

BOEING NFC PROJECT

By

James Kim

Neil Misak

Shao-Chi Ou Yang

Final Report for ECE 445, Senior Design, Spring 2013

TA: Kevin Bassett

1 May 2013

Project No. 43

Abstract

Our Senior Design Project will provide a live demonstration on how NFC technology can be used to improve Boeing's current part tracking and manufacturing process. An example part will be embedded with an NFC tag upon imaginary arrival to Boeing's facilities. The part will then undergo a series of case studies (explained below) that utilize NFC's capability to write and read from this tag and consistently send data to a server, thus monitoring all activity for the part. Finally, our "smart" jig will aid a worker and automate the quality assurance check for a manual drilling case study. There are two teams that worked on this project; we are Team 2 and will focus on the Smartbench and SmartJig cases (3 and 4 respectively).

Contents

1. Introduction	1
1.1 Statement of Purpose	1
1.2 Boeing’s Current Situation	1
1.3 Benefits of NFC.....	1
1.4 Case Overview.....	2
2 Design and Tests	4
2.1 SmartBench Overview.....	4
2.1.1 SmartBench Block Diagram.....	5
2.1.2 SmartBench Schematic	5
2.1.3 Block Description-Adafruit PN532 NFC Shield	5
2.1.4 Block Description-13.56 MHz Antennas	6
2.1.5 SmartBench Software Flowchart	9
2.1.6 Block Description-Arduino Uno Board	9
2.1.7 Block Description-Arduino WiFi shield.....	13
2.2 SmartJig Overview.....	15
2.2.1 SmartJig Block Diagram.....	16
2.2.2 SmartJig Schematic	16
2.2.4 Drill and Jig Verification Flowchart	16
2.2.5 Block Description-Force Sensor	17
2.2.6 Block Description: Freaks HC-SR04 Ultrasonic Range Finder:.....	18
2.2.7 Block Description: Light Detection Sensor	19
2.2.8 SmartJig results	20
3. Costs.....	21
3.1 Parts	21
3.2 Labor	21
4. Conclusion.....	22
4.1 Accomplishments.....	22
4.2 Failed Verifications.....	22
4.2 Uncertainties.....	22

4.3 Ethical considerations	22
4.4 Future work.....	23
References	25
Appendix A Requirement and Verification Table.....	27
Arduino Uno.....	27
TRF7970A NFC Module	28
Ultrasonic Range Finder: HC-SR04	30
Adafruit NFC Shield (For Arduino)	31
External Antenna Board.....	33
FSR Model 408 Force Sensor.....	34
Arduino Wi-Fi Shield	35
NFC Passive Tags	37
Light Sensor.....	38
4:1 MUX: ADG904-R, SN74HC153D, and HP4067.....	39

1. Introduction

1.1 Statement of Purpose

Near Field Communication (NFC) is a short-range wireless technology that operates at 13.56 MHz. NFC typically requires a distance of 4cm or less to enable data transfer at rates between 106 to 848 kbit/s. The majority of smartphones in today's market are equipped with NFC, which allows for pairing with NFC tags or stickers. Boeing wants to evaluate the technical and financial viability of upgrading its legacy system to one equipped with NFC capability. The focus of our project will be to demonstrate this technology in a series of vignettes.

1.2 Boeing's Current Situation

Boeing is the world's leading aerospace company and the largest manufacturer of military and commercial aircrafts combined. Due to stringent regulations in the aerospace industry, each and every part, tool, and process during assembly must be recorded. This verification process in Boeing's factories still consists of mostly barcode or even paper and pencil methods, which is outdated compared to available technology. After our team's tour of the factory in St. Louis, this was indeed verified. There were bins of parts that had barcodes on the front of the shelf, parts in bags with barcodes on them, and even red ribbons with "Low Stock" on shelves that would manually signify to employees that it was time to order new parts. We believe that NFC can be utilized in a novel system to improve the efficiency and accuracy of assembly and part tracking. Nevertheless, some advanced technology was indeed already being used throughout the factory. RFID is used to constantly monitor high end parts; large tags are embedded on wings and readers are strategically placed at certain workstations. The goal of our project is not to replace these existing advanced systems, but make further improvements through increased use of digital technologies in the manufacturing process.

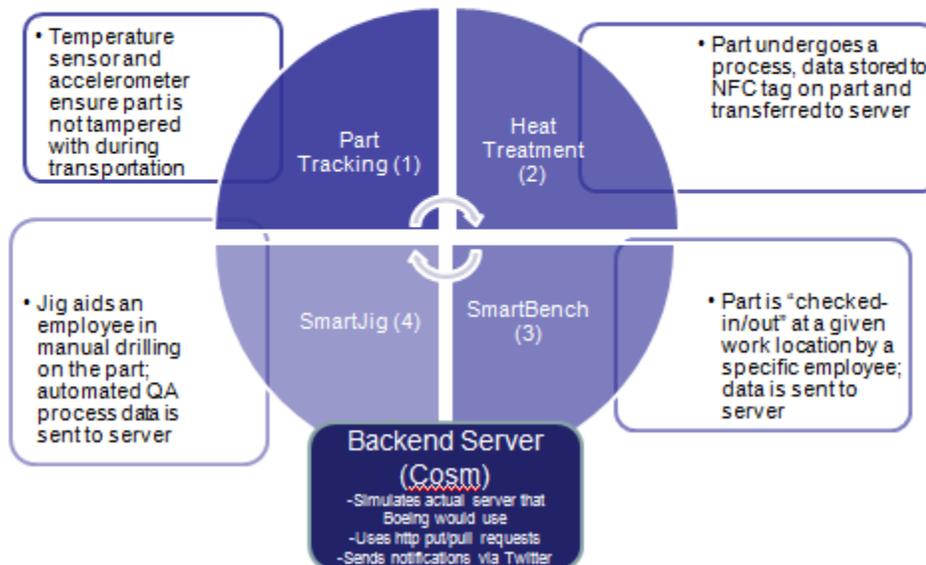
1.3 Benefits of NFC

The first benefit of NFC is that the technology is readily available for use in the vast majority of smartphones in today's market. Smartphones cost a few hundred dollars compared to RFID readers, which cost between \$5,000 - \$10,000. This means that any employee with a smartphone can instantly become connected to a digital part tracking system for much cheaper than ever before. Smartphones also have more processing power and more capabilities than RFID readers, which can be utilized in a variety of ways in the manufacturing process as well. Despite the fact that smartphones are currently not allowed within Boeing's factories, we are proceeding with this project assuming that every employee has a smartphone available to them at work. Boeing employees also currently have NFC equipped ID badges, which will be used for identification in this project.

1.4 Case Overview

The diagram below shows the process that a part undergoes once it arrives to our simulated factory. The four case studies are explained in further detail after the diagram.

Flow Chart for Four Vignettes



Team 1: Case 1- The Industrial Process Tracking case demonstrates a heat treatment scenario. NFC elements that are featured in this case study are the NFC enabled user badge and a NFC tag on the part that is going to be heat treated. To start the process, the employee accesses the NFC unit with his/her badge and then scans the NFC tag on the part to initialize the process. The heat treatment unit assures the successful completion of the process by giving visual aid to the employee in the form of a LCD screen. After the process has been completed, a QA check is written on the NFC tag on the part to indicate the success or the failure of the process.

Team 1: Case 2- For our Part Tracking case study, we employ a sensor bundle that consists of a digital temperature sensor as well as a 2-axis accelerometer which can measure tumbles and falls. The information coming from those sensors are logged on to an SD card, in the case that the information acquired with the sensors are not in the desired range, thus meaning that the part experienced less than ideal storage conditions and could have been tampered with. When the part arrives at a Boeing facility, it is then taken to the NFC unit which an employee would access with his/her NFC enabled user badge, and put a QA check mark on the NFC tag associated with the part to indicate it went through its transportation successfully.

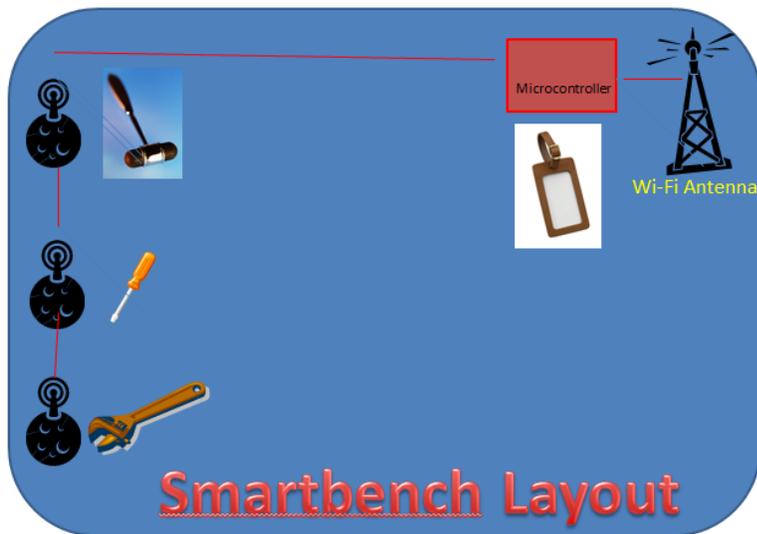
Team 2: Case 3-The FOD Mitigation case features a workbench with embedded 13.56MHz antennas to detect the presence or absence of tools and part. There is also an employee check-in to the workbench, which allows parts to be assigned to each worker. Each tool and part will have a passive NFC tag and

corresponding ID number written on it. The embedded antennas are able to thus check in or out multiple parts, and also link a given part to which employee used it last. A message can be sent to an employee from our server if a part has been removed for an excess amount of time, thus mitigating loss and also ensuring that tools and parts aren't left in unknown places.

