The Wi-Fi Seeker

Christina Leichtenschlag, Adrian Morgan, Jimmy Wong

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

***Abstract* — Wireless technology is an integral part of modern life, yet the ability to determine the strength of a wireless signal is not often considered. The Wi-Fi Seeker is a robot that can connect to a wireless network and autonomously roam about an area to determine where the selected network's signal is the strongest. Additional features of the Wi-Fi Seeker include control via an Android application, a video feed broadcasting the robot's perspective, an original motor control design, and a solar panel paired with a charge controller to provide the robot with a renewable power source.**

***Index Terms* — Bluetooth, infrared sensors, maximum power point trackers, wireless communication, wireless networks.**

I. Introduction

The inspiration for this project was derived from the modern day need of establishing a good connection the local Wi-Fi. In scenarios where the user knows exactly where the router is located, the user can simply move themselves close to the router in order to receive the best possible wireless connection. In public domains, however, routers are often hidden and the user likely does not know where the router is located. Enter the need for the Wi-Fi Seeker robot; it can autonomously determine where the specified Wi-Fi signal is broadcasting the strongest in an area. There are limitations to this, such as if the router is behind a wall or in another room, therefore the robot will determine the local maximum signal strength and go to that location within its confined boundaries.

The Wi-Fi Seeker robot is remotely controlled by an Android application. The application serves as the user interface that allows the user to command the robot. The hardware of the robot includes a microcontroller, a Bluetooth module, a wireless module, an Internet Protocol (IP) camera, infrared sensors, custom motor controllers, voltage regulators, a solar panel, and a charging controller circuit. The typical use case of the Wi-Fi Seeker robot begins with the user starting up the Android application and completing the three stages of setup. Afterwards, the user is able to manually control the movement of the robot, with the option to start the autonomous functionality. Once initiated, the robot will continue on its mission until completion, at which time the user will be notified via a notification within the application.

II. System Components

Prior to getting into the design, the components of the Wi-Fi Seeker will be introduced and their importance to the final product will be explained.

A. Microcontroller

The brain of the Wi-Fi Seeker robot is an MSP430F5529 microcontroller. Chosen for its bountiful 80-pin layout, 8KB of RAM, 128KB of memory, and 25kHz clock speed, this low-powered microcontroller was perfect for this application. [1] The MSP430F5529 connects to both the Bluetooth and wireless modules via UART, and connects to the infrared sensors via GPIO pins. The MSP430F5529 is responsible for organizing the data from its various inputs and sending it back to the Android application. It is also responsible for the logic behind the autonomous algorithm. The MSP430F5529 is programmed by connecting four specific pins to an MSP430G2553, which in turn is connected directly to a computer and programmed using USB.

B. Android Application

To provide a complete user interface for the Wi-Fi Seeker robot, an Android application was developed. Programmed for operating system version KitKat, the application handles all setup required before the robot is operational, then takes all user-inputted commands and forwards them to the robot. The internal Bluetooth capability of all Android-compatible devices is essential, since the application uses Bluetooth in order to communicate with the MSP430F5529 microcontroller.

C. Bluetooth Module

In order for the commands and data inputted by the user in the Android application to affect the robot, Bluetooth was chosen as the means for wireless communication between the two devices. The RN41 XV Bluetooth module operates at 3.3V and 30mA with a sustained data rate of 240 Kbps. [2] The module is set to transmit at 115,200 baud, which is fast enough to accurately transmit all character sequences between the application and the microcontroller.

D. Wireless Module

The ability to determine the strength of a wireless signal is achieved using the RN-XV Wifly module. Equipped with the function "show rssi", the result of this function is an integer measurement of the current Received Signal Strength Indication (RSSI) of the selected wireless network, which will later be the key component for the autonomous algorithm. This module operates at 3.3Vand 38mA, and is also configured to run at 115,200 baud. [3]

E. Internet Protocol Camera

The feature of a video feed was subsequently added to the Wi-Fi Seeker robot. Thus it was decided to use the D-Link DCS-932L IP camera. Operating at 5V and 1.2A, this camera is entirely responsible for transmitting the video feed to a specified IP address on the local network.

