# 3. Approach

[little intro to wirad]

## 3.1 Functional Overview

[put stuff in, with picture]

## 3.2 Hardware Design

[do a little hardware design intro]

### 3.2.1 Microcontroller

#### 3.2.1.1 Microcontroller Approaches

The central processing requirement of this project is to interface the various system components. Such interfacing involves data movement, not intense computation. Consequently, a microcontroller was chosen for this task, rather than a full-fledged microprocessor. Microcontrollers are cheaper, simpler, and sufficient for data movement.

The chief factors involved in microcontroller choice were cost, clock speed, program memory, random access memory (RAM), and on-chip capabilities. Initial efforts focused on Microchip's PIC series of microcontrollers. While these microcontrollers exhibit adequate clock speeds at low cost, very few contain on-chip USB controllers and none contained on-chip MP3 hardware decoders. If a PIC-based design had been pursued, these functions would almost certainly have been delegated off-chip. The additional cost, complexity, and component count of this design precluded the use of PIC microcontrollers.

The search for greater on-chip functionality yielded the Atmel AT89C51SND1C. This microcontroller is intended for use in portable consumer electronics devices and includes an on-chip USB controller and MP3 hardware decoder. The Atmel device also contained more program memory and RAM when compared to the PIC. Below is a summarized comparison.

**Table I: Microcontroller Comparison**

| | PIC18F45J10 | Atmel AT89C51SND1C |
|---|---|---|
| **Maximum clock speed** | 40 MHz | 20 MHz |
| **Program memory** | 32 KB | 64 KB |
| **RAM** | 1 KB | 2 KB |
| **Price** | $4.25 | $9.00 |
| **Integrated USB** | No | Yes |
| **Integrated MP3 decoder** | No | Yes |

As a result of superior on-chip functionality and increased memory, the Atmel AT89C51SND1C was chosen for this design.

#### 3.2.1.2 Atmel AT89C51SND1C Overview

The Atmel AT89C51SNDC is an 8-bit 8051 architecture microcontroller available in an 80-pin surface-mount package. It includes 2304 bytes of internal RAM, 64KB of program memory, and 4KB of boot flash memory. In addition to the on-chip USB controller and MP3 hardware decoder, the AT89C51SNDC also contains specialized hardware for serial peripheral interface (SPI), two-wire interface (TWI), multimedia card (MMC) interface, and a full duplex universal asynchronous receiver and transmitter (UART). The major

### 3.2.2 WiFi

#### 3.2.2.1 WiFi Approaches

WiRAD's WiFi aspect is arguably its only significantly innovation. In an effort to ensure manufacturability, speed, and processor independence, three approaches to achieving WiFi connectivity were explored.

The first was to utilize a radio frequency (RF) transceiver and a software TCP/IP stack implemented on the microcontroller. While the manufacturability of such a design is high due to simple component circuitry, this method exhibits no independence from the microcontroller, which would necessarily be involved in the movement of each bit. Consequently, speed would be severely limited as a result of such a design.

[put details about other two approaches here]

#### 3.2.2.2 Airborne WLNB-AN-DP100 Details

[put module details here]

### 3.2.3 Antenna

#### 3.2.3.1 Antenna Approaches

Antennas come in numerous shapes and sizes. When selecting an antenna, the application of the antenna is the most important factor. For instance, a directional antenna provides a good signal in an aimed position. In addition, a portable device needs a small, lightweight antenna. There are many different types of antennas are available. The following table is a quick glance at the different technologies.

Table I. Different Antenna Technologies

| **Antenna Technologies** | **Highly Directional (yes/no)** | **Lightweight (yes/no)** | **Common Applications** |
|---|---|---|---|
| Helix Wire | NO | YES | Mobile Phones, Broadband |
| Aperture | YES | YES | Microwave Frequencies |
| Microstrip | NO | YES | Aircraft, Spacecraft |
| Array | NO | YES | Mobile phones, Broadband |
| Reflector | YES | NO | Radar, Satellite |
| Lens | YES | NO | Radar, Satellite |

Wire antennas are the most commonly used antenna. There are various shapes of wire antennas such as a straight wire (dipole), loop, and helix (spring shaped). Helix antennas are commonly used for broadband applications. Helix antennas are capable of omni-directional characteristics; however, it is complicated to achieve good multiband performance.