Team 2: Case 4-The Quality Assurance case will feature a "smart" jig with contact, light, and depth sensors. These sensors will determine that a hole has been drilled at a particular location, and also how deep the hole has been drilled at that location. This will not only aid the worker but also automate the QA process and save Boeing significant time.

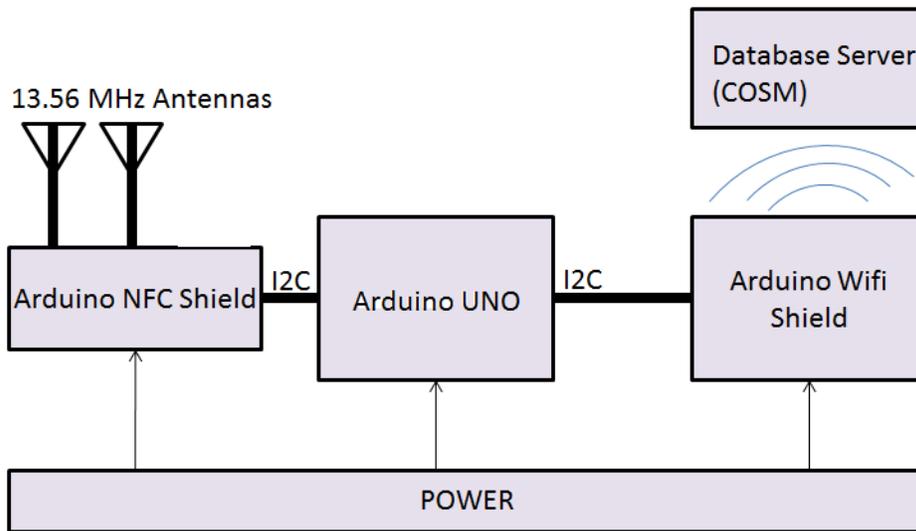
2 Design and Tests

2.1 SmartBench Overview

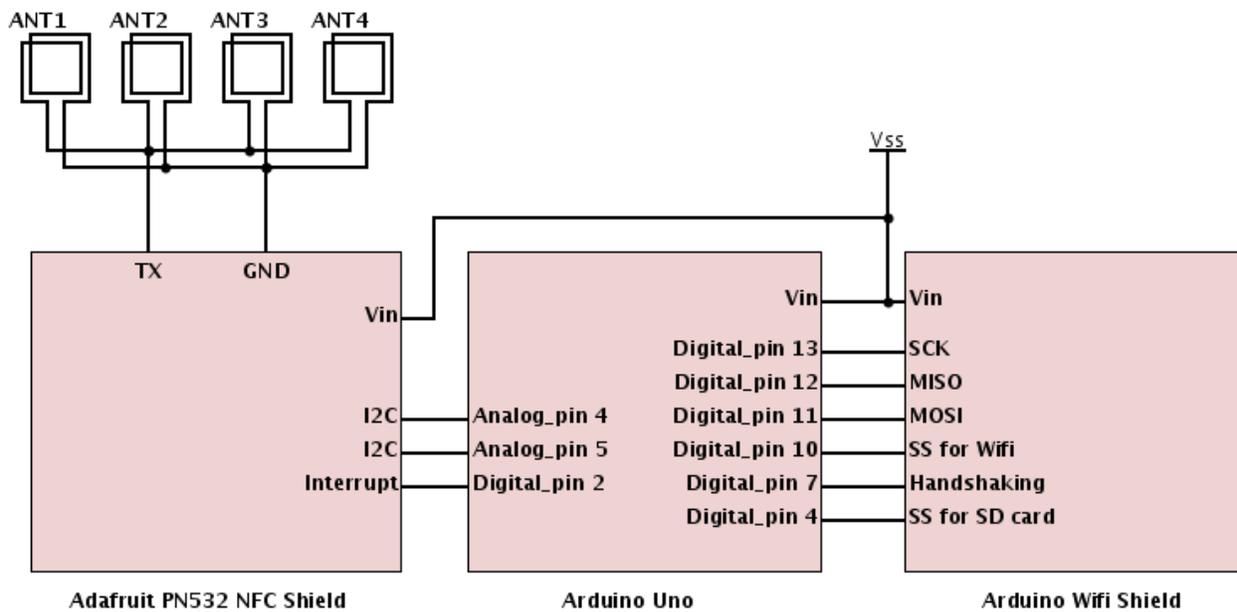


- NFC tags are placed on all tools and parts
- 13.56 MHz antennas read/write to these tags
- Able to determine presence/absence of tools and parts
- Able to track which employee is in possession of tool/part

2.1.1 SmartBench Block Diagram

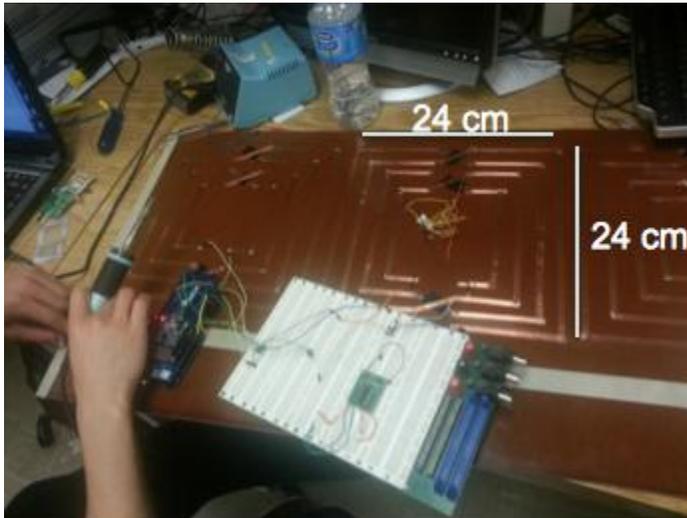


2.1.2 SmartBench Schematic



2.1.3 Block Description-Adafruit PN532 NFC Shield

The NFC Shield uses PN532 chip-set to perform all NFC related functions. The PN532 is based on an 8051 core, with 40 Kbytes of ROM and 1Kbyte of RAM. This powerful chip allows us to both read and write to passive NFC data exchange format (NDEF) tags and cards. The NFC shield comes with a built-in antenna that allows 0-10cm range of detecting NFC tags. We have used the default I2C interface to communicate with our Arduino Uno microcontroller although the NFC shield does provide UART interface. Analog pin 4 and 5 are used to create the I2C connection. Digital pin 2 of the Arduino is used to read interrupt notifications once a tag has been detected on the surface of the antenna board.



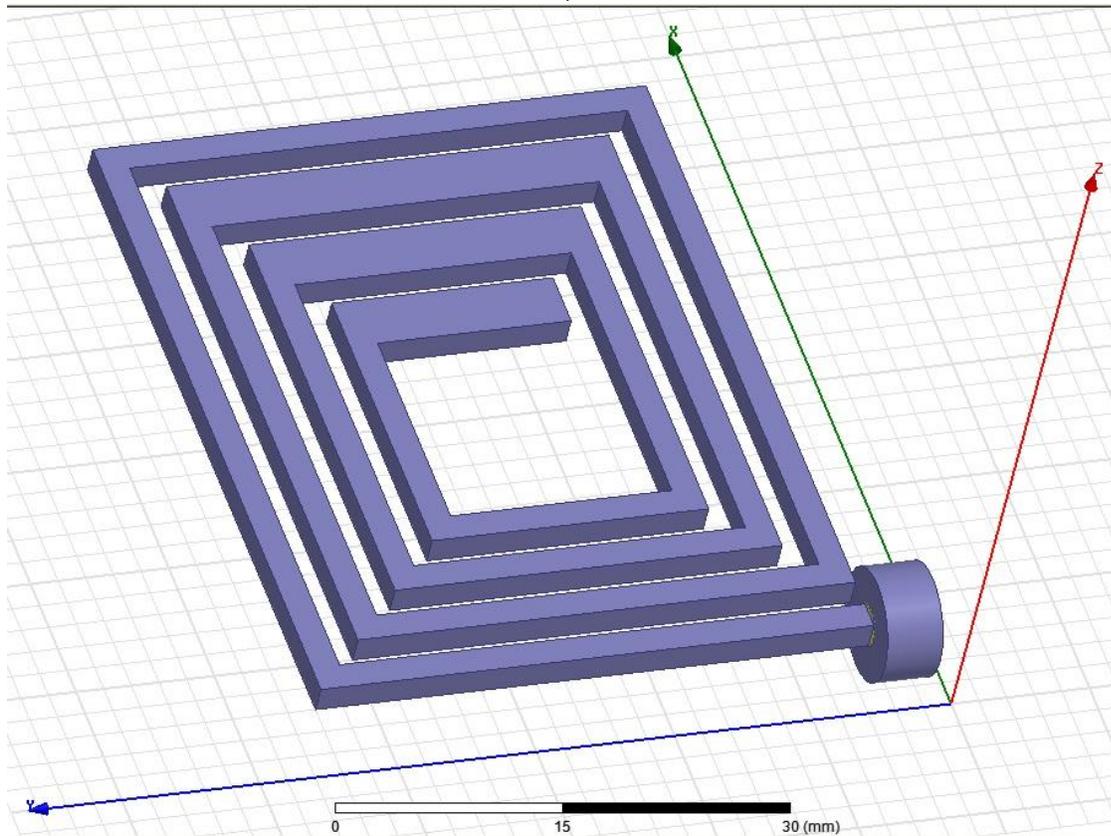
2.1.4 Block Description-13.56 MHz Antennas

