18-549 Final Write Up

Team 20: Low Power Rangers

Clayton Ritcher (critcher)
Nora Shoemaker (nshoemak)
Shashwat Srivastava (shashwas)

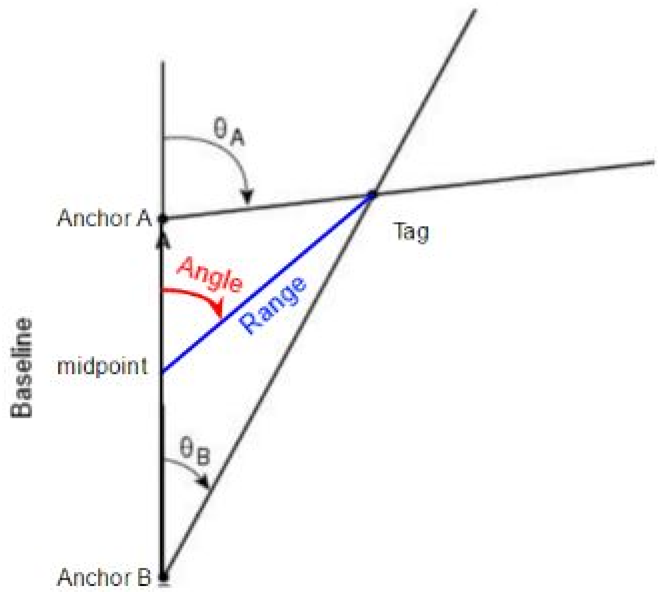
5/7/2016

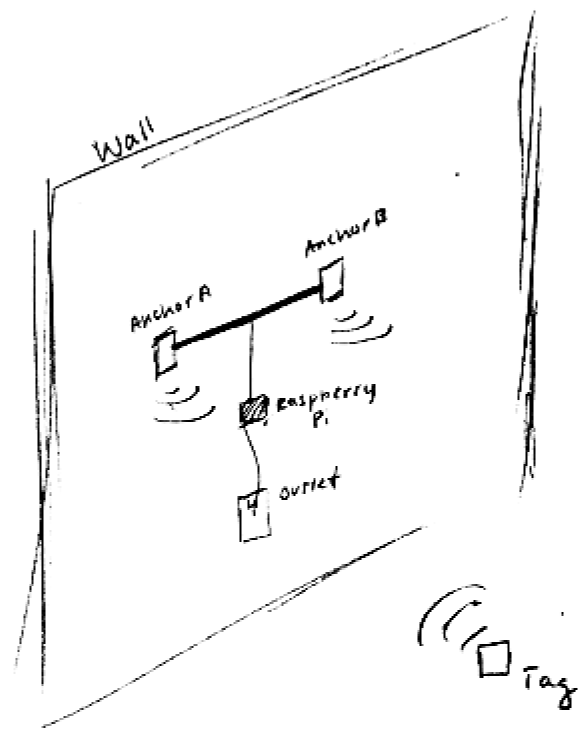
Table of Contents

Project Description

The main goal of this project is to use two Decawave DWM1000 ultra wide band (UWB) radio transceivers to collect angle of arrival and range information of a single other transceiver, relative to the measurement unit. We will call the two transceivers in the measurement device Anchors, and the other device a Tag. Because angle of arrival detection has never been done before publicly with UWB, we want to explore the space and experiment with different configurations in order to evaluate what kind of accuracy we can achieve at various baseline distances between the Anchors.

On top of this, we would like to expose a web API for accessing the angle of arrival and range data collected. Applications that would like to use these data can then be built on top of our API.





Design Requirements

Motivations:

M1. Location-focused client applications want near real-time range and angle tracking of users.
M2. Client applications want information to be easily accessible and platform independent.
M3. Useful client applications will necessitate that the tracked object be mobile and always on.
M4. We want to keep the scope of our project reasonable -- if a feature does not supplement the demo, it is likely unnecessary.

Primary Requirements:

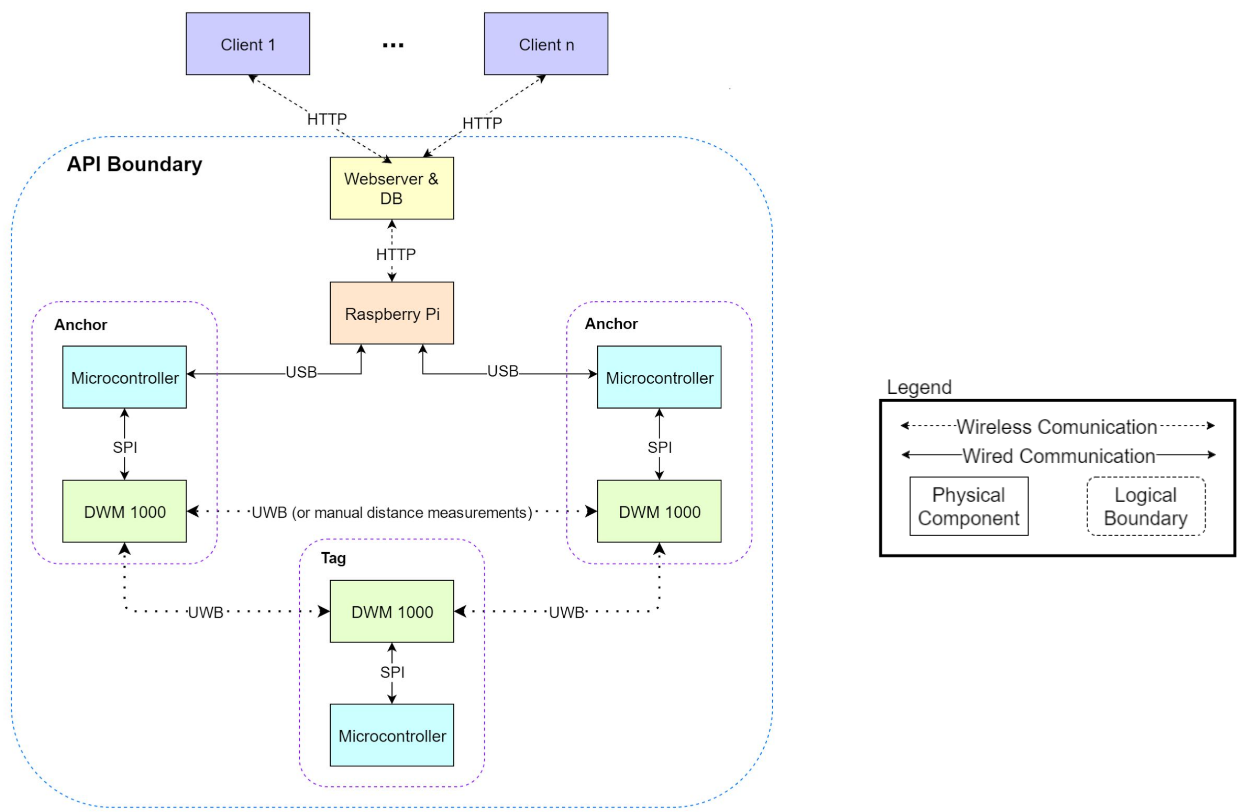
PR1. Range and angle data shall be available through a web-accessible API [M2].
PR2. Range and angle data should be available through the API in under two seconds after the data is collected [M1].
PR3. Range and angle data shall be collected at a rate of at least 1 Hz [M1].
PR4. Tag shall be portable (battery powered) as to allow tracking of mobile objects [M3].
PR5. Tag shall be able to run at least two hours on a full battery [M3].
PR6. Webserver shall be able to handle load of up to 100 requests per second [M2,M4].