F. Sensors

Four GP2Y0A21YK infrared sensors are mounted to the robot chassis to help aid the robot with its autonomous mission. The sensors are used to detect obstacles in the robot's path, and this information is used to help the robot navigate around said obstacles. Each sensor operates at 5V and 2mA, and has a detection range up to 80cm.

G. Motors

The robot body chosen for the Wi-Fi Seeker robot is the Wild Thumper 4WD chassis, which comes with four motors. Each motor has a 34:1 gear ratio, which can reach a maximum rate of 350 RPM. Each motor also has a stall torque of 5 kg-cm per motor at 7.2 V.

H. Motor Controllers

In order to reduce some of the processing that must be done by the MSP430F5529, custom motor controllers were designed specifically for the Wi-Fi Seeker robot. The two bipolar junction transistors (BJTs) that are used to create each motor controller are the MJH6287 PNP and the MJH6284 NPN. These transistors are capable of supporting up to 100V and 20A. A third transistor controls these BJTs, which is the 2N5551 that can support up to 160V and 600mA.

I. Lithium-polymer Battery

Two lithium-polymer batteries are responsible for supplying power to all of the components on the robot. Each battery by Venom is rated at 7.4V, has a capacity of 3200mAh, and has a discharge rate of 20C. This means each battery can supply up to 64A of current continuously. The maximum current the robot will draw is 28A, therefore these batteries are able to supply more than enough power to the robot.

J. Solar Panel

One goal of this project was to use renewable energy to power the robot. In order to reach this goal, a solar panel was added to recharge the batteries. The selected solar panel from Solartech Power Inc. is rated at 10W. At its maximum power point, it has a voltage of 17.3V and can supply 590mA of current. This panel is able to fully recharge a battery in only 5.4 hours.

K. Charge Controller

Since the lithium-polymer batteries are going to be recharged, a charge controller is a necessity. This is because lithium-polymer batteries can be extremely dangerous if charged improperly. The chosen charge controller is the Texas Instruments BQ24650. It can support solar panel voltages ranging from 5-28V, and it can handle up to 8A of current. This controller charges the battery using a constant current and voltage cycle. It also implements maximum power point tracking, which allows the solar panel to operate at maximum efficiency.

L. Voltage Regulators

Voltage regulators are used to ensure that all of the components are supplied with the correct power. In order to increase efficiency and minimize the heat due to power loss, switch mode is used instead of linear. Due to the power specifications of the various components mentioned, both 3.3V and 5V regulators are used. The 3.3V regulator is a Texas Instruments LM2592HV operating in buck mode, and can handle up to 2A of current. The 5V regulator is a Texas Instruments LM22670 operating in buck mode, which can handle up to 3A of current.

III. SYSTEM DESIGN

A block diagram of the Wi-Fi Seeker robot can be seen in Fig. 1. The subsystems of the robot include: the Android application system, the wireless communication system, the motion system, and the power system. The Android application system includes the functionality of the application, and must be able to handle any and all user inputs. The wireless communication system involves how the data is handled between the wireless module and microcontroller as well as between the application, Bluetooth module, and microcontroller. The motion system includes the hardware and software responsible for the motion of the robot. The power system includes the lithium-polymer batteries, which power all of the other components, and the solar panel, which charges the battery.

A. Android Application System

The Android application system includes the three stages of setup, which involves connecting to the

Bluetooth module, accessing the IP camera feed, and selecting and connecting to a wireless network. Once setup is complete, the user enters manual control mode, where the user can directly control the motion of the robot. From here, the user can activate the autonomous functionality of the robot. All of the data received as user input is transmitted wirelessly to the Bluetooth module, which is then received by the MSP430F5529 as an interrupt.