Four external antennas are used, which correspond to the four separate tool locations on the workbench. The operating frequency of these antennas is 13.56MHz, which is the operating frequency of the passive tags embedded on each of the three tools. A close up image of our antenna board is shown above. These antennas were created using adhesive copper tape and respective matching networks if needed. The copper tape strips were soldered together to ensure proper connection on top of a 1 ft by 4 ft PCB with a thickness of $\frac{1}{4}$ inch to simulate a workbench. Approximately $\frac{1}{4}$ inch wide strips of copper tape were placed in an inward spiral rectangular shape on a piece of cardboard, with four loops in total. The dimensions of this antenna are approximately 24 cm by 24 cm at the widest points. The wide size of our antenna board provided a bigger hotspot for passive tag detection compared to our initial 4 cm by 5 cm design.

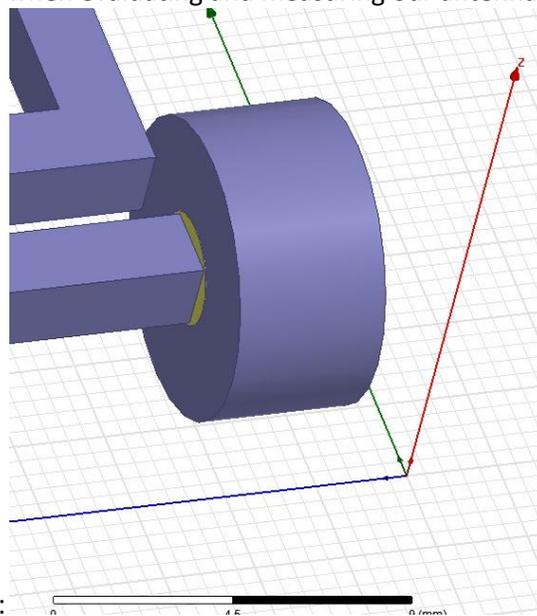
Antenna Simulation

After meeting with Professor Schutt-Aine and several of his respective graduate students, we decided that the best software to use in simulating our antenna was HFSS (High Frequency Structural Simulator), which is the industry-standard simulation tool for 3-D full-wave electromagnetic field simulation. We

constructed a simulated version of our antenna, which is seen below:



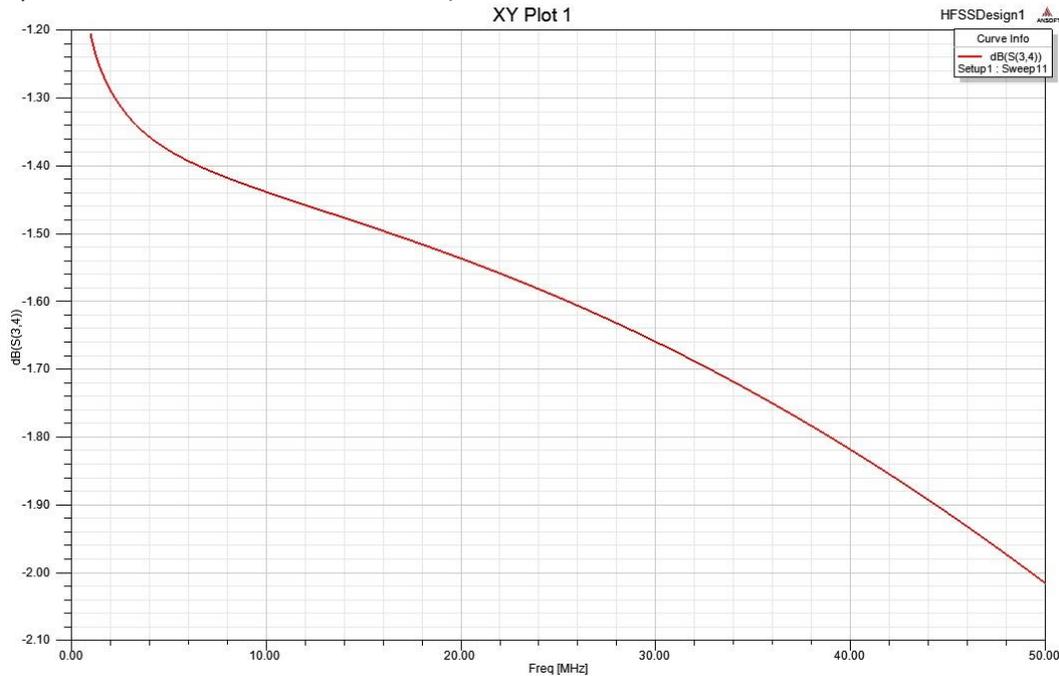
As seen above, the exact dimensions of the antenna are easily measured and adjusted using this software. This allows for optimization of our antenna design to ensure maximum power transfer and minimum return loss (S_{11}). The connector at the end of the antenna simulates our coax connection to the Vector Network Analyzer when evaluating and measuring our antenna. This coax connection is



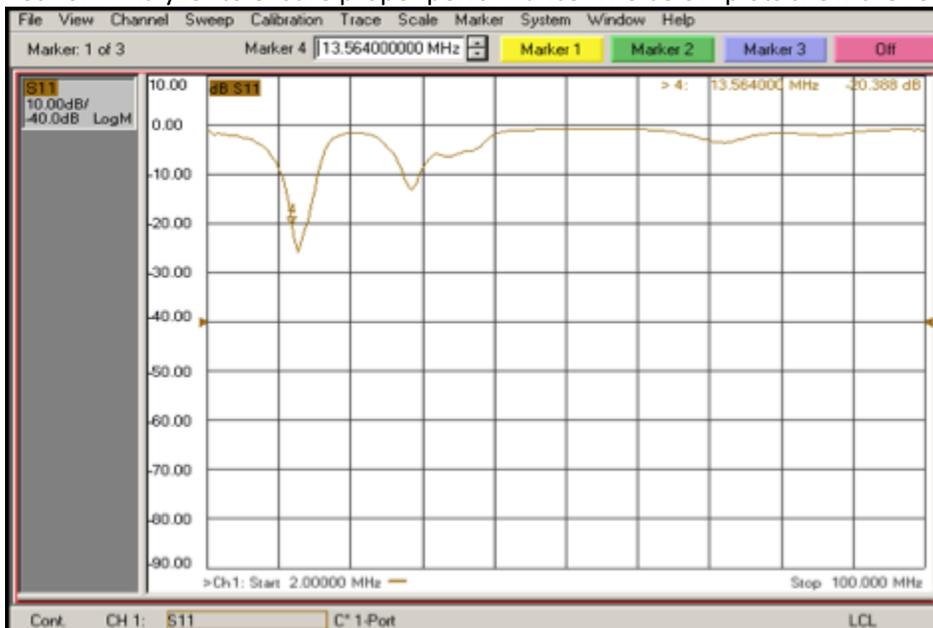
shown in better detail below:

Once our antenna was accurately created in HFSS, we can then run different simulation tests on it. The primary test of concern is calculating the return loss across our desired frequency range. An output of