Aperture antennas have become more familiar with the need for more sophisticated forms of antennas and the utilization of higher frequencies. Horn and rectangular waveguides are the more common forms of aperture antennas. These antennas can be covered with a dielectric material to protect them from hazardous conditions of the environment; however, these antennas are highly directional antennas.

Microstrip antennas consist of a metallic patch on a grounded substrate. The metallic patch can take many different configurations; however, the rectangular and circular patches are the most popular because of ease of analysis and fabrication. These small, durable antennas are simple and inexpensive to fabricate using modern printed-circuit technology. Microstrip antennas are low efficiency, low power, and have a narrow bandwidth.

Array antennas are used for applications that require radiation characteristics that may not be achievable by a single element. Array antennas are a combination of similar antennas working at different frequencies. Array antennas can broaden the range of working frequencies (1-5 GHz) as well as offer a variety of different working frequencies (2 GHz and 14 GHz).

Reflector antennas are commonly used for the transmission and reception of signals that have traveled millions of miles. Common reflector antenna forms are corner reflector and parabolic (satellite receivers). Reflector antennas are large and highly directional.

Lens antennas are used to focus all energy to a desired point, preventing energy from spreading in undesired directions. They can be used in most of the same applications as the parabolic, especially at high frequencies. Lens antennas are large, heavy, and highly directional.

#### 3.2.3.2 Wireless Approach Justification

WiRAD's need for portability eliminates many of these antennas. The highly directional antennas, lens, aperture, and reflector antennas, were eliminated because WiRAD must receive information regardless of the direction it is facing. The array antenna was eliminated because 802.11 networks work at a bandwidth, 2.4 – 2.48 Ghz, narrow enough to be achieved by one antenna.

The most suitable antennas for WiRAD are the helix wire and the microstrip antenna. The following table provides two different antennas we considered.

Table II. Comparison of Microstrip and Helix Antennas

| | ***Shaper Concepts* 2.4 GHz 8dBi Flat Patch Antenna** | **2.4 GHz Rubber Duck 2.2 dBi RP-SMA Antenna** |
|---|---|---|
| **Antenna Technology** | **Microstip** | **Helix Wire** |
| **Frequency** | 2400-2500 MHz | 2400-2500 MHz |
| **Gain** | 8 dBi | 2.2 dBi |
| **Impedance** | 50 Ohm | 50 Ohm |
| **Weight** | 0.4 lbs. (.18 Kg) | 0.52 oz. (15 g) |
| **Dimensions** | (L x W x H)<br>4.5 x 4.5 x .9 inches, 114 x 114 x 23 mm | (Length x Radius)<br>4.7 x 0.4 inches,<br>105 x 10 mm |
| **Operating Temperature** | -40° C to 85° C<br>(-40° F to 185° F) | -40° C to 85° C<br>(-40° F to 185° F) |
| **Polarization** | Vertical or Horizontal | Vertical or Horizontal |
| **Cost** | $ 19.95 dollars | $ 5.95 dollars |

As shown in the Table II, the performance of the microstrip antenna and helix wire antenna are comparable to each other. The microstrip antenna offers a smaller solution, but at a much larger price. However, constructing a microstrip antenna can reduce costs. With the use of a 3-D, full wave electromagnetic field simulation software, HFSS™, the cost of the microstrip antenna is reduced to $0.50 dollars. Thus, constructing a microstrip antenna would be the smallest and lowest cost solution.

**3.2.3.3 Antenna Design**

The microstrip antenna consists of a patch antenna (Figure 1) and a ground plate separated by a dielectric (Figure 2).