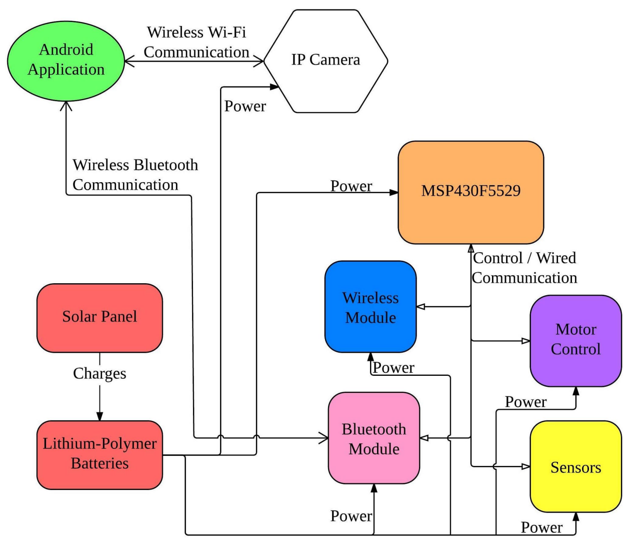


Fig. 1. Block diagram of the Wi-Fi Seeker robot.

B. Wireless Communication System

There are multiple means of wireless communication embedded within this project. One instance is the Bluetooth communication that takes place between the Android application and the Bluetooth module, which communicates via hardware UART with the MSP430F5529. Second we have the wireless module that communicates with the various wireless networks, and similarly the data is sent to the MSP430F5529 via hardware UART. Finally we have the IP camera video feed, which is streamed directly to an IP address. The Android application directly connects to the IP address to access the feed, thereby creating the third instance of wireless communication.

C. Motion System

The motion system encompasses two types of motion: user controlled, and autonomous. The logic for manual control and for the autonomous algorithm will be coded for the MSP430F5529 microcontroller and will be initiated by a user command. When motion occurs, the infrared sensors and motor controllers are utilized. The sensors are used as a precaution that can affect the direction of motion, while the motor controller causes the motion to actually take place.

D. Power System

The Wi-Fi Seeker robot's power system is supplied by two 7.4V, 3200mAh lithium-polymer batteries. Two batteries will be used instead of one to extend the battery life of the robot. Since the voltage of each battery is greater than what is required for various components, the voltages must be stepped down and regulated. In order to power the MSP430F5529, Bluetooth, and Wifly modules, the voltage will be stepped down to 3.3V. In order to power the infrared sensors and IP camera, the input voltage will be stepped down to 5V. The motor controllers can be directly connected without regulation.

The second section of the power system is the recharging of the batteries using solar power. The solar panel will connect to the charge controller, then to the batteries using a single pole double throw switch. The switch will allow for the user to choose whether s/he wants to recharge the batteries or to use the Wi-Fi Seeker robot.

IV. Software Detail

The two major pieces of software for this project are the Android application and the code for the microcontroller. In this section both pieces will be examined in close detail.

A. Android Application

The application begins with setup. Before any communication between the application and the microcontroller can take place, a Bluetooth connection must first be established. Therefore in the first stage of setup the user is prompted to turn on their Bluetooth if it is not already on. The user can then click a button to scan for Bluetooth devices, causing a list to populate containing the names of all discovered devices. By clicking on the name of the Bluetooth module, the device will attempt to pair with the module. Upon successful pairing, an instance of the BluetoothAdapter class is created. All future communication with the MSP430F5529 will be executed by sending and receiving data through this adapter. The application will then automatically move onto the next stage of setup. Should the pairing fail, the user simply needs to try again. [4]

The second stage of setup involves connecting to the IP camera. The address for the camera is already coded into the application, but in the event the user wants to enter another URL the option is available. The user must click on a button to refresh the camera feed. The video feed is contained within an instance of the WebView class, and pressing the refresh button activates the code that will update the WebView. Upon confirming that the video stream is working, the user must click a second button to continue the setup process.

