the battery pack</td>
</tr>
<tr>
<td></td>
<td>• One 4.2 Amp polyswitch installed to limit maximum discharging current and to protect wrong polarity</td>
</tr>
<tr>
<td>Capacity</td>
<td>2.4 Ah (53 Wh)</td>
</tr>
<tr>
<td>Max. Discharging Rate</td>
<td>4.2 Amp limited by polyswitch.</td>
</tr>
<tr>
<td>Weight</td>
<td>411 grams (14.5 oz)</td>
</tr>
<tr>
<td>Dimensions (L x W x H)</td>
<td>5.2” (134 mm) x 72 mm (2.8”) x 32 mm (1.3”)</td>
</tr>
<tr>
<td>Cost</td>
<td>$102.95</td>
</tr>
</tbody>
</table>

Table 4: Li-Ion Battery Specifications

G. Solar/DC Battery Re-Charging

The DC voltage from the panel will vary depending on light intensity based on the time of day and solar panel temperature. On the battery side of the system, the battery voltage will vary depending on the load connected to it. In order to maintain optimal battery charging, it is extremely important that the panel voltage and current matches the required battery charging stage at that particular moment.

A DC-to-DC regulator will be implemented to increase or decrease the input panel voltage to the required battery level. There are many DC-to-DC converter topologies used today, such as Buck, Boost, Buck/Boost, and CUK. These regulators do not produce power. In fact these regulators consume some of the input power according to their efficiency. Therefore the adjusted voltage level affects the current level, ideally maintaining the same power level. Since current and voltage are both directly proportional to power, it is intuitive that in buck mode the voltage is lowered as the current increases. While in boost mode, the voltage is increased as the current decreases. As a matter of fact, a Boost converter will be introduced in order to step down the voltage coming out of the solar panel to step up the current.

H. Solar Panel Efficiency

In order to increase the efficiency (output power/input power) of our charging system, many solutions were researched.

For example:

- Increasing the size of the solar panel
- Introducing a single-axis tracking system
- Introducing a dual-axis tracking system

Increasing the size of the solar panel is not practical, since the project requirement suggested that the solar panel should be small. On the other hand, implementing a dual-axis solar tracking system will enhance the efficiency by 6%, which is not a great outcome when it comes to complexity of building it and maintain it.

So the mount will be constructed to adjust around one axis. It will be powered by a stepper motor. Single axis tracking will use light sensors and a control module that operates the electrical motors in accordance with the light variation sensed by these light intensity sensors. This solar tracking technique is the most efficient observed; therefore it is important to ensure that its benefits outweigh its energy consumptions.

I. Voltage Regulators

There are three types of voltage regulators that are being used in this project. They are the DE-SW033, DE-SW050, and DE-SWADJ. We used switching voltage regulators instead of linear voltage regulators due to efficiency. Switching voltage regulators operate by taking small amounts of power, almost piece by piece, from the input voltage and then redirect this power to the output. This process occurs through the use of an electrical switch and a simple controller which both regulate the flow of energy that is transmitted to the output. The loss of energy in moving these small pieces is much smaller than with what occurs in a linear voltage regulator. As a result, switching voltage regulators can achieve an 80-85% efficiency. Add that to the fact that this efficiency is less dependent on the input voltage and it can be seen that switching voltage regulators can power useful loads from high input voltage.
voltages. While the switching voltage regulator trumps the linear voltage regulator in efficiency, it does, however, fall short in design simplicity. Switching voltage regulators involve a complex circuit design which in turn makes them unpopular with hobbyists and even some enthusiasts. The switching voltage regulator also shines with respect to high input voltages and moving loads over 200 milliamps (mA). Using a linear voltage regulator for such a situation becomes impractical. The first regulator we employed is the DE-SW033, which is a 3.3 volt switching voltage regulator which has an input voltage range from 0 V to 30 V. This regulator provides continuous 1 amperes of output current and a peak of 1.25 amperes for approximately one minute. This switching regulator will be used to regulate power from the battery to 3.3 V for the MSP430G2553 microcontroller. The next switching regulator we used is the DE-SW050. This is a 5 volt regulator which has an input voltage range from 0 V to 30 V while still providing a continuous 1 amperes of output current and a peak of 1.25 amperes for approximately one minute. This regulator is being use to provide the correct amount of power for the USB charging port. Both of the aforementioned switching voltage regulators achieve a typical efficiency of 83% and up to 87%. Figure 3 below portrays the efficiency versus the input voltage of the DE-SW050 regulator when compared to a linear regulator:

![Figure 3: Efficiency vs. Input Voltage](image)

Figure 4 below portrays the efficiency versus the output current of the DE-SW050 regulator:

![Figure 4: Efficiency vs. Output Current](image)

The final regulator we used is the DE-SWADJ. This is an adjustable regulator which will be adjusted for 9 volts of output voltage. This regulator has an output voltage range of 1.25 V to 13 V with an output current of 1 ampere from 1.25 V to 10 V. Any output voltage higher than 10 volts reduces the regulator’s output current. This won’t be much of an issue as we don’t need anywhere near 1 ampere of current. This regulator is used for the inductive charging mat that PowerBot features for charging.