Stretch Goals (possible extensions, in no particular order):

SG1. A 'radar' app for finding your keys, phone, etc (anything with a Tag attached).
SG2. A client app that integrates the system with [IFTTT](), allowing actions such as turning on the TV when you sit on the couch.
SG3. Add a third Anchor, making an 'L' with the other two, in order to measure both the vertical and horizontal angle of arrival.
SG4. Ability to change the baseline distance between the measurement Anchors without manually changing the configuration parameters in the code.

Architecture

Below is a physical architectural diagram showing the components of our project and the various forms of communication that the components use amongst themselves.



UWB communication is used to obtain the range data between each of the Anchors and the Tag using TOF analysis. Microcontrollers on each Tag/Anchor talk to the DWM1000 chip through SPI. Once each of the Anchors know their distance from the Tag, this information will be sent to the Raspberry Pi, along with the Tag ID. The Raspberry Pi will then compute the angle of arrival based on the two range estimates and the known distance between the Anchors. Once the angle of arrival is calculated, the range between the Tag and the midpoint of the Anchors will be calculated and these two pieces of data will be sent to the web server, with the Tag and measurement device IDs, through an internal API call.

Design Trade Studies

A major trade study we conducted involved the general message protocol that we plan to use for ranging. To fulfill our requirements, our system must be able to calculate the distance from each of the two Anchors to the Tag at a rate of at least 1 Hz. Additionally, we wish to keep the power consumption of the Tag low so that it can be battery powered, and we wish to achieve the highest accuracy that we can. Ideally, the protocol that we use will allow for Tag discovery, as well.

One ranging protocol is Single-sided two-way ranging (SS-TWR). This protocol involves two messages total and allows a propagation time to be calculated by the initiating device based on the time the other device takes to reply, T_{reply} and the round trip time experienced by the initiating device, T_{round}. Then, the propagation time can be calculated as:

$$T_{prop} = \frac{1}{2}\left(T_{round} - T_{reply}\right)$$

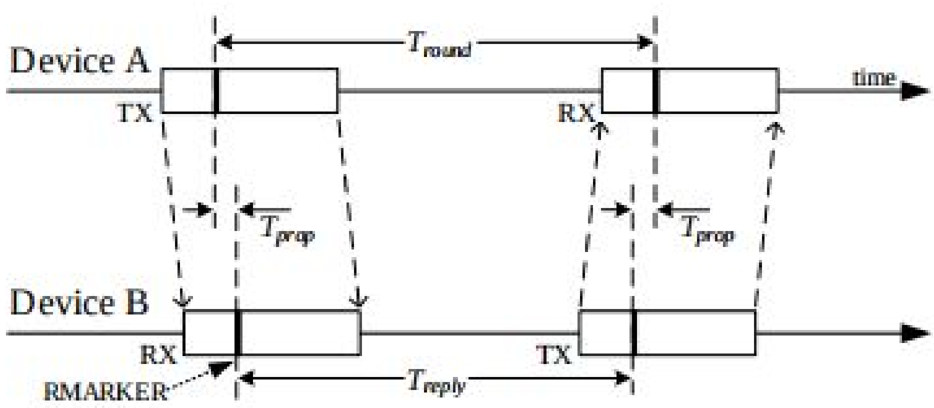


Figure 2. The two-message ranging scheme defined by SS-TWR.[1]

Another ranging protocol we considered is Double-sided two-way ranging (DS-TWR). This protocol uses three messages and varies from SS-TWR in two main ways because of the extra message. First, this protocol can account for error due to clock drift between the devices. Second, this protocol calculates the range on the second device, not the initiator.

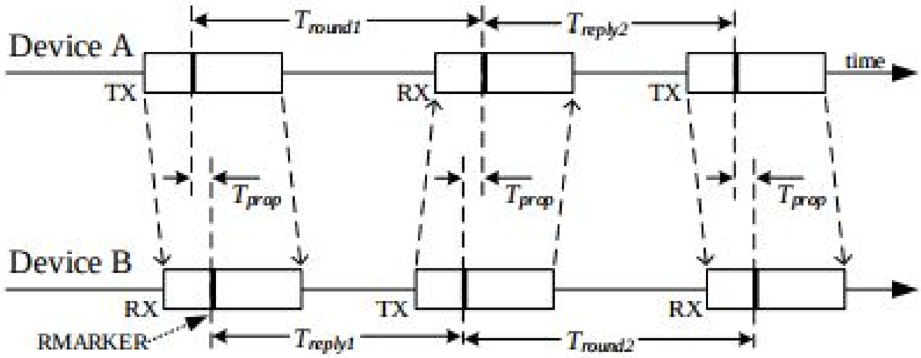


Figure 3. The three-message ranging scheme defined by DS-TWR.[1]

With DS-TWR, the propagation time is then calculated as:

$$T_{prop} = \frac{(T_{round1} * T_{round2} - T_{reply1} * T_{reply2})}{(T_{round1} + T_{round2} + T_{reply1} + T_{reply2})}$$

A final ranging protocol we considered is Broadcast-double-sided ranging (BDSR). This protocol is similar to DS-TWR, but cuts down on the number of messages needed when the initiator is ranging with multiple other devices. This is achieved by having the Tag broadcast its messages and letting all other devices respond in a stagger.