The third stage of setup involves selecting exactly which wireless network the Wi-Fi Seeker robot will be seeking on, which means communicating with the wireless module. This process, depicted in Fig. 2, begins with registering the buttons pushed on the screen as commands that are sent via Bluetooth to the MSP430F5529, which recognizes the command and then sends the appropriate data to the wireless module. All responses from the wireless module are routed back through the MSP430F5529, sent to the Bluetooth module, and then the Bluetooth module wirelessly sends the data to the Android application where it is received and analyzed. As per the instructions displayed on the screen in the application, the user must first turn on command mode, which puts the wireless module in a state such that the "show rssi" command can be used. Next, the module must be properly configured, which is done by hitting the button for command setup.

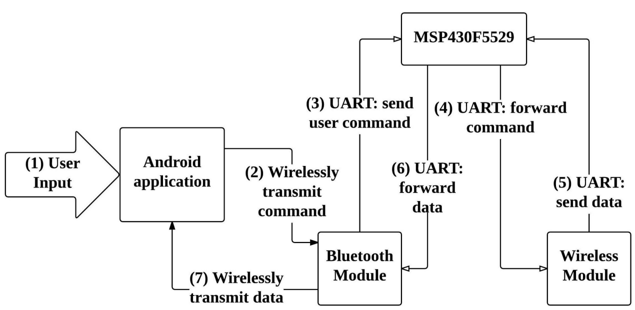


Fig. 2. Flow for sending commands and receiving data between the Android application and the wireless module.

The user can now scan for wireless networks in the area in order to select the network that s/he wants to seek on. Once the button to scan is hit, a dialog will pop up prompting the user to select a network from a list of detected networks. After selecting a network, a second dialog will appear prompting the user for a password. Once the password is submitted, the wireless module will connect to the specified network. If the connection was successful, the user can click the button to complete setup. Doing so will bring the user to the manual control screen.

The manual control is straightforward. At the top of the view is text that displays the current RSSI value, i.e. the one that was last measured. Below it is the button that will cause the robot to begin its autonomous functionality. Further down is the container for the video feed that should be constantly updating. Finally there are four buttons that manually control the robot's motion: forward, left, right, and reverse. When the user clicks the button to begin the autonomous functionality, the motion control buttons disappear, and upon completion of the autonomous functionality a dialog appears to alert the user. Once the user acknowledges the dialog, the application will again display the manual control screen. The flow of the entire Android application can be seen in Fig. 3.

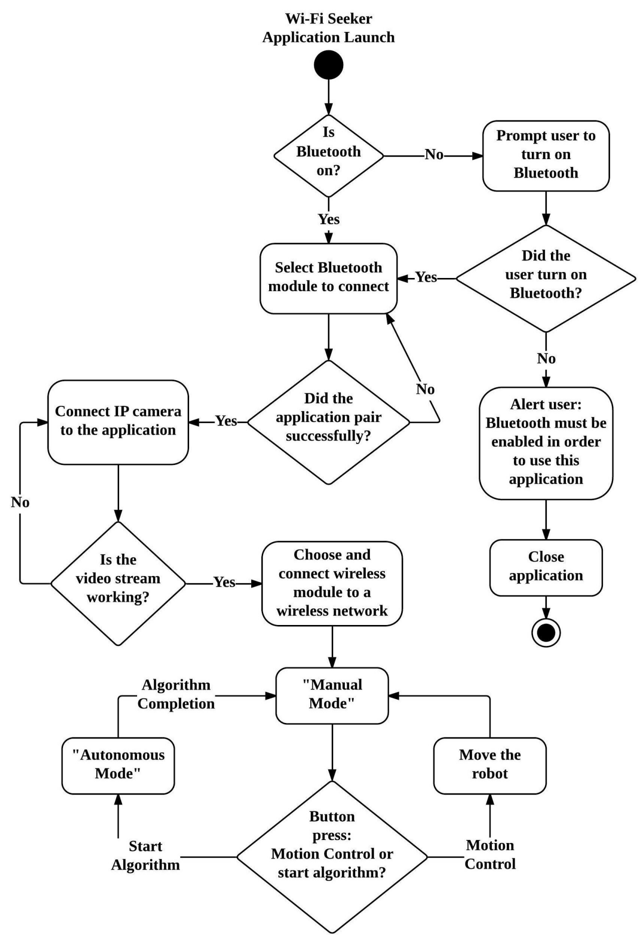


Fig. 3. State diagram for the Android application.