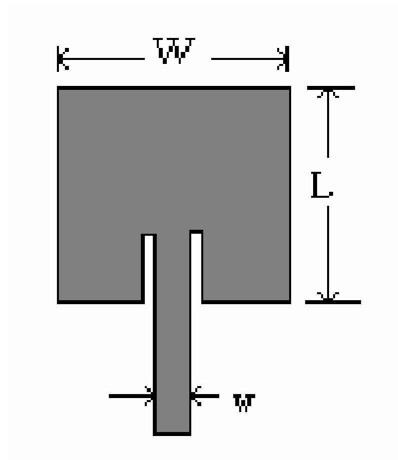


Figure 1: Square patch antenna

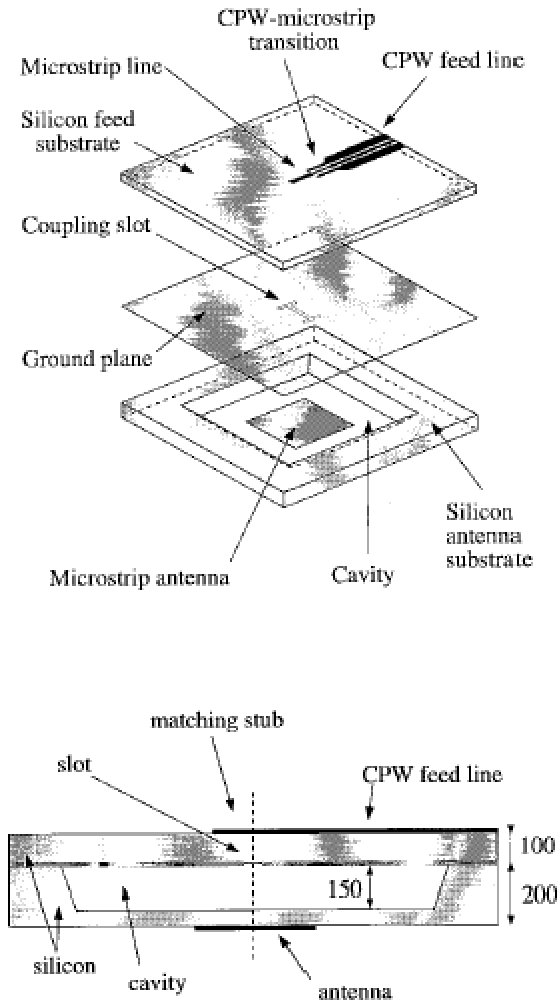


Figure 2: Microstrip antenna

The dimensions (L) and (W) of the patch antenna were found by the following wavelength-dimension equations:

$$W = \frac{\lambda_o}{2} * \left(\frac{\varepsilon_r + 1}{2}\right)^{-\frac{1}{2}} = \frac{3x10^8}{2 * 2.4x10^9} * \left(\frac{4.4+1}{2}\right)^{-\frac{1}{2}} = 3.26cm$$

$$\varepsilon_{eff} = \frac{(\varepsilon_r + 1)}{2} * \frac{(\varepsilon_r - 1)}{2} \frac{1}{\left(1 + 12\frac{W}{h}\right)^{\frac{1}{2}}} = \frac{(4.4+1)}{2} * \frac{(4.4-1)}{2} \frac{1}{\left(1 + 12\frac{3.26cm}{0.15cm}\right)^{\frac{1}{2}}} = 4.0645$$

$$\Delta L = 0.412h\frac{(\varepsilon_{eff}+0.3)}{(\varepsilon_{eff}-0.258)}*\frac{\left(\frac{W}{h}+0.264\right)}{\left(\frac{W}{h}+0.8\right)} = 0.412*0.15cm\frac{(4.0645+0.3)}{(4.0645-0.258)}*\frac{\left(\frac{3.26cm}{.15cm}+0.264\right)}{\left(\frac{3.26cm}{.15cm}+0.8\right)}$$