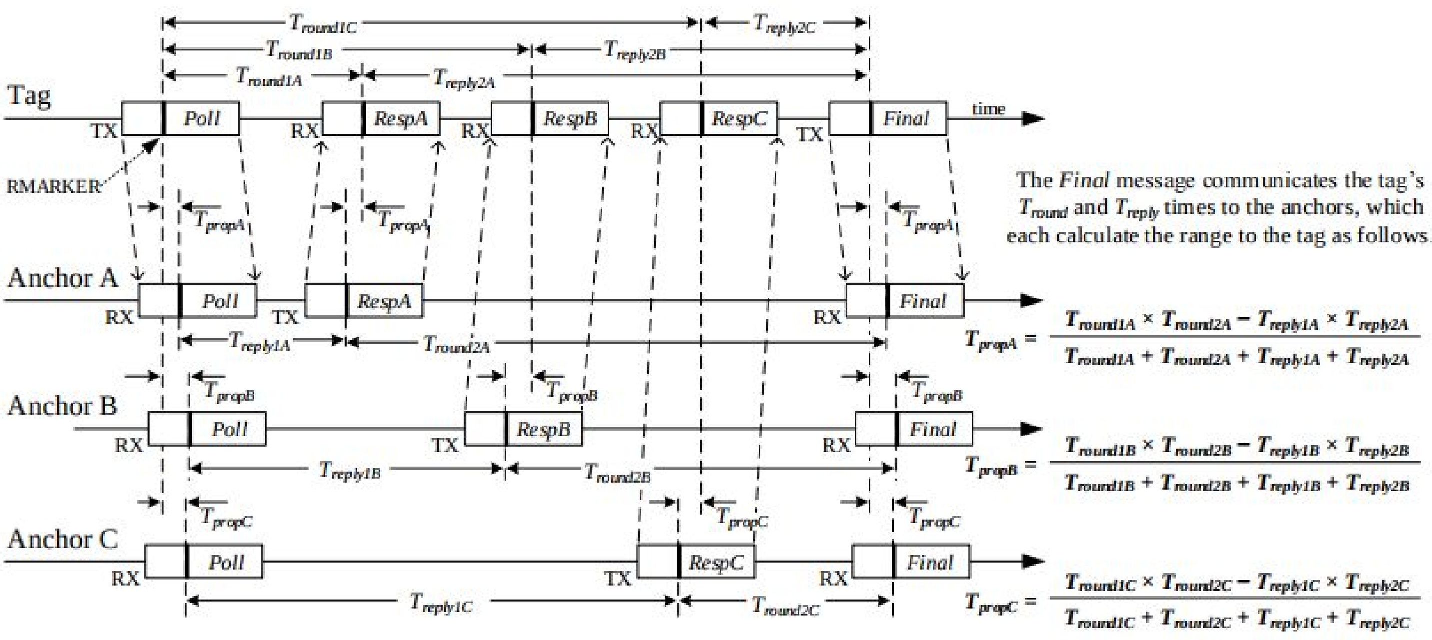


Figure 4. BDSR with three other devices (Anchor A-C). Each Anchor calculates its range from the initiator as shown on the right.[1]

Using the criteria listed above, we decided to evaluate these three protocols against one another:

(All Scores Between 1 and 5)	Weight (higher is more important)	**SS-TWR**	**DS-TWR**	**BDSR**
Possible Update Rate	2	4	1	4
Tag Power Consumption	3	1	3	4
Achievable Accuracy	4	1	4	4
Tag Discovery	1	0	0	5
Total (higher is better)		**15**	**27**	**41**

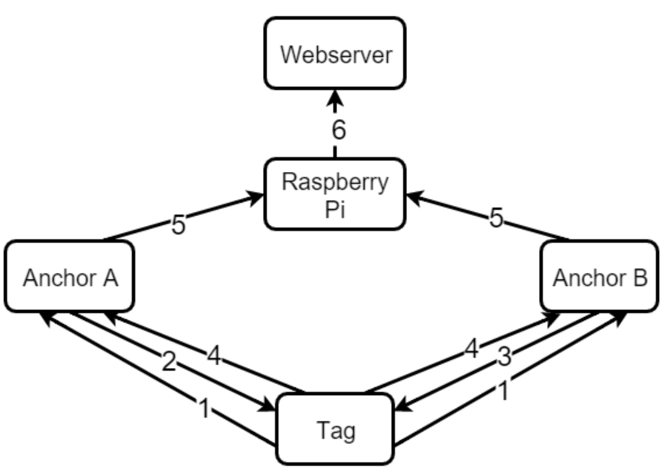
As you can see, we decided on BDSR as our ranging protocol. One reason for this is that, for two Anchors, each ranging session only requires four messages. DS-TWR would require six (three for each Anchor), and while SS-TWR only takes four as well, the number of messages needs scales twice as fast with the number of Anchors for SS-TWR. Thus, BDSR will perform better if we decide to add Anchors.

Additionally, because the Tag initiates the ranging (and thus is not constantly listening) and only has to send two messages (regardless of the number of Anchors), BDSR allows for the lowest Tag power consumption. Furthermore, BDSR allows for measurements as accurate as DS-TWR, and more accurate than SS-TWR (which does not handle clock drift). Finally, because the Tag only sends broadcast messages in BDSR, Tag discovery is integrated (the devices do not need to know each others addresses).

System Description/Depiction

The goal of our system is to collect range and angle of arrival information about a single Tag, relative to a measurement device, over a period of time and create a web API from which applications can retrieve this information.

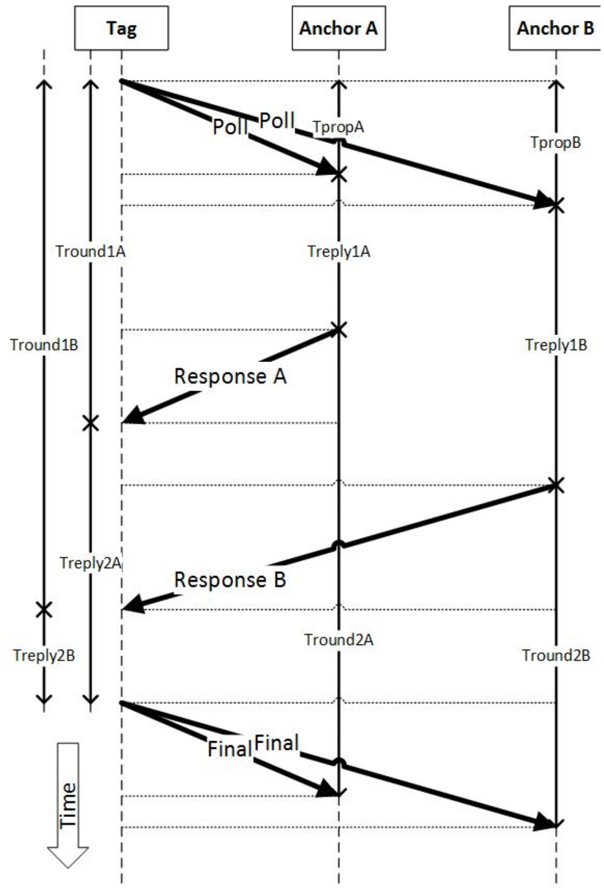
The diagram shows the interaction between various components going through the process of collecting range and angle of arrival information for a single point in time.



Steps:

1. Poll - Tag polls both Anchors in a broadcasted message.
2. ResponseA - Anchor A sends response to Tag.
3. ResponseB - Anchor B sends response to Tag.
4. Final - Tag broadcasts Final message to Anchors containing enough information for each Anchor to estimate range.
5. Anchors tell Raspberry Pi their newly calculated range values.
6. Raspberry Pi calculates angle of arrival and then makes API call to store new data on the Webserver.