B. Microcontroller Code

The flow of the code on the MSP430F5529 is driven mainly by the hardware UART interrupts caused when data is received by either the Bluetooth or wireless modules. Since the user control is the most important input, the microcontroller mainly reacts to the data received from the UART interrupt connected to the

Bluetooth module. When a command is received, the MSP430F5529 must first determine what the command means. While there are only two categorical possibilities, which are to either send a command to the wireless module or to move the robot by signaling the motor controllers, there are several possibilities for each type of command. However, by using a switch statement it is simple to determine exactly which command the user entered into the Android application.

While the user is entering settings or in manual mode, the microcontroller is simply responding to the commands when the UART interrupt tells it to respond. However once the autonomous functionality begins, user commands are neither wanted nor needed, for the interim. The basis for the autonomous functionality is as follows: the purpose is to move the robot to the location where the wireless signal is strongest. The strength of the wireless signal is determined by the RSSI value, which is an integer returned by the "show rssi" command that is sent to the wireless module. An RSSI value near 0 dBm means the module is extremely close to the router, while an RSSI value near -85 dBm means the module is barely picking up the wireless signal.

The autonomous algorithm begins by taking an initial sample of the RSSI of the network. The robot will then move forward, and stop to take another measurement of the RSSI. If the new RSSI is stronger than the previous value, the previous value will be updated, and the robot will continue to move forward in its current direction. If the new RSSI is approximately the same value, the robot will make a note (by incrementing a counter variable) and continue to move forward in its current direction. If the new RSSI is weaker than the previous value, the robot will change its direction by first reversing n times, where n is half of the value of the counter variable. Then the robot will turn and move forward in a new direction. Fig. 4 depicts how the robot will move as the RSSI changes. Once the robot measures an RSSI within an acceptable range, the autonomous algorithm is finished and the user is alerted of the completion of the algorithm.

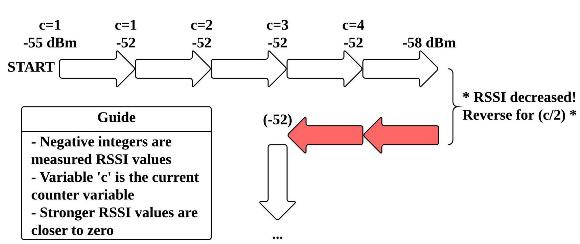


Fig. 4. Image depicting the basis of the autonomous algorithm.

The basic algorithm is missing a couple of key details, the first being, what happens when the robot encounters an obstacle? Fig. 5 outlines the entire algorithm, which is as follows: Every time the robot attempts to move forward, the infrared sensors are pinged to determine whether or not there is an obstacle in the way. If an obstacle is registered, the robot will first wait a few microseconds and then take a second measurement. This is to ensure that there is actually an obstacle, and not either a rogue measurement or a temporary object in the way, such as if a person walked in the robot's path. Should the robot verify that there is an object in the way, it will reverse and turn left, then continue with its algorithm. If the obstacle is not verified, the robot will move forward as it originally intended.

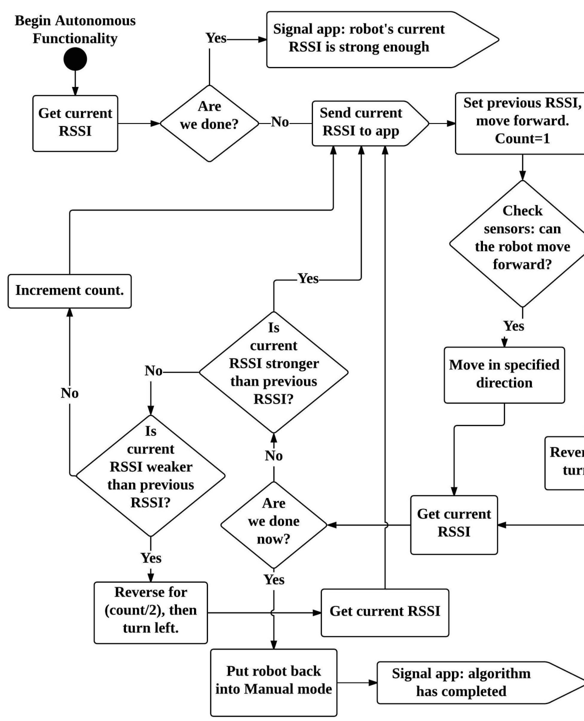


Fig. 5. State diagram for the complete autonomous algorithm.