$$\Delta L = 0.06917cm$$

$$L = \frac{1}{\left(2f_r\sqrt{\varepsilon_{eff}}\sqrt{u_o\varepsilon_o}\right)}*2\Delta L = \frac{V_o}{2f_r\sqrt{\varepsilon_{eff}}}*2\Delta L = \frac{1}{\left(2*2.4x10^9\sqrt{4.0645}\right)}*2*0.06917cm$$

$$L = 2.52cm$$

### 3.2.4 Power System

WiRAD requires a battery that is small and light weight for mobility. WiRAD needs a battery that can be recharged and has over 2000 mA-H for usability and cost effectiveness. There are a few different ways to connect the battery to the device. Batteries that are connected in parallel increase the current output of the power system. Batteries connected in series increase voltage output of the power system. Due to the size and the usage of WiRAD, a dependable, long lasting, and small battery is required. Several options had to be considered.

#### 3.2.4.1 Battery

The three batteries that were considered are Nickel Cadmium (NiCd), Nickel Metal Hydride (NiMH), and Lithium Ion Polymer (LiPo). Each battery had its pros and cons.

| | **NiMH** | **NiCd** | **LiPo** |
|---|---|---|---|
| **mA-H** | 3000 | 3000 | 2700 |
| **weight** | 260g | 215g | 70g |
| **charge cycles** | 400 | 800 | 500 |
| **cost** | $10.99 | $6.27 | $12.19 |

Table 1. Battery comparisons.

Based on the table above it can be inferred that all three batteries that were compared have similar mA-H ratings. The NiMH battery has to be connected in parallel to achive the mA-H that is required. The NiCd battery has to be connected in a series to achieve the 3.3 voltage required by the circuit. The connections are demonstrated in figure 1 below.

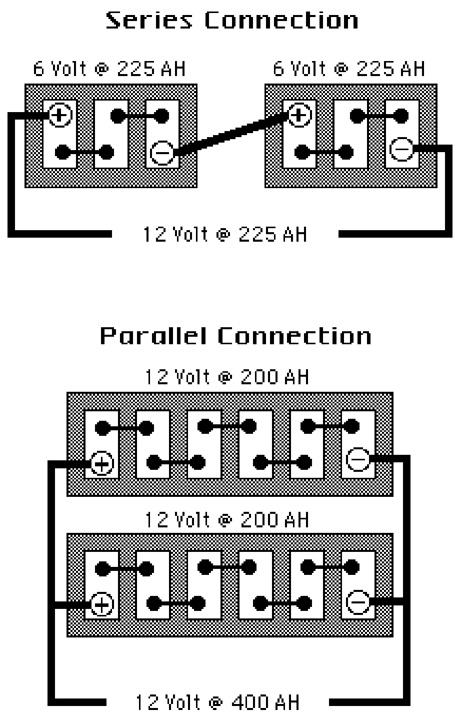

Figure 1. Battery connection types [1].

The primary difference between all three batteries is the weight and size. The LiPo battery has the best weight and dimensions in comparison with the other two batteries. The price of the battery is slightly higher but because the primary constraint to consider here is the portability of the device, the LiPo battery is the clear choice.

The circuit uses a maximum current draw of under 500mA-H with the LiPo battery, WiRAD will operate for five hours and twenty hour minutes. The calculations can be seen below.

$T_{op} = 2700/500$

$T_{op} = 5.4\ hours$

The major components and their maximum current draw are listed below in table 2.

| **component** | **voltage** | **current draw** |
|---|---|---|
| WiFi Module | 3.3 | 450mA |
| Atmel Microprocessor | 3.3 | 35 mA |
| Protection Circuit | 3.7 | 5 mA |
| LEDs | 3.3 | 10mA |

Table 2 Major Components.