Part of the interaction is the UWB communication protocol we will use. We chose a variant of the BDSR protocol described in Figure 4. Because we only intend to use two Anchors, we plan on using something similar to the protocol below:



Protocol Step	Information Sent
Poll	Range Session ID, Tag Address
Response	Range Session ID, Anchor ID
Final	Range Session ID, Tag Address, Tround1A, Treply2A, Treply1B, Treply2B

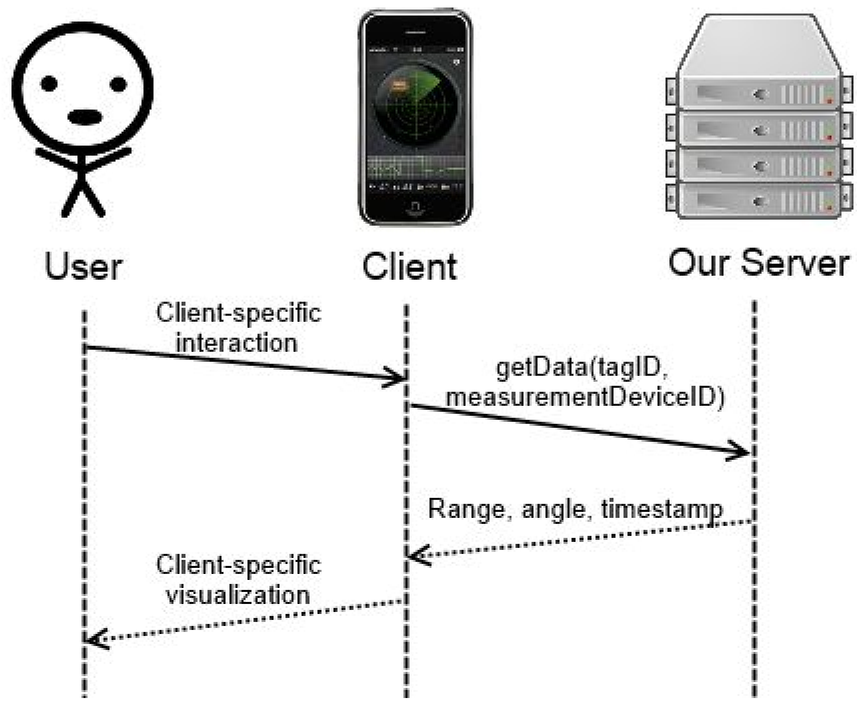
With this protocol, the propagation times, and thus the ranges can be determined for Anchor A and Anchor B as follows:

$$T_{propA} = \frac{(T_{round1A} * T_{round2A} - T_{reply1A} * T_{reply2A})}{(T_{round1A} + T_{round2A} + T_{reply1A} + T_{reply2A})}$$

$$T_{propB} = \frac{(T_{round1B} * T_{round2B} - T_{reply1B} * T_{reply2B})}{(T_{round1B} + T_{round2B} + T_{reply1B} + T_{reply2B})}$$

One thing to note is that when we refer to a ‘Tag’ or an ‘Anchor’ we are referring to the microcontroller / DWM1000 pairs, as specified in the architecture section earlier. Therefore, when we say something such as ‘the Tag sends a message’ we are abstracting away the fact that the microcontroller is actually setting registers on the DWM1000 over SPI in order to command the module to send the message.

At a higher level, our project itself has very little user interaction (by design, for ease of use); the user simply needs to carry a Tag with them while in the presence of a measurement device to use the system itself. However, here is what we would expect user interaction with a client application to look like:

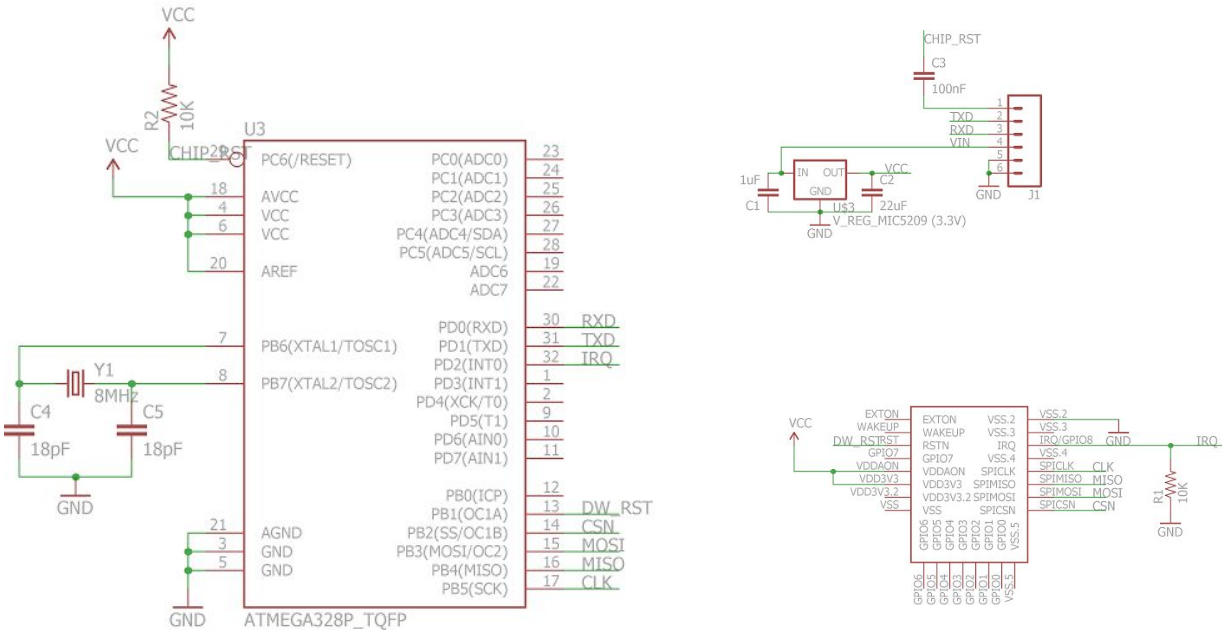


Implementation and Construction

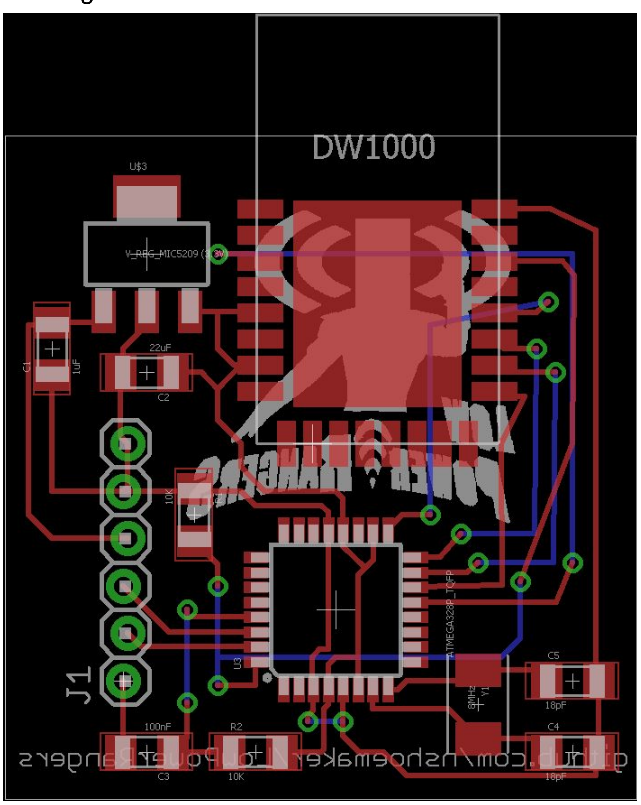
PCB For Tag and Anchor

We decided to create the same PCB design for both Tags and Anchors.This meant the same schematic and board layout. We made this decision for a couple reasons. Using the Atmega328s on both Tag and Anchor boards meant that we would only need to write one SPI driver library to interface with the DWM1000s (whereas we could have had the Raspberry Pi communicate directly with the DWM1000s, but decided against it).