The second key detail is, what happens if the robot is incapable of finding a location where the measured RSSI is within the acceptable range? A major challenge with the Wi-Fi Seeker application is that there is no confined space, especially if the user is outside. This means that it would be highly expensive to map out every movement made by the robot and the corresponding RSSIs for each location. Instead, the number of moves the robot makes will be counted each time a new strongest RSSI is encountered.

At the beginning of the algorithm, the local maximum RSSI is set to the current RSSI. Each time the robot moves and gets the RSSI of its current location, it will be compared to the value for the strongest RSSI so far, which is the local maximum. If the current RSSI is stronger, the local maximum RSSI updates its value, and the count resets. For every move the robot makes after setting its new local maximum RSSI, a counter for the number of moves will be incremented. If the counter reaches 15 or higher, this means that in 15 or more moves a stronger RSSI value was not found, which essentially means a lower RSSI value does not exist. As the counter is incrementing, the robot will be traveling in circles around the local maximum RSSI. Once the counter reaches 15, the threshold for an acceptable RSSI value will be adjusted to the local maximum RSSI. This will allow for the robot to complete its algorithm once it circles back to the location where it measured the local maximum RSSI value. Upon reaching this location the robot will notify the user that the autonomous algorithm has completed.

C. The Wifly Module

The wireless module is crucial to the Wi-Fi Seeker robot, because without it there would be no way to determine the RSSI of a network. This module comes equipped with various commands that affect how the module is used. In order to execute any of the commands, the module must first be placed in what is called "command mode". This mode is the key to connecting to and evaluating the signal strength of a network. In the application of the Wi-Fi Seeker, the module must first be put into command mode, which is done using command "$$$". Next, in order to connect to a specific wireless network, a few parameters must be specified, such as setting the encryption level and telling the module to only join a network when the command is given. The "scan" command is how the module discovers networks within its range. This data is parsed to retrieve the network names, and is presented to the user on the Android application. When the user wishes to join a network, the password must be set before the "join" command is issued. Therefore the user will enter a password for the network, the password will be set for the module, and finally the module will be commanded to join the user-selected network. [5] Once the module is connected, various statistics and data about the network can be retrieved. The only one of importance is the RSSI, which is gathered by sending the "show rssi" command to the module. The returned character data is parsed and converted to an integer, which is then used in the autonomous algorithm.

V. HARDWARE DETAIL

The components of the Wi-Fi Seeker robot have been introduced in Section II, but now these parts and their systems will be explained in greater technical detail. This includes the components in the motion and power systems.

A. The Motor Controllers

Each H-bridge will be connected to two DC motors, a DC power supply, and the MSP430F5529, therefore creating a differential-drive system. Since the microcontroller doesn't provide enough voltage to power each DC motor on the robot, it sends a signal to the H-bridge through a pulse width modulation (PWM) to manipulate the DC power supply to regulate the speed that the robot will accelerate.

Each motor controller is capable of speed control by simply sending an analog signal from the microcontroller to create a PWM signal for each motor controller. The analog signal is used to both control the direction of rotation and to alter the PWM to control the speed of the motors. The control signals are sent to the H-bridge to determine the direction that the transistor will allow current to flow through to the motors. The PWM determines the duty cycle duration that affects how much current is allowed to flow during each cycle, which enables speed control of the motors.

To implement forward motion, one side of both H-bridges will be set to high to allow current to flow in one direction and the opposite will be done for the other sides. To move left and right, one of the H-bridge sides will be set high and the opposite will be done to the other H-bridge. The robot will also have a constant forward, reverse, and steering speed set by the duty cycle of the PWM.

