

Proposal for Team 8 (Smart Phone Control of Advanced Sensor Systems)

Sponsor: Battelle Laboratories – Sensor Systems Group

Group Members: Stephen Hilton, Donghun Ha,
Micah Zastro, Michael Allon, Paul Krutty

February 24, 2012

Executive Summary

Battelle Labs, has asked design team 8 to find and develop a method of controlling and monitoring four sensors using an android based smart-phone. This report describes Team 8's solution to this problem. Team 8 plans to create an android application that runs on the phone, enabling the user to monitor the sensors through a graphical user interface (GUI). The phone will be connected via Universal Serial Bus (USB) to a development board, which is connected to a MiWi Radio Frequency Transmitter that connects to the multiple sensors.

Table of Contents**Pages**

3	Introduction
	Background
5	Fast Diagram
6	Design Specification
7	Conceptual Design
9	Feasibility
10	Design Matrix
11	Proposed Solution
12	Risks and Concerns
13	Project Management
14	Budget
15	References

Introduction

Battelle Laboratories has recently developed sensors that detect and identify various materials using optical spectroscopy. Team 8 was challenged to find and develop a method of controlling and monitoring four sensors using an android based smart-phone. The method also must be self-contained, meaning that the sensors and phone cannot connect to wireless networks that are already in place. The phone must be capable of wireless communication with the sensors, detecting problems, sending commands, monitoring status, and controlling multiple sensors simultaneously. Battelle has challenged team 8 with making this design wireless, portable, and user friendly.

Team 8's solution to this problem is to create an android application that runs on the phone, enabling the user to monitor the sensors through a graphical user interface (GUI). This GUI will show the four sensors on the screen, and the user will be able to select each sensor individually. The GUI will also be able to correctly monitor the status of all four sensors simultaneously. The phone will be connected via Universal Serial Bus (USB) to an external Radio Frequency Transmitter that connects to the multiple sensors. The goal for Team 8 is to get the sensors working at greater range than just near each other: for example, across the room. Team 8 also plans on having everything run on battery power. This will help make the design portable and user friendly.

Background

Nowadays, when you are in a bus or on a train you see many people using handheld Smartphones. Smartphones have many strengths; they can consolidate communication and computing functions, directly access the internet using Wi-Fi, and make use of elegant user interfaces. Our society has drastically transformed with the widespread use of the personal computer and the internet. Right now it can be further transformed with the widespread use of smartphones and smartphone development.

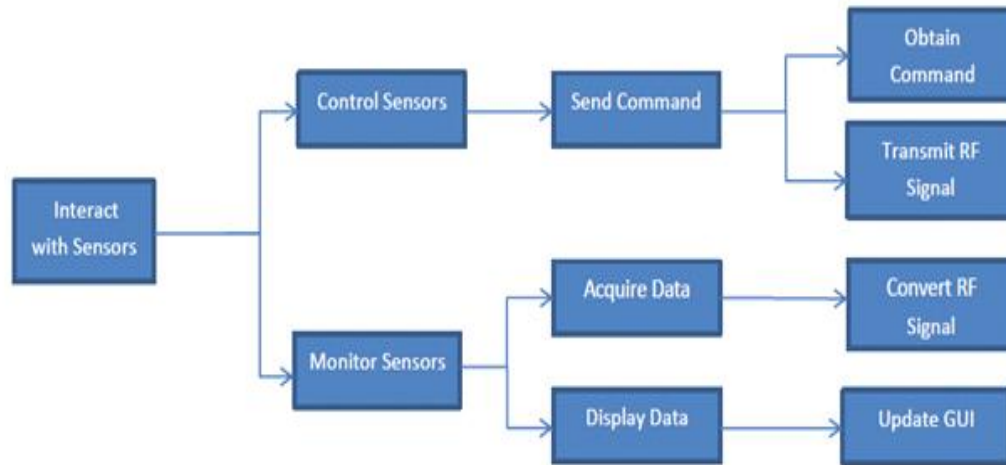
In Engineering, a sensor is a device that produces a measurable response to a change in a physical condition, such as temperature or thermal conductivity, or to a change in chemical concentration. Right now, sensors are generally controlled and monitored by dedicated hardware, which can be expensive and difficult to use. Our team aims to bring the power and ease of smartphone technology to integrate easily with sensors. The goal of the project is to design a smartphone application that can control a Resource Effective Bio-Identification Sensor made by Battelle. We hope that similar technology might one day be used in many applications including the detection of hazardous chemicals on the battlefield or in the event of a terrorist attack. In such extreme scenarios our sensors would accumulate and analyze data, send the information to the smart phone, and warn the user of hazardous conditions, possibly by ordering an evacuation.