**3.2.4.2 Charging**

One of the main factors that must be considered in battery selection and power source design is battery charging. Rechargeable batteries are susceptible to over charging. LiPo batteries lose their capacity if over charged. The LiPo needs a protection circuit to protect the battery from over charging and too much current draw. The circuit is required to preserve the integrity of the battery and protect the consumer from fire, injury or death. The protection circuit is an extremely small circuit that is attached to the battery and then to the power source. It simply regulates the battery operation.

The battery will be charged via a computer's Universal Serial Bus (USB) port. The USB port allows for a maximum current draw of 500mA. The protection circuit regulates the current draw to keep it under the maximum 500 mA limit.

#### 3.2.4.3 Voltage Regulation

The voltage that is supplied to the actual product has to be limited to 3.3 volts. The LiPo battery that will be used is rated at 3.7 volts. A voltage regulator is required to bring down the 3.7 volts down to 3.3 volts. Every component in WiRAD operates at 3.3 volts.

## 3.3 Software Design

### 3.3.1 Development

Deciding how to conduct our development process was one of the major obstacles associated with our decision to use an 8051 architecture micro-controller (MCU). All of our previous class and lab interaction with MCUs was limited to the PIC MCU by Microchip. The support structure for the PIC included free use of a C compiler and multiple programmers for the device. The PIC, being the more hobbyist friendly of the MCUs, had a much more open community and tool set. Our decision to use the 8051 placed us firmly in the proprietary and corporate camp, meaning that our software development would require either high-price development suites or questionable open source tools.

The big issue in choosing a development environment involves the compiler header files associated with our specific sub-architecture, the SND1. Because ever MCU is not the same, different aspects of its construction need to be taken into account by the compiler. These aspects include data pointer size, heap location, total program memory, special function register (SFR) locations, and interrupt vectors. How these different properties are communicated to the compiler depend upon the environment being used. Each one has its own proprietary syntax that must be followed if the environment is to understand our sub-architecture. This syntax must be exactly followed if the program is to compile and the MCU run correctly. So, in choosing an environment, we find ourselves unable to switch between different environments without first having to go through the hairy process of porting our code by changing the compiler header file syntax.

#### 3.3.1.1 Small Device C Compiler

The first compiler for the 8051 architecture was the Small Device C Compiler (SDCC). SDCC is a small, command-line, open source compiler maintained and originally written by Sandeep Dutta. While this compiler should work for our purposes, the mere act of setting the environment up and configuring the associated compiler headers for our device seemed a worthless headache when there were other more friendly and focused compilers on the market.

#### 3.3.1.2 Keil

Keil seemed like the most obvious choice for a development environment. The development board we ordered from Atmel included compiler header files and project source code tailored specifically for the Keil environment and compiler. Kiel, unlike the SDCC, was a Windows compatible graphical development suite. These factors helped it to outweigh the SDCC in our minds with its compatibility, ease of use, and possibility for increased productivity. Economic concerns, however, played a very large role in our devision not to use the Kiel environment.

Like most professional and proprietary development software, Keil comes with a very large price consideration, upwards of $2,000. As a small senior design team with a very limited budget, this price seemed to be far and above what we were willing to pay for our development environment.

#### 3.3.1.3 IAR Embedded Workbench

Using the IAR Embedded Workbench for the 8051 architecture was to be our final environment decision. Like Keil, the IAR Embedded Workbench is also a graphical development environment, enjoying all the same advantages over SDCC as Keil. While IAR was unsupported by the software and source code associated with our development board, we were able to find the correct methods to translate the Keil source into something IAR compatible. This meant going line by line through some portions of the code in order to correct syntax errors. This is a process we will need to continue as we use bits and pieces of the code supplied to us by Atmel as a template for our own code.

The IAR environment was no cheaper than Kiel, unfortunately. However, the university was able to supply us with an on-site license that allowed us to use the IAR compiler while on campus. In this way, we were able to avoid having to pay the hefty licensing fees associating with the environment and compiler. We plan to continue using IAR for all of the software design associated with WiRAD.

### 3.3.2 Test Cases

[test cases go here]

## 3.4 References

[references go here]