B. Regulating the Batteries

As previously mentioned, the Wi-Fi Seeker is powered by two batteries. One of the batteries is connected solely to the motor controllers, since this subsystem draws the most amount of current. The other battery is used to power all other components on the robot.

The voltage needed to power each motor controller is 7.2V, but each battery is rated at 7.4V. However, the voltage sent to the motor controllers can be modified with pulse width modulation, which means that the power coming from the battery does not need to be regulated. Therefore each motor controller will connect directly to the first battery.

As seen in Section II, there are various other components that connect to the robot, but they all operate at either 3.3V or 5V. In order to accommodate for these desired voltages, switching voltage regulators were used. The output of the 3.3V regulator connects to the MCU and the wireless communication modules. The output of the 5V regulator connects to the sensors and the camera. Both regulators are connected to the second battery.

C. The Charge Controller

The BQ24650 charges the lithium-polymer battery in different phases. First, if the battery is completely depleted, a pre-condition phase will charge the battery at only 10% of the charging rate. This phase prepares the battery to accept more charge. Second, a constant-current fast charging phase charges at 100% of the charging rate until the voltage increases to a certain point. Third, a constant-voltage phase supplies enough current to keep the voltage constant, therefore as the battery continues to charge, the current will decrease. Lastly, the charging will terminate once the charging current drops below 10% of the charging rate.

The regulation of the battery voltage is programmed using a resistor divider at the output. The battery has a nominal voltage of 7.4V but when it is fully charged it reaches a maximum voltage of 8.4V. In order to calculate the battery regulation voltage, use (1) and set V_{BAT} equal 8.4V and R1 equal 100kΩ. R2 was calculated to be 300kΩ.

$$V_{BAT} = 2.1V * \left[1 + \frac{R2}{R1}\right] \quad (1)$$

The charging current is determined by a sense resistor. The battery manufacturer recommends charging the battery at a maximum of 1C, which is 3.2A. The charging current was decided to be 3A. The equation to calculate the charging current is shown in (2).

$$I_{CHARGE} = \frac{40mV}{R_{SR}} \quad (2)$$

Setting I_{CHARGE} to 3A, the sense resistor R_{SR} was calculated to be 13.3mΩ. The charging rate is important because other parameters are determined based on it. The pre-charging rate is 10% of the charging rate, or 300mA. This charging termination will also occur when the current supplied to the battery is less than 10% of the charging rate.

The chip also implements maximum power point tracking for the solar panel by regulating the input voltage. This enables the solar panel to output its maximum power. It uses a basic algorithm and is preset with a constant voltage. The voltage at which the solar panel is at its maximum power is at 17.3V. The equation used to calculate the resistors needed is shown in (3), where V_{MPPSET} is set to 17.3V and R4 is set to 35kΩ. R3 was calculated to be 469.5kΩ.

$$V_{MPPSET} = 1.2V * \left[1 + \frac{R3}{R4}\right] \quad (3)$$

VI. Printed Circuit Board Design

The custom printed circuit boards (PCBs) used for the Wi-Fi Seeker robot were designed by all three members of Group 30 and were fabricated by OSH Park. In total there are four PCB designs that were used in the final prototype of the robot.

A. The Main Board

This PCB is contains the brain of the Wi-Fi Seeker robot as well as the wireless components. This board houses the MSP430F5529 chip and the crystal for it, headers for both the Bluetooth and wireless modules, as well as headers for the control of the infrared sensors sensors and motor controllers. It also has LEDs that are used to indicate the current behavior of the robot. This board was implemented using a two-layer design, and the components are almost exclusively mounted on the surface.

B. The Motor Controller Board

Since there are four motors, and each motor controller controls two motors, it was decided the motor controller should have its own board in the event that we needed to put one motor controller on either side of the robot. Also implemented using a two-layer design, this board simply contains the motor control circuit. As a result of the potentially high voltage and current, the components of this board are through-hole. Also, each controller will feature thicker traces in order to support the current that needs to be pulled along the path.