This project is a continuation from the fall 2012 team. We aim to fix problems with their design and add new functionality. The previous team attempted to establish communication between an Android node and one Sensor node via RF transmission. They connected the smartphone to a dsPIC33E USB starter kit which then connects to an I/O Expansion board. An MRF49XA transceiver is then connected to the expansion board. This node is supposed to be able to communicate with a remote Sensor node which also consists of a dsPIC33E, an expansion board, and an MRF49XA transceiver via Microchip's MiWi RF protocol. This communication would be controlled by a simple GUI android application. However, the Fall 2012 team was unable to establish a wireless connection between the Android Node and the Sensor node. The main reason for their failure was incompatibility between their hardware.

Our requirements will be to establish wireless communication between and Android node and four sensor nodes. We will build off the previous team's project. We will modify their existing android GUI, and we will continue to use Microchip's MiWi protocol will use the same MRF49XA transceivers. We will, however, be using more compatible hardware to interface with the RF transceivers. We will use the Explorer 16 development board, the USB PICtail+, and the PIC24FJ256GB110 PIM, which will greatly simplify connections between the smartphone, transceivers, and sensors. We also aim to add new functionality. We hope to improve upon things such as range and noise levels. Our communications system will also enable the smartphone to communicate with 4 different sensors. For this we will need to modify the GUI, change settings in MiWi communication code, and account for new problems such as data collisions during communication.

Fast Diagram

Below (Figure 1) is a Function Analysis System Technique (FAST) Diagram which outlines our project. It can be read from left to right or right to left with different interpretations. Moving from left to right shows ‘how’ each task is accomplished. For example, “How do we interact with sensors?” can be answered by “control sensors” and “monitor sensors”. Reading from right to left shows ‘why’ each task is accomplished. For example “Why do we obtain commands?” can be answered by “to Send Commands”.



<Figure 1>

Design Specification

After speaking with our sponsor from Battelle, our team was able to transform the customer's stated needs into real-valued specifications and design parameters. The following specifications are a translation from the voice of the customer to values and constraints that the group can work with. After each parameter, there is a rating from 1(lowest) to 5(highest), which shows how important each is to the overall design requirements of our project.

Communication technology:

- Radio frequency (RF) transmission: The customer has expressed the need to communicate without the use of a pre-existing network which RF transmission does not require (rating: 5)
- Our team will try each of these sub-GHz frequencies to test for range vs. channel noise (rating: 1).
- Transceiver: Microchip's MRF49XA: This transceiver has a range of ~300 meters (rating: 3). It has power-saving modes that will help meet the customer's low-power requirement[1] (rating: 5). We also believe it will integrate well with the other Microchip hardware we are using[6] (rating: 3).
- Transfer Protocol: Microchip's MiWi protocol is "optimized for low-power, low data rate, cost sensitive applications"[5] (rating: 3). MiWi also provides built-in encryption for secure data transfer [5](rating: 1).

Hardware Interface:

- Development board: The Explorer 16 100-pin development board will integrate well with the Android smartphone via the PIC24FJ256GB110 PIM and the USB PICtail+[6]. It will also integrate well with the XRF49XA[6] (rating: 3) It has a 9V power supply outlet so it will not have to be charged with the phone[2] (rating: 5).

Network:

- The team has not yet decided on a network configuration. A star network configuration may be simplest, but a mesh network might provide longer range communication [5] (rating: 2)

Operating System:

- Android OS: The team has not yet determined which version, but we have decided to develop the application on Android OS for its open source software, ease of modularity, and friendly user interface(rating: 5).