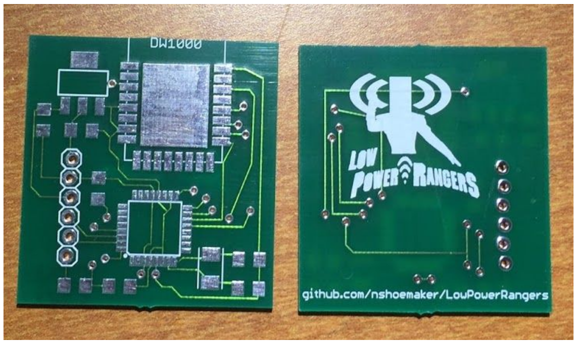
Below is an image of our final schematic. There are three main parts of our schematic.The Atmega328 chip and its associated crystal (left), the FTDI header and voltage regulator (top right), and the DWM1000 module (bottom right).



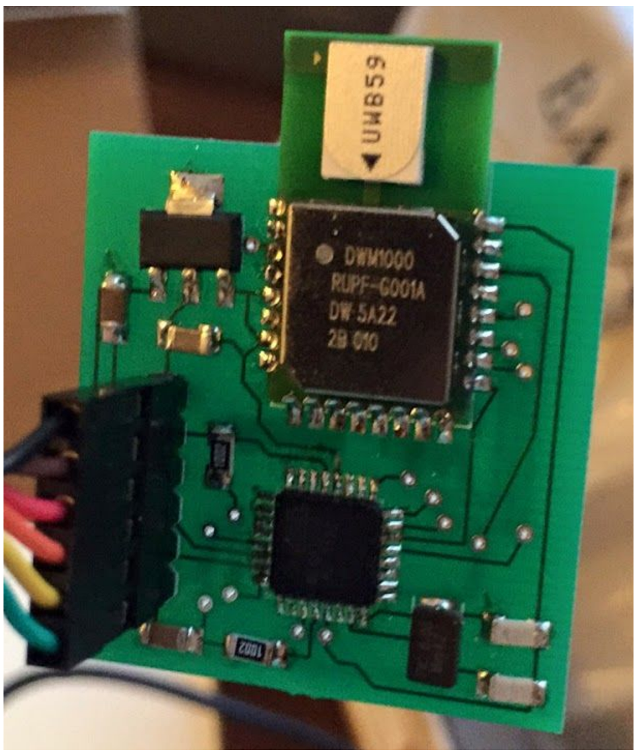
Below is our final board layout. As you can see, the DWM1000 footprint sticks out above the top of our board. This was intentional as the DWM1000 documentation states that the antenna works best with nothing blocking it.

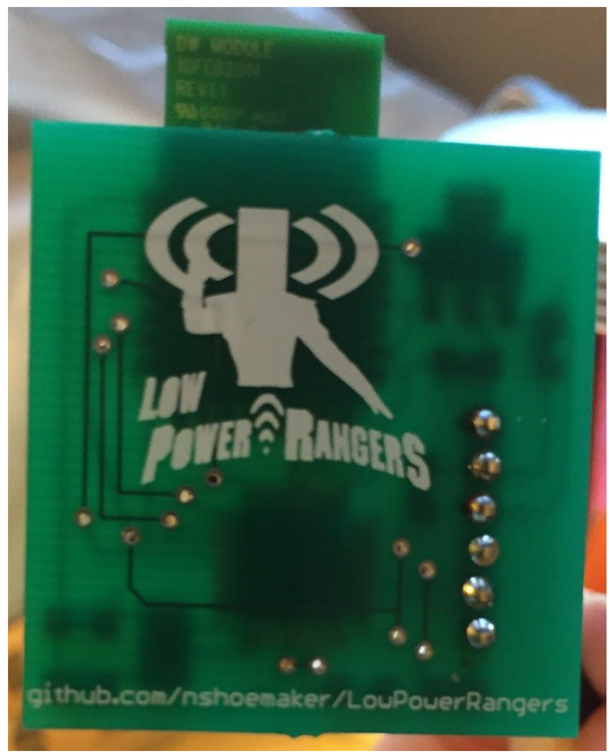


Below we have the printed circuit boards we used for both our Tags and Anchors (pre-bringup). We chose to put components only on one side because we wanted the option of easily laying the Anchors flat against a surface.



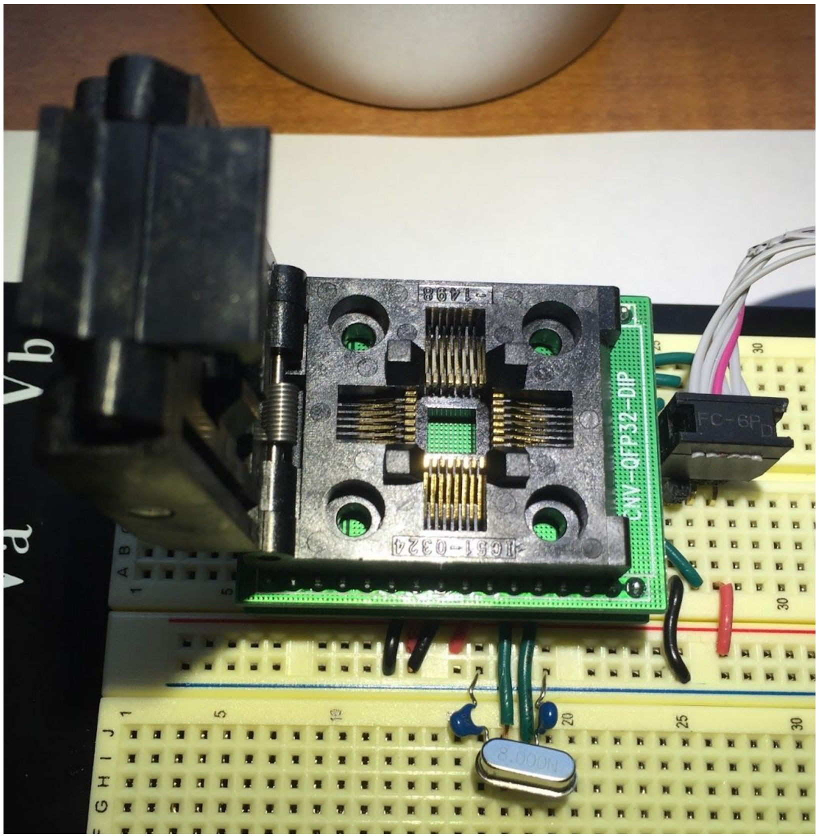
Here are the circuit boards after we brought them up, with the FTDI cable connected.