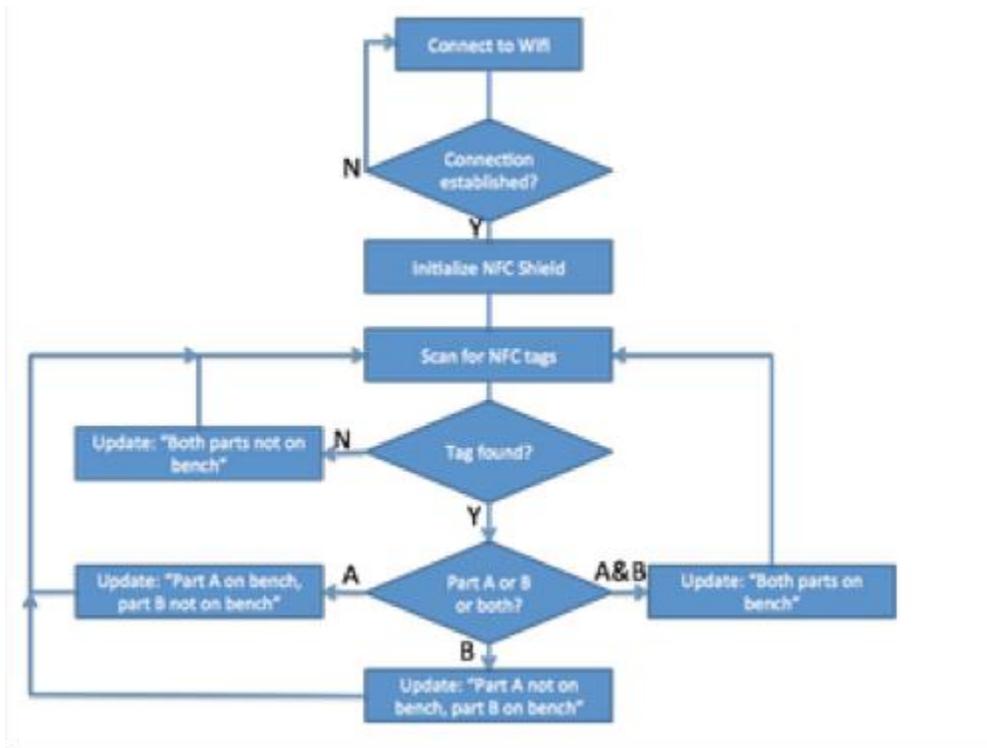
our simulation is shown below (Note: Our antenna is not optimized yet and thus the graph is not a true representation of our ideal return loss)



Instead of creating a matching network for each antenna, we will instead vary the connection point on the antenna trace, which can be easily simulated in HFSS. This works because the impedance of the copper tape varies with length. Once all three antennas are simulated in HFSS, we will then actually create them as stated above using copper tape. These antennas will then be measured using a Vector Network Analyzer to ensure proper performance. The below plots show the return loss of our antennas:



2.1.5 SmartBench Software Flowchart



2.1.6 Block Description-Arduino Uno Board



The Arduino Uno is a microcontroller board. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get powered.

Microcontroller	ATmega328
Operating Voltage	5V

Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Inputs and output:

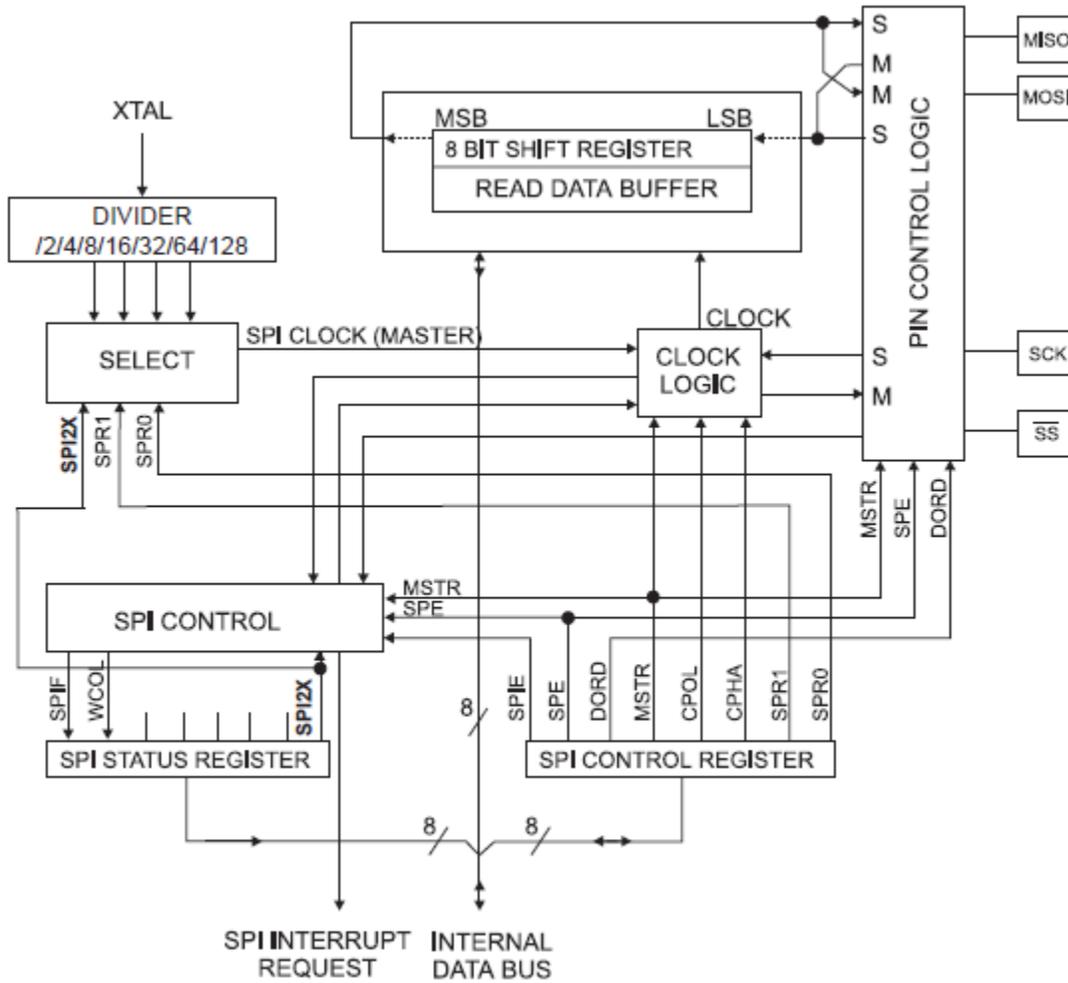
- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the [SPI library](#).
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Connection through SPI:

Block diagram of the SPI on Arduino Board

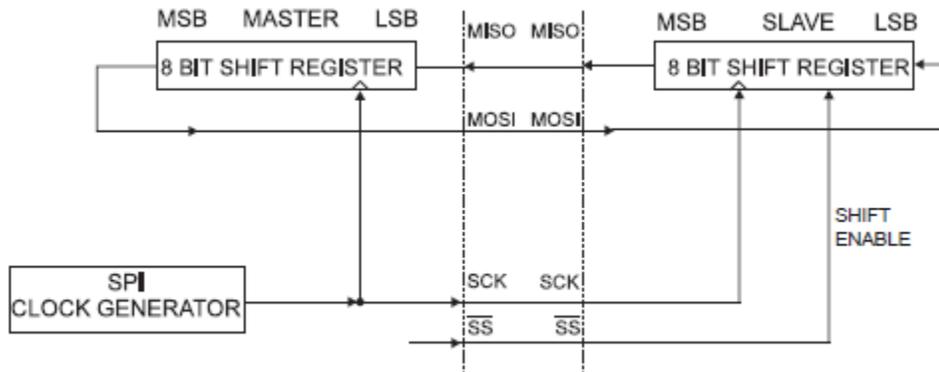


Arduino Uno Data sheet [1]

We will be using the MISO, MOSI, SCK and the SS (WiFi) (Pin 10-13) to connect to our NFC module (TRF7970A).

The SPI gives us 16 bits of register to communicate between the board and the NFC module.

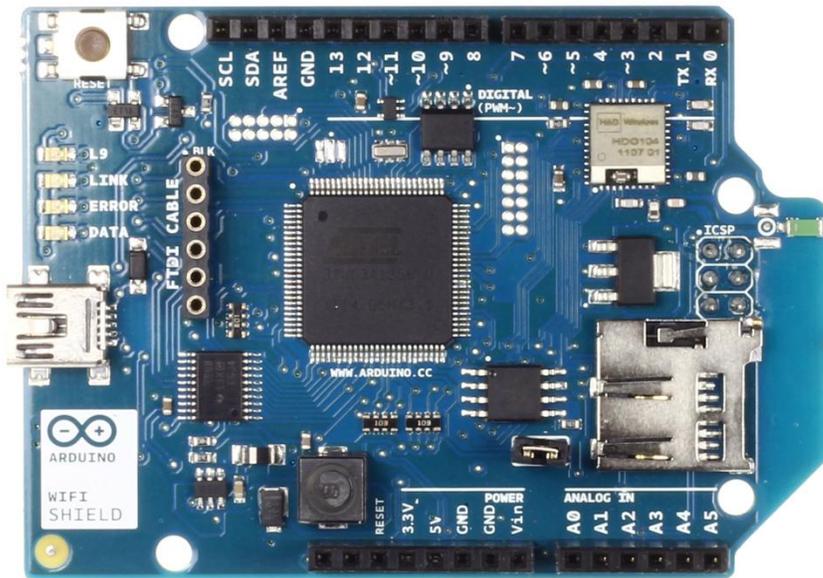
Here is our register master-slave interconnection for SPI:



Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
\overline{SS}	User Defined	Input

↑ Table of Pin override

2.1.7 Block Description-Arduino WiFi shield



The Arduino WiFi Shield connects your Arduino to the internet wirelessly. Connect it to your wireless network by following a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits.

- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Connection via: 802.11b/g networks
- Encryption types: WEP and WPA2 Personal
- Connection with Arduino on SPI port
- on-board micro SD slot
- ICSP headers
- FTDI connection for serial debugging of WiFi shield
- Mini-USB for updating WiFi shield firmware

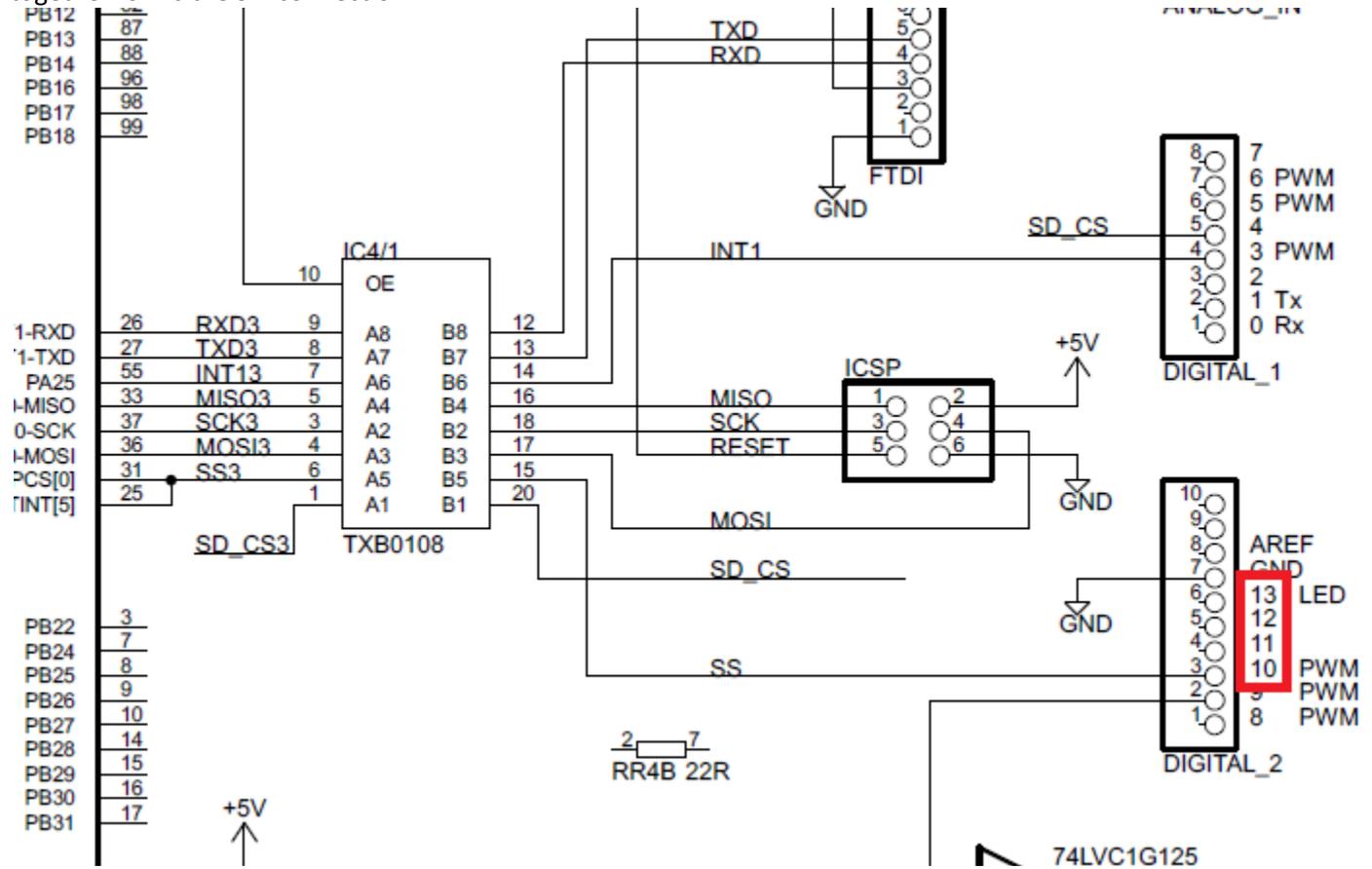
Connection to the Arduino Board:

The Wifi Shield is designed to be an easy plug and play to the Arduino board. Therefore, to attach them we simply attached the pin with the socketed and stack the Wifi shield on top of the Arduino board.

Connection from the Wifi Shield to TRF7970A(NFC modules):

We will be using pin 10-13 to connect to the TRF7970A using an SPI connection.

Below is part of the schematic of the wifi shield, the full schematic can be find in the appendix. Pin 10 is the SS for Wifi, pin 11&12 are the MOSI & MISO protocol, and finally pin 13 is the SCK. These 4 pin together forms the SPI connection.

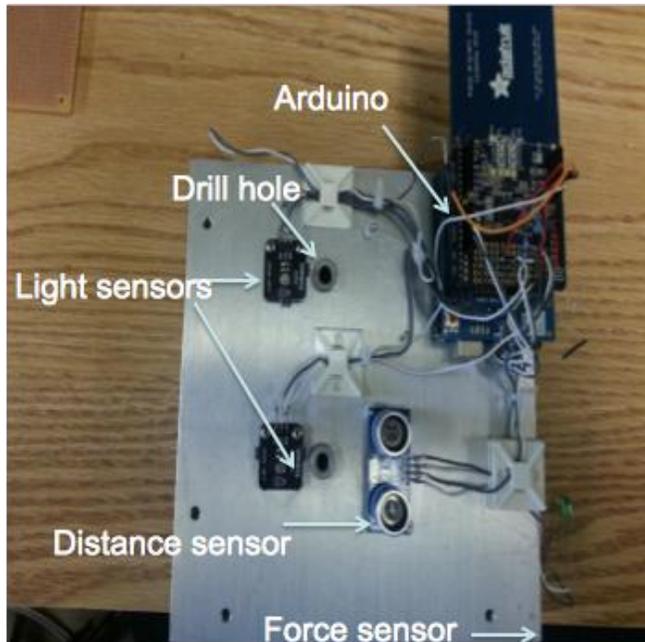


“Arduino Wifi Shield” Schematic received from [4].

Coding of the WiFi Shield and Arduino Board:

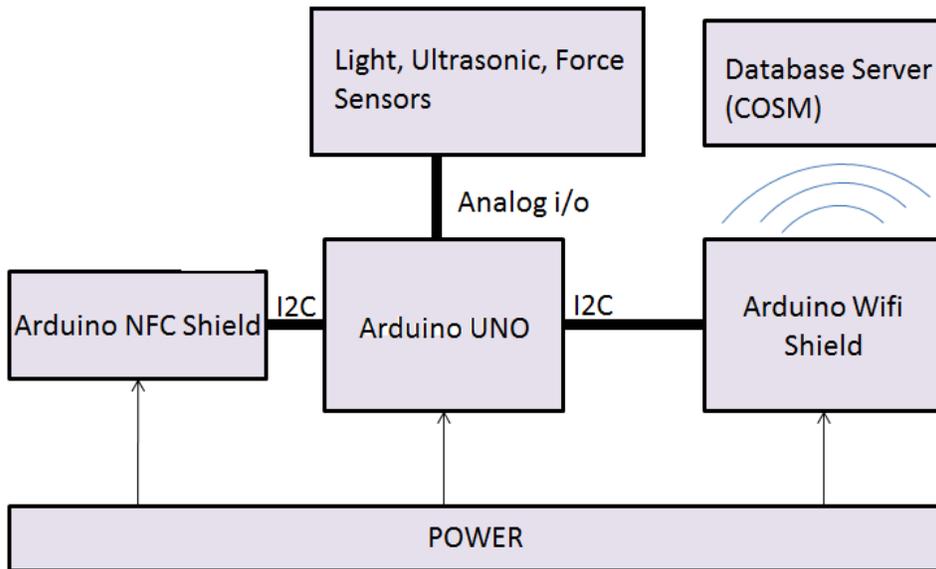
- Arduino WiFi shield comes with a folder of pre-existing library. The library includes the firmware for the board, a client and a server source file.
- The Arduino Board comes with an folder of sources and an existing interface to easily configure the board.