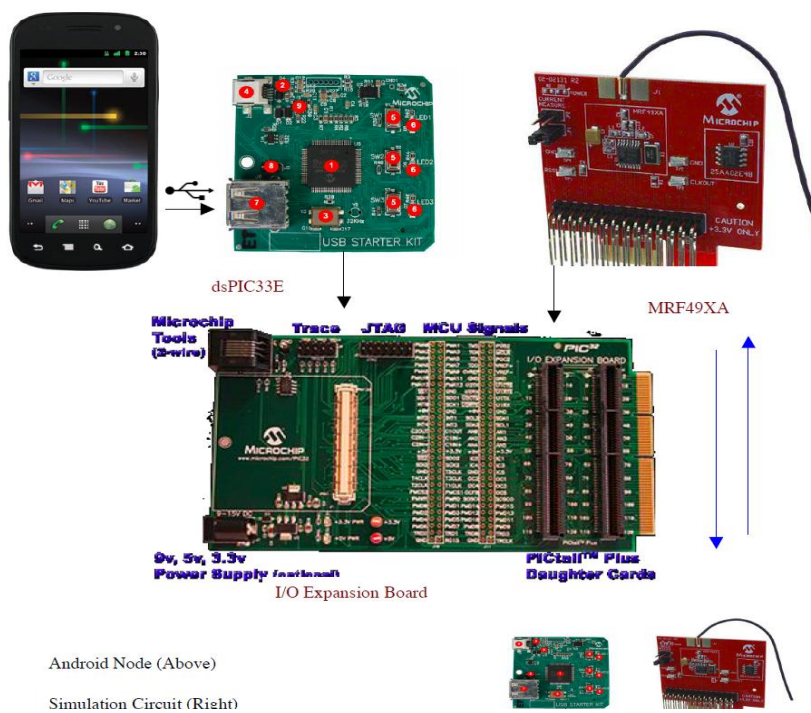
Software Interface:

- GUI: The interface needs to be simple and reliable. We will build off of the previous team's interface but add functionality to support many sensors at once (rating: 5).
- Data Transmission: Microchip's MiWi will be used for high level data transmission. This provides ease of use, modularity and reliability (rating: 2)

Set of Conceptual Designs

Design 1

When deciding on a design solution, two conceptual designs were considered. The team before us created a system using MiWi for wireless communication. Their system (Design 1, Figure 3) included two nodes each composed of the following: the dsPIC33E USB starter kit, an I/O expansion board and a MRF49XA wireless transceiver [1]. These boards were used to create what they called the "Android Node" and the "Sensor Node" (also referred to as the Simulation Circuit).