**J. Inductive Charging Mat & USB**

PowerBot employs a 9 volt inductive pad for device changing. This mat uses the Qi standard for wireless charging. For the system to work as designed, the user would make sure that his or her device had the appropriate Qi standard charging jacket already installed. From there, simply placing the device onto the inductive charging mat would initiate device charging of 9 volts at 0.5 amperes.

PowerBot also employs a 5 volt female USB A charging port that is used to charge a vast array of mobile devices. The inclusion of this USB port is important as USB has become very ubiquitous throughout the world and can be unofficially considered a standard for low-power/voltage charging. The user would simply take the USB cable supplied by their device manufacturer and plug it in to the female USB A port. From there, the device will initiate charging of 5 volts at 0.5 amperes.

**K. Motors**

The motors used to drive PowerBot and the solar tracker array are low-powered as PowerBot’s speed requirement is relatively low. The stock car’s DC motors were reused as they fit into special plastic housings of the chassis. A stepper motor was purchased for the solar tracking turret
which will be connected to a belt system which will turn the affixed solar panel.

The DC motors are small and do not require much operating power. There are quite a few advantages to using these motors. Some advantages are that they have been around for quite some time and are relatively inexpensive. This means that should any issues arise, they are readily available in many shapes and sizes with a wide range of voltage, current, and torque levels. Also, the speed and torque in motors with permanent magnets are linear to the input voltages applied. That means that a low supply voltage while varying the duty cycle of the PWM signal preserves the low-power condition set for this project.

The stepper motor used for the solar tracking system is advantageous to our project due to its design. The stepper motor design includes no need for any type of feedback since the system is of an open loop design. Should anything break the motor never falls dependent on the feedback from the circuit and will thus stop. The motor also cannot be damaged by any type of mechanical overload due to its high reliability, low cost, and relatively high torque at low speeds.

III. SOFTWARE SYSTEMS

PowerBot’s software systems are divided between the two microcontrollers and the iOS application. The code on the MSP430 is designed to drive the motors and control the solar panel system. The PIC32 is programmed to read the ultrasonic sensors, have PowerBot avoid obstacles, and handle the Wi-Fi communication with the iOS application.

A. PIC32 Embedded Software Overview

Controlling the PIC32 and the MRF24WB0MA Wi-Fi module is the embedded runtime algorithm which simultaneously handles incoming TCP data from the iPhone application. The algorithm runs the TCP/IP stack and handles the obstacle avoidance algorithm should an object come close to collision during operation.

The program operates by first initializing all of the MCU parameters such as the peripheral bus clock speed, the master clock speed, and any necessary UART baud rates. After the main initializations are complete, the TCP/IP stack is initialized to allow for Wi-Fi communication. Once the Wi-Fi module is powered on, the TCP/IP stack is loaded and the stack timers are started. Upon loading the TCP stack, a connection is established to a known SSID. If no SSID is available, an ad-hoc network is created that can be joined by any compatible Wi-Fi device. The Microchip Technology, Inc. TCP stack includes support for zero-configuration networking through the use of Bonjour™ which is initialized and can be used with any compatible devices.

Upon finishing the Wi-Fi initialization phase, the software goes into a perpetual control loop handling any incoming network traffic and passing it to the appropriate input and output buffers. It is within this loop that any incoming mobile application commands are received and parsed. The resulting command is then sent to the MSP430 to drive the motors. Commands that will be sent are in the form of high and/or low signals which drive our dedicated motor control pins. Each pin maps to a given car direction: up, down, left, or right. Depending on the desired heading for PowerBot, that pin is set to be a logical 1 (or high). When PowerBot is commanded to stop, the pin is set to a logical 0 (or low).

The perpetual control loop also will probe the four onboard ultrasonic sensors to determine PowerBot’s distance away from objects. Should the robot come within a certain distance of an obstacle, or an obstacle come within a certain distance of PowerBot, the onboard obstacle avoidance routine will be triggered. Once the obstacle avoidance algorithm completes, normal operation will resume.

B. Obstacle Avoidance

The goal of the obstacle avoidance system is to subtly and automatically alter PowerBot’s trajectory to avoid obstacles while driving. This will be accomplished by taking the ultrasonic sensors’ range data and making decisions based on the proximity of obstacles. There are three modes of obstacle avoidance: Active Adjustment (AA), Reverse-Reset (RR), and Off. Technically, AA and RR work simultaneously. An avoidance decision is triggered based on the nature of the robot’s situation.