2.2 SmartJig Overview

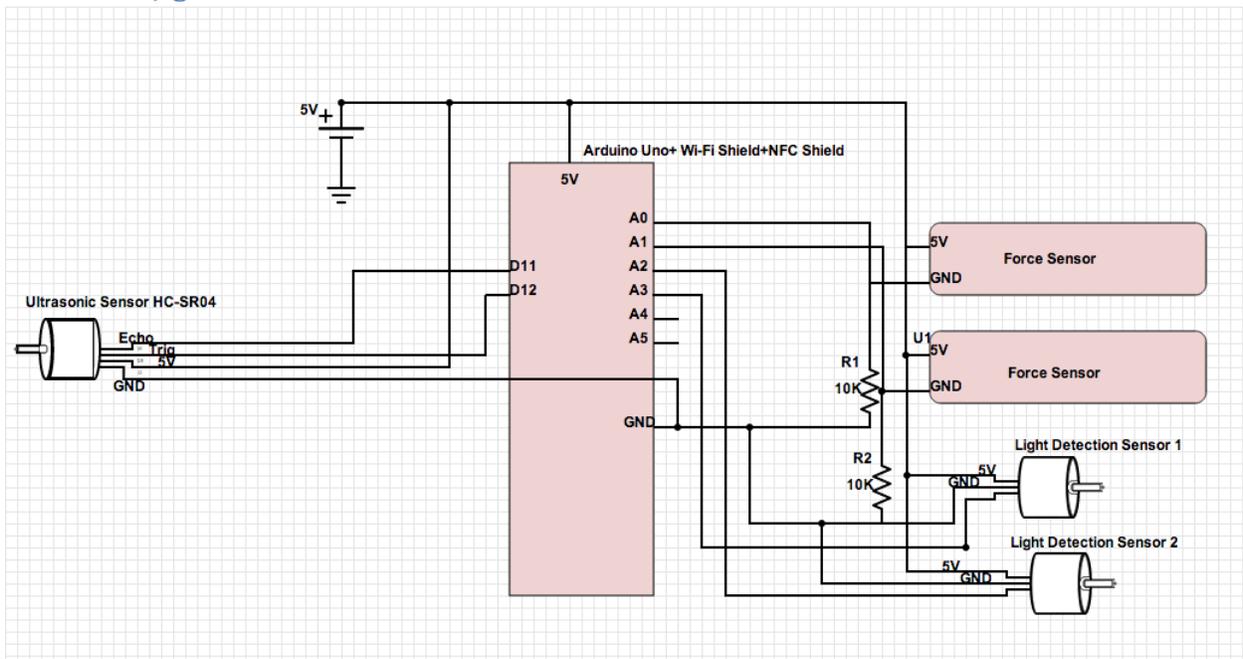


- NFC reader allows employee check-in
- Aid employee in manual drilling and automate QA
- —Force sensors ensure jig is in correct location on the part
- Light sensors on jig determine what hole is being drilled
- Ultrasonic sensors on jig determine the depth of the drilled hole

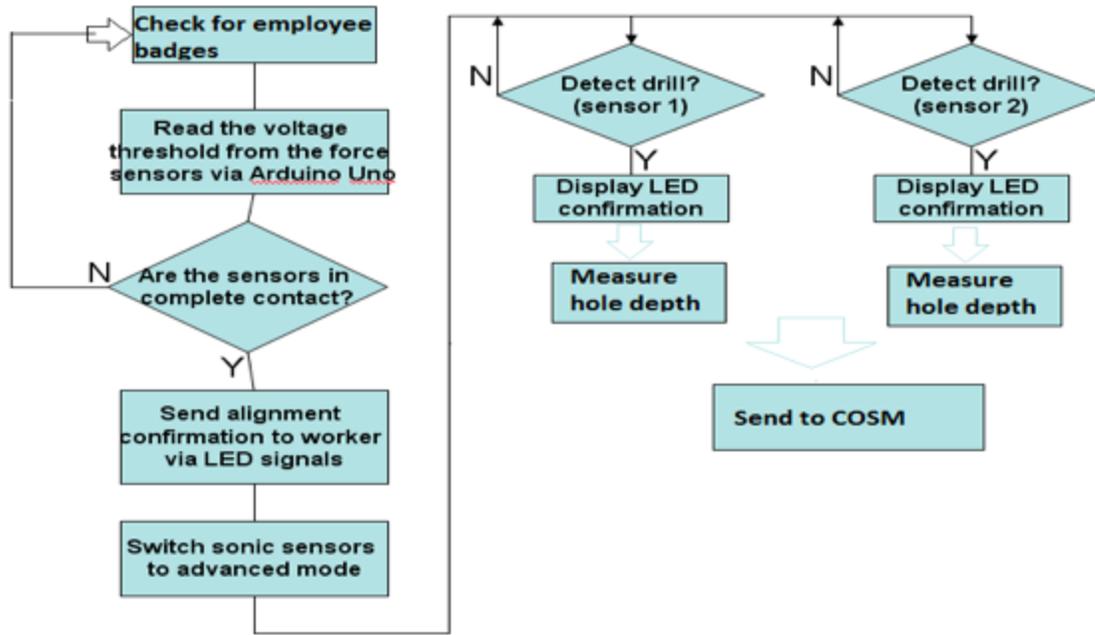
2.2.1 SmartJig Block Diagram



2.2.2 SmartJig Schematic



2.2.4 Drill and Jig Verification Flowchart



2.2.5 Block Description-Force Sensor



Interlink Electronics 24" Long FSR

The FSR force sensor is a robust polymer thick film (PTF) that exhibit a decrease in resistance with increase in force applied to the surface of the sensor.

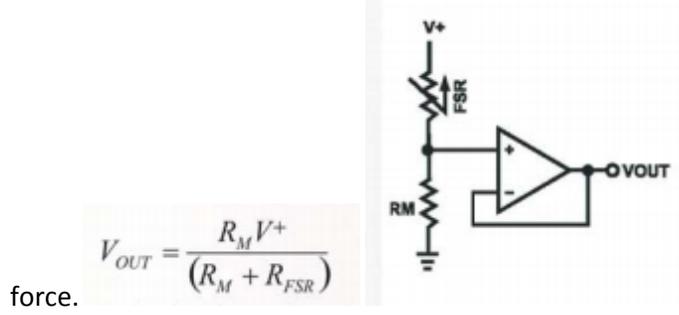
Important Characteristic:

The force sensitivity range goes from ~0.2 N - 20 N.

The sensor's rise time is less than 3 microseconds.

The force precision can change due to the weather, which is included in our table below.

The force act as a resistor to the sensor. Therefore, with a simple voltage divider we can alter the voltage we provided and measure the difference of the voltage output to determine the strength of the



Interlink Electronic FSR 400 datasheet [12]

Operating Temperature Performance	
Cold: -40°C after 1 hour	-5% average resistance change
Hot: +85°C after 1 hour	-15% average resistance change
Hot Humid: +85°C 95RH after 1 hour	+10% average resistance change
Storage Temperature Performance	
Cold: -25°C after 120 hours	-10% average resistance change
Hot: +85°C after 120 hours	-5% average resistance change
Hot Humid: +85°C 95RH after 240 hours	+30% average resistance change

↑Temperature Table

Connection:

We will solder the two pin (Vin and Vout) to the Arduino Uno 5V and IOREF Pin. The data will be collected by the Arduino Uno then send to our Arduino wifi shield for database storage.

2.2.6 Block Description: Freaks HC-SR04 Ultrasonic Range Finder:



The above ultrasonic sensor will indicate the amount of distance between the drill and the sensor to output information about the drilling process. The sensor uses serial I/O interface to connect with our arduino software serial port. The advanced mode of the sensor allows for close proximity sensing from 2cm to 400cm. The sensors will receive commands through the arduino board based upon the following commands:

- Using IO trigger for at least 10us high level signal,
- The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
- Test distance = (high level time×velocity of sound (340M/S) / 2

2.2.7 Block Description: Light Detection Sensor



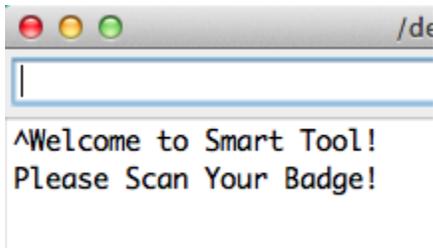
WRobot Arduino Compatible Light Sensor

The light detection sensor is used to verify which hole the worker is drilling and to make sure the worker is drilling at the right angle. There is an infrared laser pointer attached to our drill and it will shine on the light sensor to change the light sensor's reading and then the arduino software will enable the hole to begin drilling.