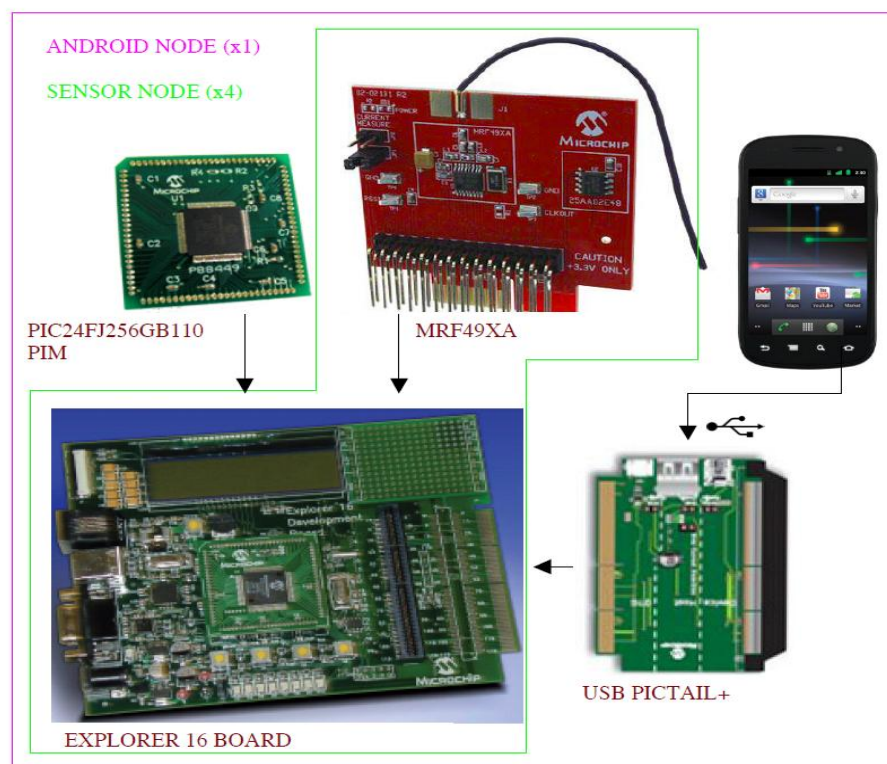


<Figure 2>

The system described above was for a one-to-one communication system; one sensor and one smart phone. To adapt this design to our project we would have 3 additional simulation circuits connected in a star network configuration where the smart phone would act as the PAN Coordinator [5]. However, building upon this design would require establishing the wireless transmission between the two receivers, which hasn't been done yet using these boards. This is due, but may not be limited, to the remapping of Serial Peripheral Inputs needed to connect the MRF49XA to the I/O expansion board which led to some I/O register conflict. The clock frequencies of the MRF49XA and the dsPIC33E were not syncing correctly and would require multiplying the oscillator to obtain the correct value. A system as the one created above has never been done before with these parts and it's still not clear whether or not achieving wireless communication using this system is possible [1].

Design 2

Our team came up with a second concept design (Design 2, Figure 4) that would utilize the MiWi network created by the team before us but included different hardware which are known to have work together. The Android Node in this system is composed of the following: Explorer 16 Development Board, MRF49XA wireless transceiver, USB PICtail+ Board, and the PIC24FJ256GB110 PIM. Each Sensor Node in this system is composed of an Explorer 16 Development Board and an MRF49XA wireless transceiver.



<Figure 3>

As in the first design we will be able to connect with multiple sensors at once using a star network configuration with the Smart Phone as the PAN Coordinator. A mesh network configuration may also be used (where each of the sensors can talk to eachother), this may be favorable since it will increase the total range of the network [5].

Feasibility

Feasibility	Subquestions	Weight	Design 1	Design 2
Operational	Will the solution work?	30%	5	5
	Does the solution meet all requirement?		5	5
	Will users find the solution suitable?		5	5
	Total:		4.5	4.5
Technical	Is the solution practical?	30%	5	5
	Do we have the techology?		5	5
	Do we have the expertise?		3	4
	Total:		3.9	4.2
Schedule	Will parts get ordered in time?	30%	5	5
	Will we complete in time allotted?		5	5
	Ease of testing?		4	4
	Forseen problems?		2	5
	Total:		4.8	5.7
Economical	Cost to develop?	10%	1	1
	Cost for maintenance?		5	5
	Total:		0.6	0.6
Total		100%	13.8	15

<Figure 4>

Above (Figure 5) is the feasibility matrix for our conceptual designs. In terms of Operational Feasibility, both designs use a MiWi network in order to communicate between the smart phone and the sensors. MiWi was chosen before us. It's a good choice for wireless communication because we didn't want to use any existing infrastructures (such as WiFi or the cellular network), since the locations that the sensors will be placed may not be near any of these pre-existing infrastructures.

For Technical Feasibility, Design 1 is slightly less feasible due to the fact that still there hasn't been a successful attempt to create a MiWi network using the parts listed in the description of Design 1. For Design 2, all parts are compatible with eachother. We're using recommended parts for Android communication with the Explorer 16 and the RF transceivers are compatible. Last year, Team 4 programmed and proved that wireless communication between two Explorer 16's is very feasible.

As far as Scheduling Feasibility, by the time this Pre-Proposal is due parts will have been sent to Battelle to be ordered. There are no foreseen problems to get the basic design to work. We're hoping to get it done early enough to be able to make tweaks to get the design to work as well as possible, such as increasing the range of the MiWi network.