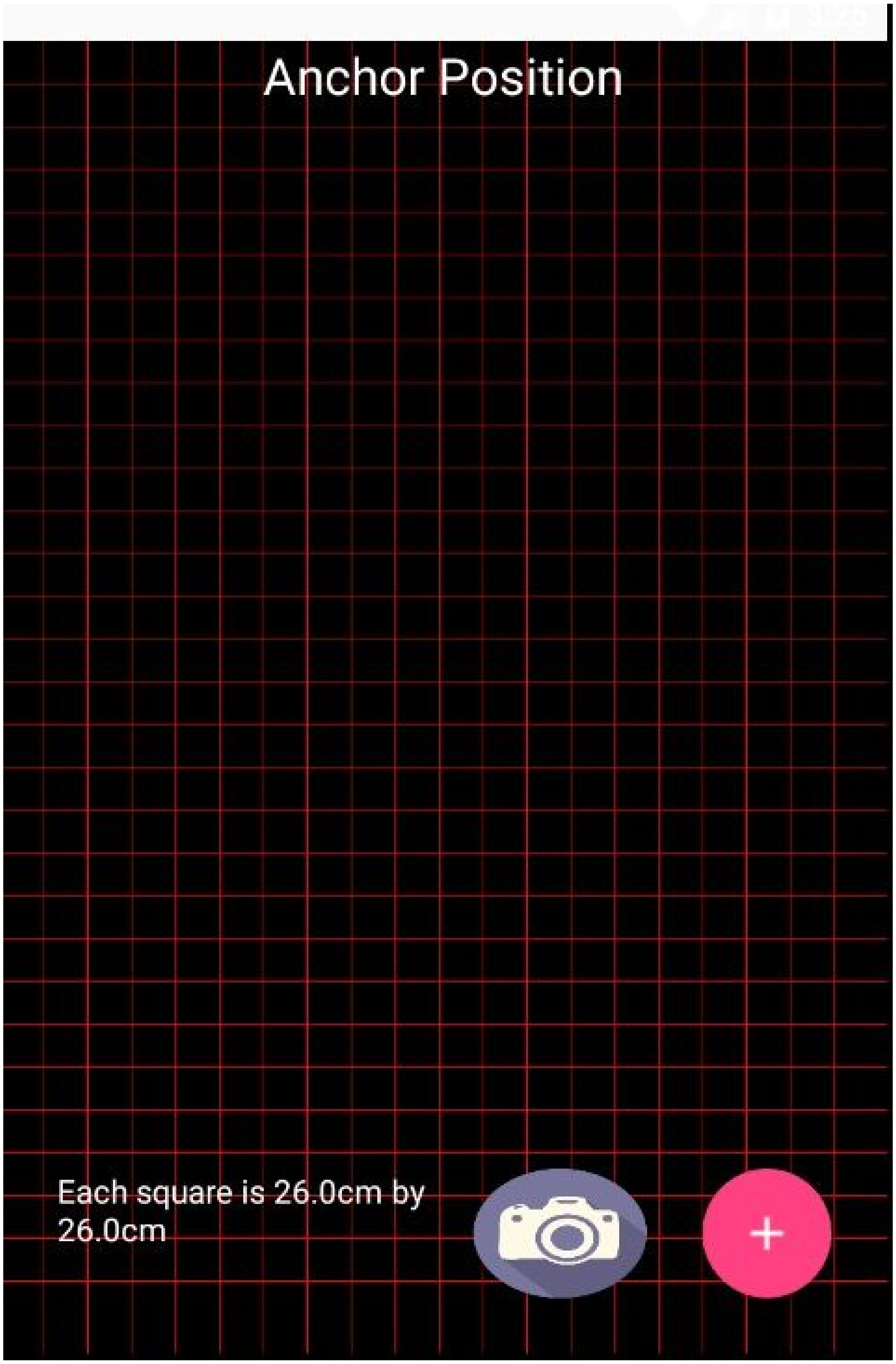
Bootloader Circuit

Another thing worth noting is that we decided to exclude the In-System Programming (ISP) circuit from our Tag and Anchor boards, opting to create an off-board ISP circuit for bootloading our microcontrollers. We decided to breadboard one bootloader circuit for ourselves, using a TQFP test socket.[5] We then soldered the pre-bootloaded microcontrollers onto our board, and were able to reprogram them through the FTDI headers as needed. Below is a picture of the test socket and our bootloader circuit.