1. Can detect the surrounding luminance or light then output analog value.
2. Interface : analog.
3. Adjustable Sensitivity , Stable Performance.
4. Size : 23mm*21mm.
5. Working Voltage : +5V.
6. Adopt photoconductive resistance , suitable for detecting a wide range of luminance or light.
7. Applicable to a variety of platforms including Arduino / AVR / ARM

2.2.8 SmartJig results

Below are a series of screenshots taken from the Serial Monitor on the Arduino. This summarizes what would be displayed to the employee working with the SmartJig.



```
Jig is not installed
Jig is not installed
Jig is correctly installed
Please choose a hole!
```

```
Please choose a hole!
Hole 1 is enable
2
Please choose a hole!
Please choose a hole!
```

Result:

```
Hole1 drilled in:
2cm
Over Drilled
```

```
Hole2 drilled in:
773cm
Drill not complete
```

```
Jig is removed
Please Scan Your Badge!
```

3. Costs

3.1 Parts

Part	Quantity	Unit Cost	Part Number	Cost (Note: Prices do not include shipping)
Arduino Uno	1	\$30	R3	\$30
Arduino Wifi Shield	1	\$89.95	N/A	\$89.95
Arduino NFC Shield	1	\$39.99	Adafruit PN532 NFC/RFID Controller Shield	\$39.99
MiFare Classic (13.56 MHz) tag assortment-1kb	10	\$2.50	ADA365	\$25.00
13.56 MHz antenna	1	\$39.95	DLP-RFID-ANT	\$39.95
Interlink Force sensor	2	\$17.95	FSR-408 RB-Int-04	\$35.90
Freaks HC-SR04 Ultrasonic Range Finder	1	\$5.19	HC-SR04 RB-Dev-28	\$5.19
¼ inch * 36 yard Copper Tape	1	\$17.86	JVCC CFL-5CA	\$17.86
Total				\$857.43

3.2 Labor

	Hourly Salary	Hours per Week	Total Weeks	Total Hours	Cost	Total (Cost*2.5)
Neil	\$30	15	10	150	\$4,500	\$11,250
Vincent	\$30	15	10	150	\$4,500	\$11,250
James	\$30	15	10	150	\$4,500	\$11,250
Total	\$90	45	30	450	\$13,500	\$33,750

4. Conclusion

4.1 Accomplishments

We were able to successfully demonstrate our SmartBench and SmartJig during our demo. Although Boeing will not use the exact circuitry or software that we used in their factory, we successfully accomplished our mission of providing our client with a hands-on, working product that shows how NFC technology can be used to improve their current system in place. We anxiously await presenting our product and results to Boeing when they come to Champaign on May 10th.

4.2 Failed Verifications

- Sharp IR Range finder: The accuracy of the range finder varies between +- 5% and does not detect between 0-10 cm.

- TRF7970A: Cannot successfully transfer data to the Arduino (cannot put in serial SPI mode).

13.56 MHz pre-made antenna: Passive tag detection coverage too small for workbench simulation.

- SM130: Firmware update failure, cannot successfully detect tags. We've tried to contact the company's technical service provider, but there was a compatibility issue with the Arduino Uno.

- MUX: tested 3 MUX's-ADG904, 16 pin MUX, 74LS153; Insertion losses due to multiplexers and not enough currents existed to go through multiplexers.

4.2 Uncertainties

Case 1 Smart Bench:

- Tags not read properly due to signal interference
- Backend Database upload error due to COSM website server's delay
- Wifi hotspot signal uncertainties
- Reading more than 2 tags at the same read will cause uncertain read from the software
- Multiplexers compatibility with external antenna
- Better material (other than copper foil tape) for antenna design to increase performance

Case 2 Smart Jig:

- Force sensor is not attached firmly and cause an error on the reading.
- Light sensor might trigger if the jig is place in a very bright room.
- Ultrasonic sensor detect other interference object.
- Ultrasonic sensor collision with the wifi shield's port

4.3 Ethical considerations

We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

1. to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
3. to be honest and realistic in stating claims or estimates based on available data;
4. to reject bribery in all its forms;
5. to improve the understanding of technology; its appropriate application, and potential consequences;
6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
8. to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;
9. to avoid injuring others, their property, reputation, or employment by false or malicious action;
10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Concerns with RFID over privacy and security

1. IEEE Code of Ethics #1
2. Tags are small and almost unnoticeable
3. Little/no security encryption on these tags
4. Employees (not only parts) are constantly monitored

NFC badge check-in allows for increased safety in a factory setting

5. IEEE Code of Ethics #6
6. Employee must check-in before using smart-jig
7. NFC badge stores whether or not employee is qualified to use the jig (or any other equipment)

4.4 Future work

SmartBench: Going Forward

- Better success rate in uploading data and better structured Back-End server for multiple data entry
- NFC library that supports UART anti-collision interface
- Boost power to support more external antennas and tags
- Further optimized antennas for different scenarios

- Battery powered
- LCD

SmartJig: Going Forward

- More accurate distance sensor to increase performance
- Replace the light sensor with a mechanical switch sensor
- Battery powered
- LCD
- Connect to back-end server connection without using the Arduino Wi-fi Shield

References

[1] Arduino Uno, Atmel Corporation, San Jose, CA, 2009. [Online]. Available: <http://www.atmel.com/Images/doc8161.pdf>

[2] Arduino Uno, Arduino, Irving, Italy, 2009. [Online]. Available: <http://arduino.cc/en/Main/ArduinoBoardUno>

[3] Arduino Wifi Shield, Arduino, Irving, Italy, 2009. [Online]. Available: <http://arduino.cc/en/Main/ArduinoWiFiShield>

[4] Arduino Wifi Shield, Arduino, Irving, Italy, 2009. [Online]. Available: <http://arduino.cc/en/uploads/Main/arduino-wifi-shield-schematic.pdf>

[5] DLP-RFID-LP8C, DLP Design. Allen, TX. [Online]. Available: <http://www.dlpdesign.com/dlp-lp8c-ds-v14.pdf>

[6] *IEEE Code of Ethics* [Online]. Available: <http://www.ieee.org/about/corporate/governance/p7-8.html>

[7] PE42440 Product Specification, Peregrine Semiconductor, San Diego, CA. [Online]. Available: <http://www.psemi.com/pdf/datasheets/pe42440ds.pdf>

[8] TRF7970A Datasheet; Multi-protocol Fully Integrated 13.56-MHz RFID/NFC Transceiver IC, Texas Instruments. Dallas, Texas [Online]. Available: <http://www.ti.com/lit/ds/slos743f/slos743f.pdf>

[9] TRF7970A Evaluation Module User Guide, Dallas, Texas. [Online]. Available: <http://www.ti.com/lit/ug/slou321a/slou321a.pdf>

[10] TRF7960A RFID Multiplexer Example System, Texas Instruments. Dallas, Texas [Online]. Available: <http://www.ti.com/lit/an/sloa167/sloa167.pdf>

[11] SRF01 Ultrasonic Range Finder, technical document, <http://www.robot-electronics.co.uk/hm/srf01tech.htm>

[12] FSR 400 Force Sensor, technical document,
<http://www.robotshop.com/content/PDF/datasheet-30-61710.pdf>

<http://www.digikey.com/product-detail/en/ADG904BRUZ/ADG904BRUZ-ND/1007291>

Appendix A Requirement and Verification Table

Arduino Uno

Requirements	Verifications	Status?	Comments
1) Arduino Uno powers on	Connect the Arduino Uno to a PC via USB. The LED labeled "Power" should light up, indicating that the Arduino Board is communicating with the PC.	VERIFIED	
2) Arduino Uno powers on with external battery	Connect the Arduino Uno to an external 9V battery pack. The LED labeled "Power" should light up.	VERIFIED	
3) Able to upload sketches from both PC and MAC to Arduino Uno	Connect Digital Pin 1 to a LED		
	Compile and then upload example sketch "BlinkLED" (provided by Arduino website)		
	Ensure that the LED is blinking	VERIFIED	

TRF7970A NFC Module

Requirements	Verifications	Status?	Comments
TRF7970A NFC Module powers on	1) Supply power through pin outs (5V(+/- .4V) at Pin 2		
	2) Connect the module via USB and see if the Power LED lights up	VERIFIED	
Module is able to detect NFC tags presence	1) Installed GUI provided by Texas Instruments that allowed us to control the various functions of the Module		
	2) Connect module via USB (communicating with PC via UART) and place NFC tags on the antenna board attached to the module		
	3) Run "Seek Tags" function within the GUI		
	4) Verify that a UID appears when the tag is present	VERIFIED	
Module is able to detect multiple NFC tags presence with anti-collision capabilities with less than 1 second delay	1) Repeat requirement #2 except place multiple tags near the antenna		
	2) See if UID's appear for the number of tags that were placed on the antenna, and time the delay	VERIFIED-ABLE TO READ UP TO 5 TAGS CONCURRENTLY	
Module is able to transfer data via SPI		FAILED	Default mode in the Module is Parallel, which is incompatible with the Arduino. SPI Mode on the Module is only enabled through modifications of the breakout board, which

			we were unable to do by hand.
--	--	--	-------------------------------

Ultrasonic Range Finder: HC-SR04

Requirements	Verifications	Status	Comments
1)HC-SR04 operates at 5V	Connect HC-SR04 VCC Pin to 5V and GND Pin to Arduino		
	Connect HC-SR04 Pins Echo and Trigger to Digital Pins on Arduino		
	Modify and then run example code from online source, view serial monitor and see if data is being output	VERIFIED	Source: http://www.electfreaks.com/244.html
2) HC-SR04 can accurately detect distances from 2-30cm with error not greater than 2cm 95% of the time	Run code from requirement #1		
	Convert readings to actual distances using formulas provided from online source		Source: http://www.electfreaks.com/244.html
	Using a tape measure, verify that the measurements from the Ultrasonic Range Finder are accurate to spec		
	Take 100 measurements (incorporating delay into our Arduino Code), make sure that 95% of measurements meet our requirements	VERIFIED	