Economically, Battelle has offered to provide us with the parts needed to complete the project. The budget will be significantly more than last years since we're going to create a network using 4 sensors and test them simultaneously.

Design Matrix

Design Criteria	Weight
Power	5
Size	1
Range	3
Durability	1
Ease of Use	5
Speed	2
Security	1
Portability	3
Reliability	4

<Figure 5>

Above (Figure 6) is the design criteria we're going to use to evaluate our design. Among the most important criteria are Power and Ease of Use. Power is very important since the design should be able to be used anywhere, this includes places such as battlefield areas where there will be no power outlets. Therefore we are building the design to be battery powered and it is desirable for the battery life to last as long as possible. Ease of Use is a criteria given to us by Battelle. The design should be easy to use, as well as easy to modify for different sensors, since there are a variety of sensors that our design will be able to communicate with.

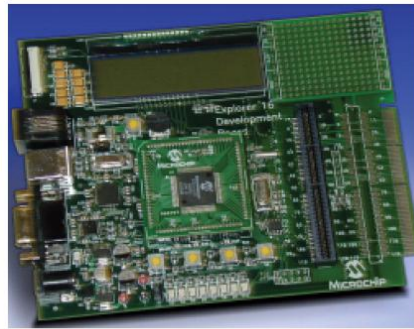
The second most important factor was Reliability. It is important that the network is reliable in that no information is lost. One of the sensors that our network will be used for detects hazardous chemicals, it is important that the phone gets the proper information in order to detect the chemical and alarm appropriately when needed.

Portability and Range are the next important. As far as the sensor nodes go, the circuitry will probably not need to move, however one of the reasons to use a smart-phone and battery power is to facilitate portability of the Android node. Range seems to be fixed in the MiWi network to 300 meters open range, we can increase this, however, by using a mesh network configuration. Speed has a weight of two since we're not sending a lot of data between the links at one time.

Other factors we deemed less important. Size, durability can be adjusted in later designs after we prove that our conceptual design is fully functional. Security is also an issue that we do want to work on to be able to work with sensors that handle sensitive data that, however this isn't an issue that is important to get the design fully functional.

Proposed Design Solution

Hardware – Brief Descriptions

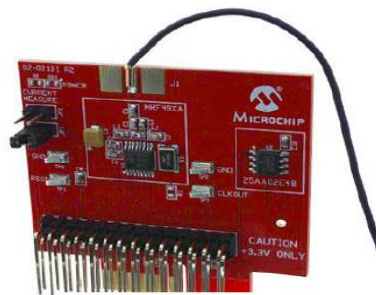


EXPLORER 16 BOARD [2]

<Figure 6>

Quantity Needed: 5

The benefits of the Explorer 16 Development Board (Figure 7) include pre-compiled code written by Microchip to aid in our programming for both the communication between the Android and the Android Node, as well as code to aid the wireless communication between each node. The following parts connect to the board nicely so that no hardwiring is required in the development of the design.

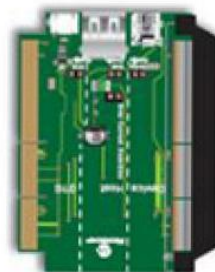


MRF49XA [1]

<Figure 7>

Quantity Needed: 5

The MRF49XA's (Figure 8) are familiar from last year's design. They provide low power consumption capabilities and are already MiWi capable[1].



USB PICTAIL+ [3]



PIC24FJ256GB110 PIM [3]

<Figure 8>

Quantity Needed: 1 of Each

The USB PICTAIL+ and PIC24FJ256GB110 PIM (Figure 9) allow for communication between the Android phone and the Explorer 16. They were recommended to be used in Microchip's Accessory Framework for Android (tm)[6].

Software

One of the advantages of the design path that we will be able to build upon the code written last year. We need to modify configApp.h to select the features we want to enable for the MiWi network as well as some of the code written for wireless transmission between the two Explorer 16s [1]. We'll need to edit and write the wireless software for each of the nodes. Microchip has code that we can use for communication between the Android and the Explorer 16, which will be used and edited for our needs. We will also need to write the GUI for the Android. We will do this building upon the code from last year using the Eclipse IDE[1].