C. The Voltage Regulator Board

Two voltage regulators were required for the robot, since the microcontroller and both modules require 3.3V, while the sensors and IP camera require 5V. The regulators were originally included on the main board, but because of a component stock issue both regulators had to be completely redesigned. As a precaution, the regulators were placed on their own board in case this issue occurred again. This board is implemented using a two-layer design and its components are surface-mounted. The board was designed with a ground plane on the top layer and the bottom layer. The thermal pads of the integrated circuit had multiple vias placed, which creates a connection to the ground plane. This allows the heat from the integrated circuit of the regulators to dissipate throughout the board, which prevents the regulators from destruction.

D. The Charge Controller Board

The MPPT integrated circuit chosen for the Wi-Fi Seeker robot is the BQ24650, which was originally designed by Texas Instruments to be implemented using a four-layer board. However, after communicating directly with members of TI, the authors were able to redesign the board using only two-layers. The components of this board are all surface-mounted. The board was also designed with a ground plane on the top and bottom layers similarly to the regulators.

VII. Conclusion

Designing and building the Wi-Fi Seeker robot has provided each member of 30 with invaluable experience. By overcoming the challenges presented throughout the Senior Design program, the authors learned many different life lessons, from working in a team and learning to rely on the other team members, to practicing writing professional technical documentation, to meeting deadlines on a strict timeline. By creating a robot with various subsystems, each team member was able to get hands-on experience in different topics of the Electrical and Computer Engineering disciplines. Each system of the robot challenged the team in its own way, whether it was tinkering with circuits or debugging code, and integrating these subsystems proved even more difficult. In the end, the members of Group 30 were able to build a functional Wi-Fi Seeker prototype.

Acknowledgement

The authors wish to acknowledge Dr. Samuel Richie for his advice and guidance with this project. Additionally, the authors would like to thank Leidos and Duke Energy for their sponsorship of the project.

References

[1] MSP430F5529 Datasheet. (2013, May 1). Retrieved March 9, 2015, from http://www.ti.com/lit/ds/symlink/msp430f5529.pdf

[2] RN41XV & RN42XV Bluetooth Module. (2012, March 12). Retrieved March 3, 2015, from http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/RN42XV.pdf

[3] RN-171-XV 802.11 b/g Wireless LAN Module. (2012, October 29). Retrieved March 19, 2015, from https://cdn.sparkfun.com/datasheets/Wireless/WiFi/rn-171-xv-ds-v1.04r.pdf

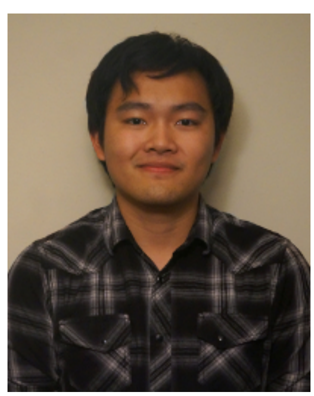
[4] Bluetooth. (n.d.). Retrieved March 24, 2015, from http://developer.android.com/guide/topics/connectivity/bluetooth.html

[5] WiFly Command Reference, Advanced Features & Applications User's Guide. (2013, April 30). Retrieved March 17, 2015, from https://cdn.sparkfun.com/datasheets/Wireless/WiFi/rn-wiflycr-ug-v1.2r.pdf

The Engineers

Christina Leichtenschlag is a senior Computer Engineering student. With a passion for programming, she has focused her studies on software development, specifically with back-end systems and with mobile development in Android. She has obtained two software engineering internship positions, and after graduating in May 2015 she will join Amazon as a Software Development Engineer.

Adrian Morgan is a senior Electrical Engineering student. With his favorite topic being electronics design, he has concentrated his coursework on analog and digital design. He currently holds an engineering internship at Cubic Simulation Systems Division, and will continue as an Associate Electrical Engineer after he graduates in May 2015.



Jimmy Wong is a senior Electrical Engineering student. He is most interested in semiconductor devices and electronic design. After he graduates with his degree in May 2015, he plans on finding a job to start his career as an Electrical Engineer.