When PowerBot is under the control of a user, it is steered through the iOS application. When the forward button is held, PowerBot is primarily in AA mode. This means that the left and right range readings of the respective front ultrasonic sensors are compared to see which has a closer obstacle. If one or both sensors indicate readings under a specified minimum threshold, then an AA mode decision is triggered. The obstacle avoidance system will steer slightly to the opposite side of the critical reading to compensate. Once the critical reading has risen above the threshold, PowerBot will be steered back to approximately its original direction. This behavior allows
PowerBot to automatically snake its way through minor obstacles in front of it.

An RR mode decision is made when PowerBot gets dangerously close to an obstacle. If one or more sensors detect a reading beyond this danger threshold, then the obstacle avoidance system stops PowerBot’s movement and switches it into reverse. Using the rear sensors, the obstacle avoidance system steers PowerBot away from the obstacle until it is at a safe distance. Additionally, when an RR mode decision is triggered, a warning is sent to the iOS application, indicating to the user that PowerBot has stopped in an effort to avoid a collision. The user must resume control for PowerBot to continue movement.

If obstacle avoidance is set to Off, PowerBot will be in full manual control with no assistance from the microcontroller.

C. iPhone Application

Any iOS device capable of running iOS 5.1 will be able to use PowerBot’s mobile application. This hardware is commonplace in any given household allowing for easy integration. The iOS SDK relies on the use of Views and View Controllers to handle different subsections of an application with different graphical layouts. These are utilized in PowerBot’s iOS application to provide a unique view that is optimal for the action being performed. The application initializes a TCP socket on a given IP address and port number. If a socket connection should fail to occur, this is an indication that PowerBot is not running. An error message is then displayed stating “Unable to Open a Socket Connection.” Once the socket connection is established, the user can send commands through the use of on-screen buttons. These operate by mapping a given function to a specific hexadecimal value. This is then transmitted to the robot and interpreted as a given command. The buttons used to control PowerBot will also send a command when the user releases a button. This allows a command to be in effect as long as a button is held.

IV. CONCLUSIONS

Reliable battery life and convenient access to a power source are becoming a necessity due to society’s growing dependence on mobile devices. PowerBot is a novel solution to the issue of providing a convenient source of power. Being a remote controlled robot loaded with charging capabilities, PowerBot is useful wherever one takes it. It operates on a four-wheeled chassis, using its internal battery to charge mobile devices. The addition of a solar panel to recharge the internal battery adds to the convenience of using PowerBot in well-lit areas. The iOS application provides a host of convenient features for one’s iPhone, such as PowerBot’s statistics and statuses are readily available. The applications also allows for remote control of PowerBot’s movement through a Wi-Fi connection. An obstacle avoidance system is programmed to assist the user in driving PowerBot by avoiding obstacles.

This project has been quite an educational experience. It quickly became apparent that there is a stark difference between the abstract theories in Electrical Engineering courses and design realizations in the real-world. Throughout the course of this project, the group has learned how to use microcontrollers to interface with peripheral devices. Several formats such as UART, I2C, Serial RS232, A/D conversion, and general purpose inputs and outputs were explored. The experience gained while working on this project will help further our knowledge of the engineering discipline and allow us to apply this knowledge in our future engineering careers. Through the design process and prototyping, we learned that some of our researched methods were more easily implemented than others, and that experience is the best aid when making a new design.

ACKNOWLEDGEMENT

The authors wish to acknowledge Aeronix, Inc. for their generous donation of funds and for their assistance with fabricating printed circuit boards used in the final project.

REFERENCES


BIOGRAPHIES

**Tarik Ait El Fkih** will be graduating from the University of Central Florida with a Bachelor's of Science in Electrical Engineering, minoring Control Systems. He is currently looking for work in a related field with emphasis on control systems and design. He is considering graduate studies in electrical studies or a Master of Business Administration.

**Luke Cremerius** will be graduating from the University of Central Florida in December of 2012 with a Bachelor's of Science in Computer Engineering with a minor in Computer Science. He is currently looking for work in Software Engineering with preferred emphasis on computer networking and mobile application design.

**Marcel Michael** will be graduating in the Fall of 2012 from the University of Central with a Bachelor's of Science in Electrical Engineering. He will be returning for the Spring 2013 semester to pursue a Bachelor's of Science in Computer Engineering with which he will complete in the Fall of 2013 along with a Mathematics minor. He currently works for the Harris Corporation.

**Jerald Slatko** will be graduating from University of Central Florida in the Fall 2012 with a Bachelor's of Science in Electrical Engineering. After gaining industry experience, he plans on going to graduate school to study signal processing and mathematics. He would prefer to stay in academia and would consider a research and/or teaching position.