Testing to Complete

There are several tests that we will implement. Range of the product must be tested. We will increase the distance between the Android Node and Sensor Nodes and send data between the nodes in order to test long-range functionality, both indoors and outdoors. Part of our range testing will be noise testing, to make sure that the noise of other equipment doesn't affect our transmissions poorly. We will also test the battery life. Additionally, we will make sure that there is no loss of information or problem with multiple sensors sending data to the smart phone at once.

Risks and Concerns

There are several concerns the team foresees for this project. The three biggest concerns that come to mind are, how to deal with channel noise, how to power all the technology with portable power sources, and lastly, how to protect the transmission channel and develop some sort of encryption.

The primary concern with the radio frequency transmitters is noise on the channel. The proposed solution to this problem is to do extensive testing once the apparatus is assembled. Setting up the system and putting several other technologies in the way such as, radios, computers and various other transmission devices should provide sufficient data to solve the problem.

Secondly, there is a concern with allowing the devices to be portable. The main concern is the phone battery powering the transmission device attached to the phone. To fix this the proposal is to add a secondary battery, something small such as a 9V battery, that can assist in powering the device. This will not take up much space and will save the battery life of the phone.

The last concern has to do with channel security. This concern was deemed low priority because it was not a concern of the customer. Fortunately, the MiWi radio

frequency device already has some built in encryption and this concern should take care of itself.

Project Management

The following (Figure 2) is a Gant chart for our project. It breaks the project down into sub-tasks and shows when the team expects to finish these tasks. The bolded area represents a broad task section and each indented non-bolded row shows more specific tasks within that section. The columns include task name, duration, start and finish dates, predecessors (the row number of the task's dependencies), and names of group members working on the section.