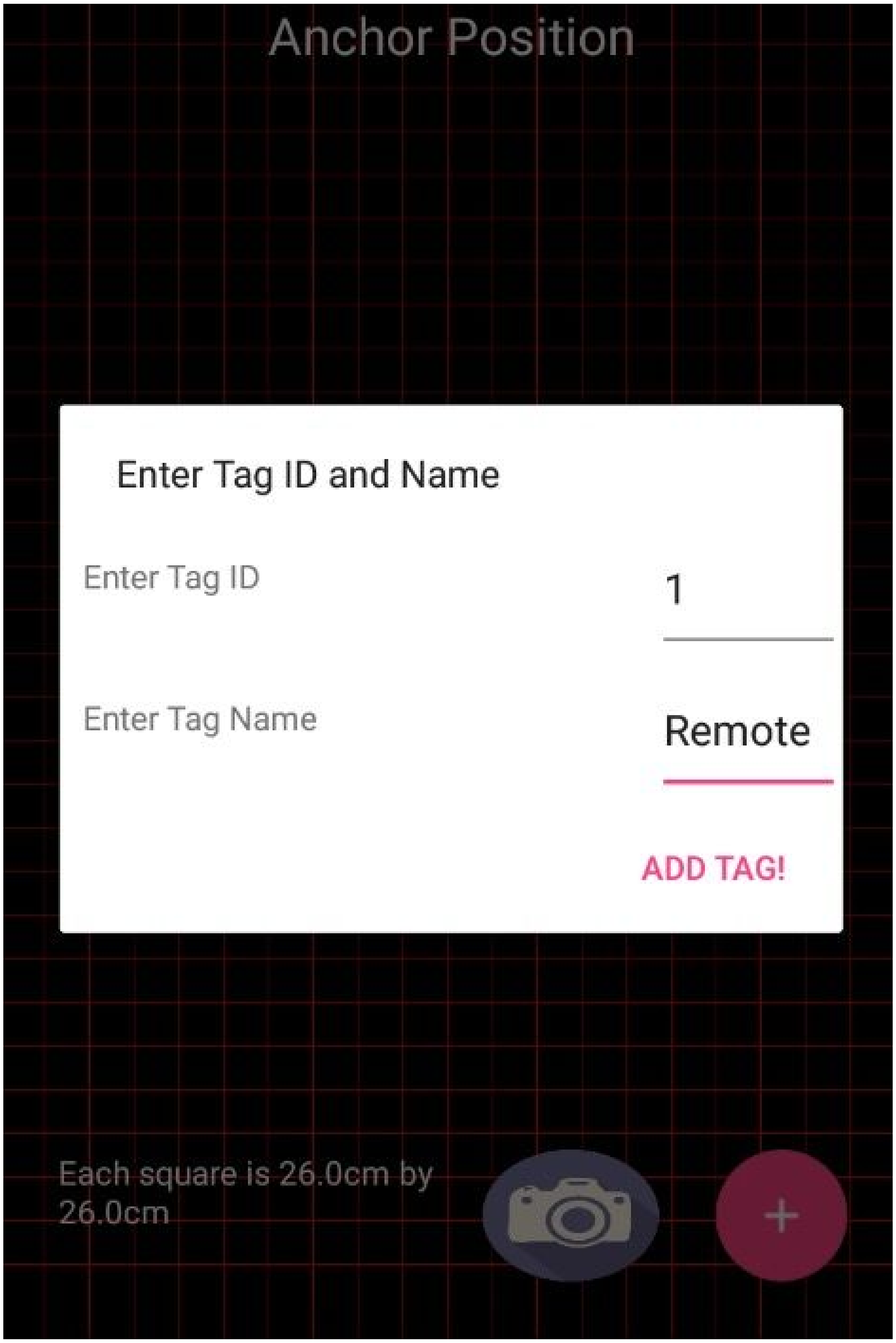

CNV-QFP32-DIP
FC-6P

Android Application

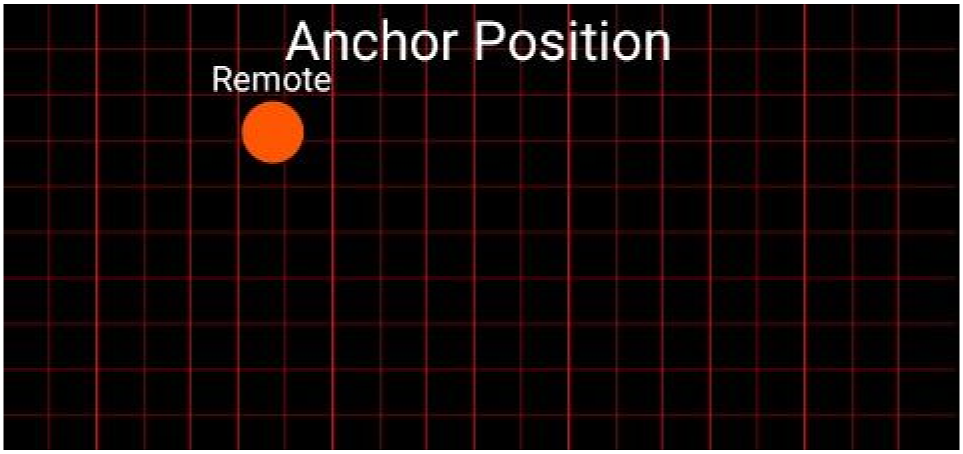
To demo the project, we built an Android application. The application would be used to display the tracking information in an intuitive way. The app pulls data from the server and then displays it on a grid.



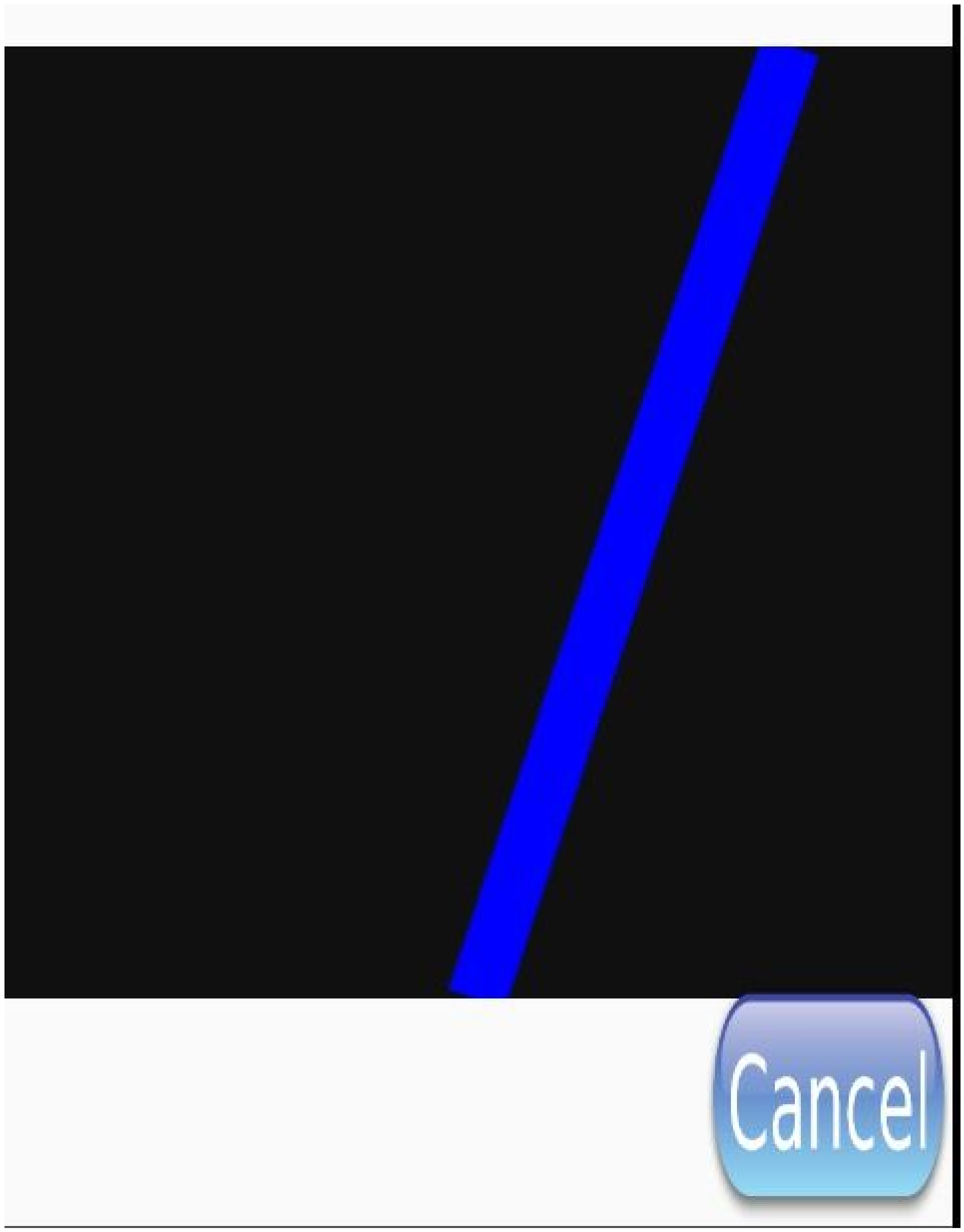
This is the start screen. You can click on the + button to add a tag and give it a name.



Adding the tag then displays it on the screen. It will show up on the screen an appropriate distance away from the anchor positions. The angle will also be the at the correct angle from the anchor position.Below is an example tag position.



The last feature of the application can be accessed by pressing the camera button on the main screen. This will launch the camera application and allow the user to take a picture of their room from the anchor position. It will then draw a line from the bottom of the screen(assumed anchor position) to the place where the tag is. An example image is attached below -