Adafruit NFC Shield (For Arduino)

Requirements	Verifications	Status?	Comments
1) NFC Shield is able to connect to Arduino	Connect the Arduino Uno to a PC via USB. Solder stacking headers on the NFC Shield as demonstrated through online tutorial. Connect the Arduino Uno to the NFC Shield via these stacking headers. The LED labeled "Power" should light up, indicating that the Arduino Board is communicating with the NFC Shield	VERIFIED	Source: http://www.adafruit.com/products/789
2) Able to read Mi-Fare tag with on-board antenna (distance up to 4cm, accuracy 75% of the time)	Download NFC Library from Github. Modify and upload code "ReadMiFare" to Arduino stack (Arduino Uno plus NFC Shield). Place Mi-Fare tag on antenna board extension on NFC Shield. Measure distance that the tag can be from the antenna before it is no longer able to be deciphered.	VERIFIED	
3) Able to read Mi-Fare tag with external antenna (distance up to 2cm, accuracy 50% of the time)	Repeat step two except with TX1 and GND on the NFC Shield connected to signal of homemade antenna	VERIFIED	We have made MANY different antennas; the distance and accuracy varies significantly with the design and material of the antenna. Our requirements were verified with all antennas on our Workbench Board
4) Able to read tag with multiple external antennas	Repeat step two except with TX1 and GND on the NFC	VERIFIED	Our accuracy and distance decreased dramatically with the inclusion of multiple antennas to our design. We

(distance 1cm (since parts will be on board), accuracy 50% of the time)	Shield connected to signal of multiple homemade antennas		experimented with MUXs and amplifiers in attempt to boost our TX1 signal strength, but had no success. Nevertheless, we meet our spec of 1cm distance and 50% read-tag accuracy.
5)Able to read a variety of NFC tags with on-board antenna at 2-4cm:			
a)Able to read Mi-Fare 1K plastic cards	Repeat procedure in requirement #2	VERIFIED	
b)Able to read Mi-Fare 1K stickers	Repeat procedure in requirement #2	VERIFIED	
c) Able to read Tag-it HF-I stickers	Repeat procedure in requirement #2	FAILED	Stickers were not in NDEF format, and thus unintelligible by our reader. Attempts to modify our reader's software was beyond the scope of our project, and thus we stuck to Mi-Fare cards
6) Able to read a variety of NFC tags with multiple external antennas at 1cm:			
a)Able to read Mi-Fare 1K plastic cards	Repeat procedure in requirement #4	VERIFIED	
b)Able to read Mi-Fare 1K stickers	Repeat procedure in requirement #4	FAILED	Since our external antennas are quite large (12" by 12"), inductive coupling was not possible with these small sticker tags. We will use all plastic cards (approx. 2" by 4") for our Demo
c) Able to read Tag-it HF-I stickers	Repeat procedure in requirement #4	FAILED	Stickers were not in NDEF format, and thus unintelligible by our reader. Attempts to modify our reader's software was beyond the scope of our project, and thus we stuck to Mi-Fare cards

External Antenna Board

Requirements	Verifications	Status?	Comments
Able to read at least three tags simultaneously	Run TRF GUI "Find Tags" command, run "ReadMiFare" from Arduino Library	FAILED	No anti-collisioning software; we are only able to read two tags simultaneously
Ensure that the range of our antenna is 1-4 cm at designated "hotspots"	Measure distance between antenna board and tag while data is still able to be read accurately according to the NFC Shield Verifications	VERIFIED	
Able to read a variety of NFC tags	Able to read Mi-Fare 1K plastic cards	VERIFIED	
	Able to read Mi-Fare 1K stickers	FAILED	Size issue as explained in NFC Shield Verifications
	Able to read Tag-it HF-I stickers	FAILED	Same as above

FSR Model 408 Force Sensor

Requirements	Verifications	Status	Comments
1. FSR operates at +5V +/- 1.5V	Connect FSR to Arduino Uno Analog Pins. Run "AnalogRead" to verify that the FSR has been powered.	VERIFIED	Output to Arduino is not actually voltage with the code we used; it is instead a scale from 1-1000. Therefore we can confirm that the output voltage differs based on the fact that this output changed.
2. Digital output at a constant when force is not presented (0-300). Digital output alters when a force is present (300-1000).	Repeat requirement 1 and view Serial Monitor in Arduino. When no force is present, digital output should be constant. When force is applied to the FSR, the digital output should change accordingly.	VERIFIED	Source: http://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr
3. Force Sensor is responsive to jig being "in place" on our part (wooden block). Output difference must be at least 400 between "in place" and not.	Test force sensor on the jig without being placed correctly against the part. Again, test force sensor with the jig in place. Log 50 data values to determine threshold level from in place vs. not.	VERIFIED	Initially failed; placed tape between the force sensor and jig to allow direct contact between the two. Then this requirement was verified.

Arduino Wi-Fi Shield

Requirements	Verifications	Status	Comments
1) Wi-Fi Shield is able to connect to Arduino	Connect the Arduino Uno to a PC via USB. Connect the Arduino Uno to the Wi-Fi Shield via stacking headers. The LED labeled "Power" should light up, indicating that the Arduino Board is communicating with the Wi-Fi Shield.	VERIFIED	Source: http://arduino.cc/en/Main/ArduinoWiFiShield
2) Wi-Fi Shield is able to search for Networks	Modify and upload code "ScanNetworks". Serial monitor should display all networks in the vicinity, as well as the signal strength.	VERIFIED	Source: https://github.com/arduino/wifishield/tree/master/libraries/WiFi/examples
3) Wi-Fi Shield is able to connect to an Open Network.	Set up Open Network via Samsung Galaxy S3 Smartphone. Modify and upload code "ConnectNoEncryption" to the Arduino Board. Serial Monitor will display whether or not connection was made successfully.	VERIFIED	
4) Wi-Fi Shield is able to connect and upload information to database. Must be able to upload both integers and strings to database	Repeat requirements 2 and 3. Create database in Cosm. Install Cosm library in Arduino. Modify and upload code "WiFiMultipleDatastreamUpload". View Cosm webpage; ensure that data transferred over WiFi Shield is accurate to spec.	VERIFIED	Source: https://github.com/cosm/cosm-arduino/tree/master/examples

with 80% accuracy and within 5 minutes.			
-----------------------------------------	--	--	--

NFC Passive Tags

Requirements	Verifications	Status	Comments
1. Must be able to communicate with an NFC reader at a range of 0-4 cm.	Write to NFC tags with NFC Shield stack on Arduino. Read values on NFC tags with NFC Shield. Measure the maximum distance at which this read/write operation can be performed.	VERIFIED	
2. Data is stored accurately for over 10 days.	Write to tags; read 10 days later.	VERIFIED	

Light Sensor

Requirements	Verifications	Status	Comments
1. Light sensor operates at +5V +/- 1.5V	Connect FSR to Arduino Uno Analog Pins. Run "AnalogRead" to verify that the Light Sensor has been powered.	VERIFIED	
2. Digital output at a constant when light is not presented (0-300). Digital output alters when a light is present (300-1000).	Repeat requirement 1 and view Serial Monitor in Arduino. When no light is present, digital output should be constant. When light is shone from the laser pointer on the sensor, the digital output should change accordingly.	VERIFIED	Output to Arduino is not actually voltage with the code we used; it is instead a scale from 1-1000. Therefore we can confirm that the output voltage differs based on the fact that this output changed.

4:1 MUX: ADG904-R, SN74HC153D, and HP4067

Requirements	Verifications	Status	Comments
1. Ensure that the MUX operates correctly 95% of the time. Accuracy is defined by the output (RFC) of the multiplexer corresponding to the correct input (RF1, RF2, RF3, RF4) based on the truth table stated in the respective datasheets of the MUXs	1. Connect a function generator to RF1, RF2, RF3 and RF4. We will toggle the address bits (EN', A0, and A1, connected to Power or Ground pins) and ensure that the corresponding output RFC is the correct one for each truth table scenario. We will measure the output of RFC using a Multimeter.	VERIFIED	All three MUXs we researched worked correctly according to the datasheet when we used 5V as the input.
2. Verify that the MUX works correctly when transmitting signals from a 13.56MHz antenna.	Connect RF1, 2, etc to inputs of the MUX. Verify that the output is high when a tag is present	FAILED	All three MUXs we evaluated were unable to RF signal. We measured the output voltage of both premade and homemade antennas with a tag present and the Voltage was in the mV range, and thus was not high enough to pass through the MUX.