	% Coi		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	100%	✓	Project Definition Task	6 days	Thu 1/19/12	Thu 1/26/12		Micah,Paul,Michael,H
2	100%	✓	Read Report	1 day	Thu 1/19/12	Thu 1/19/12		
3	100%	✓	Meeting with Dr. Ren	1 day	Fri 1/20/12	Fri 1/20/12	2	
4	100%	✓	Meeting with Dr. Ball	1 day	Mon 1/23/12	Mon 1/23/12	3	
5	100%	✓	Contact previous project manager	1 day	Wed 1/25/12	Wed 1/25/12	3	
6	100%	✓	Discuss with team	1 day	Thu 1/26/12	Thu 1/26/12	5	
7	100%	✓	Specification Milestone	0 days	Thu 1/26/12	Thu 1/26/12	6	
8	100%	✓	Develop Conceptual Designs	6 days	Fri 1/27/12	Fri 2/3/12	7	Micah,Paul,Michael,H
9	100%	✓	Determine Specification Priorities	1 day	Fri 1/27/12	Fri 1/27/12		
10	100%	✓	Brainstorm Designs	1 day	Mon 1/30/12	Mon 1/30/12	9	
11	100%	✓	Research Hardware/Software Needed for Each Desig	3 days	Tue 1/31/12	Thu 2/2/12	10	
12	100%	✓	Choose Optimal Design	1 day	Fri 2/3/12	Fri 2/3/12	11	
13	100%	✓	Conceptual Design Milestone	0 days	Fri 2/3/12	Fri 2/3/12	12	
14	55%		Develop Android GUI	23 days	Mon 2/6/12	Wed 3/7/12	13	Micah,Paul
15	100%	✓	Setup software	1 day	Mon 2/6/12	Mon 2/6/12		
16	100%	✓	Develop conceptual design	1 day	Mon 2/6/12	Mon 2/6/12		
17	100%	✓	Acquire phone	5 days	Mon 2/6/12	Fri 2/10/12		
18	100%	✓	Familiarize with previous team's code	4 days	Tue 2/7/12	Fri 2/10/12	15	
19	35%		Write code	14 days	Mon 2/13/12	Thu 3/1/12	18	
20	0%		Test / Debug code	4 days	Fri 3/2/12	Wed 3/7/12	19	
21	0%		Android GUI Milestone	0 days	Wed 3/7/12	Wed 3/7/12	20	
22	38%		Develop Android Node	28 days	Mon 2/6/12	Wed 3/14/12	13	Michael,Hun
23	100%	✓	Order parts	5 days	Mon 2/6/12	Fri 2/10/12		
24	100%	✓	Familiarize with hardware	5 days	Mon 2/13/12	Fri 2/17/12	23	
25	100%	✓	Assemble explorer16 board	1 day	Mon 2/13/12	Mon 2/13/12	23	
26	0%		Write code	14 days	Mon 2/20/12	Thu 3/8/12	24	
27	0%		Test / Debug code	4 days	Fri 3/9/12	Wed 3/14/12	26	
28	0%		Android Node Milestone	0 days	Wed 3/14/12	Wed 3/14/12	27	
29	59%		Develop Sensor Nodes	17 days	Mon 2/6/12	Tue 2/28/12	13	Michael,Hun,Steve
30	100%	✓	Order parts	5 days	Mon 2/6/12	Fri 2/10/12		
31	100%	✓	Familiarize with hardware	5 days	Mon 2/13/12	Fri 2/17/12	30	
32	0%		Write code	5 days	Mon 2/20/12	Fri 2/24/12	31	
33	0%		Test / Debug code	2 days	Mon 2/27/12	Tue 2/28/12	32	
34	0%		Sensor Node Milestone	0 days	Tue 2/28/12	Tue 2/28/12	33	
35	48%		Develop Mi-Wi communication code	10 days	Mon 2/6/12	Fri 2/17/12	13	Steve
36	100%	✓	Download software and libraries from microchip	1 day	Mon 2/6/12	Mon 2/6/12		
37	100%	✓	Familiarize with previous team's code	4 days	Mon 2/6/12	Thu 2/9/12		
38	25%		Write code	7 days	Tue 2/7/12	Wed 2/15/12	36	
39	0%		Test / Debug code	2 days	Thu 2/16/12	Fri 2/17/12	38	
40	0%		Mi-Wi communication Milestone	0 days	Fri 2/17/12	Fri 2/17/12	39	
41	0%		Interface components	13 days	Thu 3/15/12	Mon 4/2/12	28	Micah,Paul,Michael,H
42	0%		Establish communication between Android node and	4 days	Thu 3/15/12	Tue 3/20/12		
43	0%		Establish communication between Android node and	2 days	Wed 3/21/12	Thu 3/22/12	42	
44	0%		Test range, power consumption, noise etc.	5 days	Fri 3/23/12	Thu 3/29/12	43	
45	0%		Create Case	2 days	Fri 3/30/12	Mon 4/2/12	44	
46	0%		Interface components Milestone	0 days	Mon 4/2/12	Mon 4/2/12	45	

<Figure 9>

Feb. 1 st week	Order Parts
Feb. 2 nd week	Start hardware assembly, continue code modification
Feb. 20	Team Oral Proposal Presentation
Mar. 15 th	Finish up modifications & begin testing
Mar. 26 th	Team Technical Lectures
April. 15 th	Team 8 dead line
April. 27 th	Design Day

Budget

-Paid for by Battelle

Item	Quantity	Cost	Total
Explorer 16 Development Board	5	129.99	649.95
MRF49XA RF Transceivers	5	39.99	199.95
USB PICtail+ Board	1	60.00	60.00
PIC 24FJ256G13110 PIM	1	25.00	25.00
Microchip MPLAB ICD3	1	189.99	189.99
Total:			1124.89

-Paid for by ECE budget (\$500)

-Incidentals/Unforeseen (\$170)

<u>Item</u>	<u>Quantity</u>	<u>Cost</u>	<u>Total</u>
<u>Samsung Nexus S Smartphone</u>	<u>1</u>	<u>329.99</u>	<u>329.99</u>

References

- [1] Fall 2011 Final Report
- [2] http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en024858
- [3] <http://www.microchipdirect.com/ProductSearch.aspx?keywords=AC164131>
- [4] Explorer 16 Development Board User Guide 51589a
- [5] Microchip MiWi Wireless Networking Protocol Stack AN1066
- [6] Microchip's Accessory Framework for Android (tm)