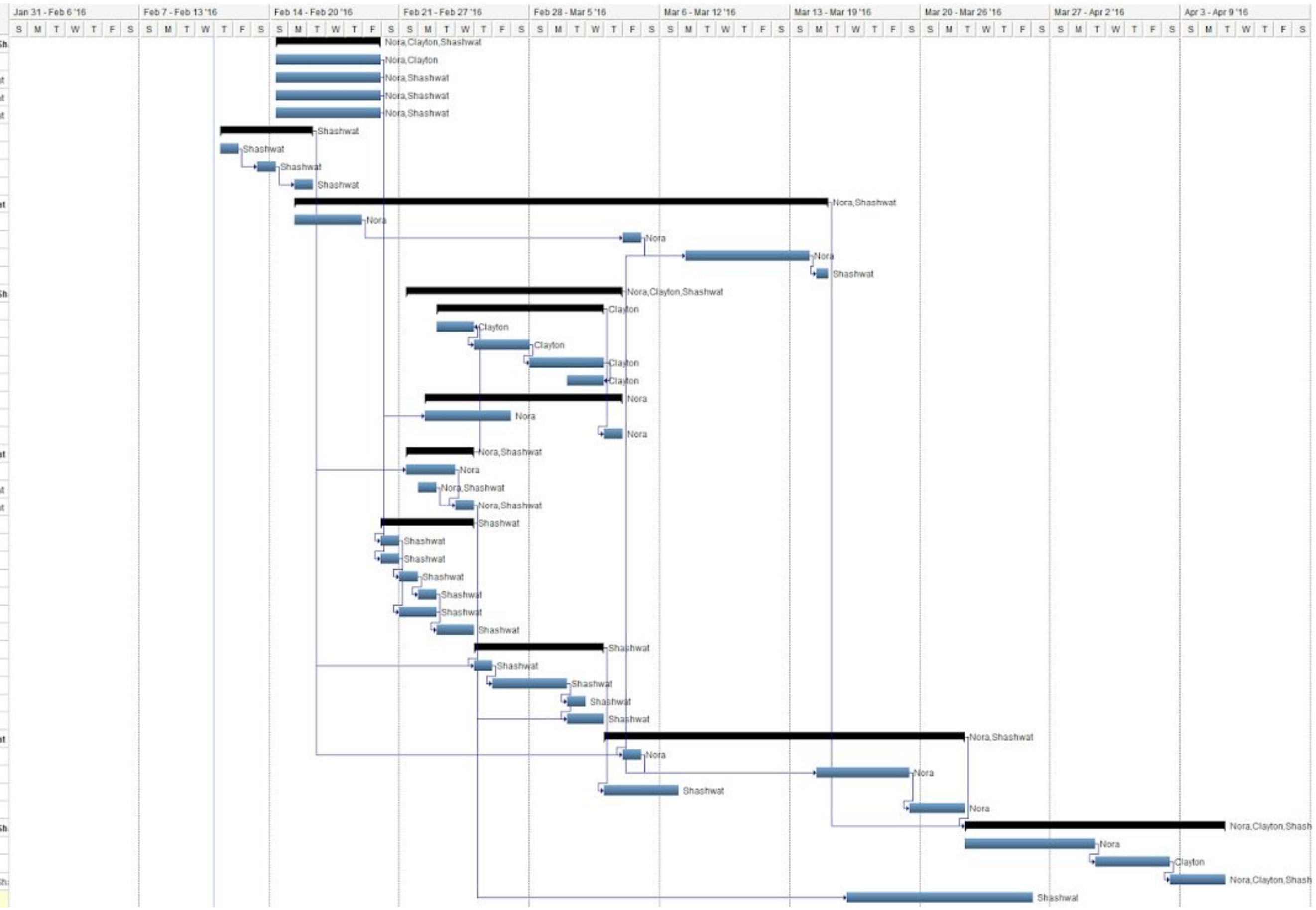


Schedule

Because we have such a hard deadline presented by the demo, we are planning on finishing our project in six weeks (half the time left in the semester). We want to make clear that we are planning on finishing in six weeks with the expectation that it will end up taking longer than that to finish our non-stretch goals. We want to factor in our inevitable mis-estimations and allow for significant padding so that we can truly be finished by the demo date.

(Please zoom in on the schedule on the next page for full details)

Name	Duration	Start	Finish	Predecessors	Resources
Interface and DB Design	**5.63d?**	**02/14/2016**	**02/19/2016**		**Nora,Clayton,Sh**
SPI Interface	5.63d?	02/14/2016	02/19/2016		Nora,Clayton
ER Diagram	5.63d?	02/14/2016	02/19/2016		Nora,Shashwat
Public Web API Interface	5.63d?	02/14/2016	02/19/2016		Nora,Shashwat
Private Web API Interface	5.63d?	02/14/2016	02/19/2016		Nora,Shashwat
Initial Parts Ordered	**5d?**	**02/11/2016**	**02/16/2016**		**Shashwat**
Choose Microcontroller	1d?	02/11/2016	02/12/2016		Shashwat
Fill Out Parts Sheet	1d	02/13/2016	02/14/2016	7	Shashwat
Turn In Parts Sheet	1d	02/15/2016	02/16/2016	8	Shashwat
Hardware Design	**28.63d?**	**02/15/2016**	**03/14/2016**		**Nora,Shashwat**
Initial Schematic	3.63d?	02/15/2016	02/18/2016		Nora
Final Schematic (tested prototype)	1d?	03/03/2016	03/04/2016	11	Nora
Layout PCB	6.63d?	03/07/2016	03/13/2016	12,15	Nora
Order PCBs	0.63d?	03/14/2016	03/14/2016	13	Shashwat
Initial 2-way Communication Prototype	**11.63d?**	**02/21/2016**	**03/03/2016**		**Nora,Clayton,Sh**
SPI Interface	**9d?**	**02/22/2016**	**03/02/2016**		**Clayton**
Get Device ID via SPI	2d	02/22/2016	02/24/2016	24FF	Clayton
Configuration API	3d?	02/24/2016	02/27/2016	17	Clayton
Message Sending Implemention	4d?	02/27/2016	03/02/2016	18	Clayton
Deal with Interrupts Properly	2d?	02/29/2016	03/02/2016	19FF	Clayton
Ranging Protocol	**10.64d?**	**02/22/2016**	**03/03/2016**		**Nora**
2-way Protocol working with mock SPI data	4.63d?	02/22/2016	02/26/2016	2	Nora
Infigration with finished SPI interface	1d?	03/02/2016	03/03/2016	16	Nora
Microcontrollers	**3.63d?**	**02/21/2016**	**02/24/2016**		**Nora,Shashwat**
Breadboard Microcontrollers	2.63d?	02/21/2016	02/23/2016	6FS 5d	Nora
Setup Microcontroller Toolchain	1d?	02/21/2016	02/22/2016		Nora,Shashwat
Microcontroller Hello World	1d?	02/23/2016	02/24/2016	25,26	Nora,Shashwat
Webserver and DB Implementation	**5d?**	**02/19/2016**	**02/24/2016**		**Shashwat**
Setup endpoints (URLs)	1d?	02/19/2016	02/20/2016	4,5	Shashwat
Implements models	1d?	02/19/2016	02/20/2016	3	Shashwat
Setup DB	1d?	02/20/2016	02/21/2016	30	Shashwat
Fill DB with Mock Data	1d?	02/21/2016	02/22/2016	31	Shashwat
Implement views.py (brains of webserver)	2d	02/20/2016	02/22/2016	29	Shashwat
Test Endpoints with Mock Data	2d	02/22/2016	02/24/2016	32,33	Shashwat
Microcrontroller to Raspi Communication	**7d?**	**02/24/2016**	**03/02/2016**		**Shashwat**
Setup Raspi	1d?	02/24/2016	02/25/2016	6FS 5d,28	Shashwat
Implement Serial Receive Code	4d	02/25/2016	02/29/2016	36	Shashwat
Test serial receive with Arduino	1d?	02/29/2016	03/01/2016	37	Shashwat
Test serial receive with our Microcontroller	2d	02/29/2016	03/02/2016	27,37	Shashwat
Full BDSR Breadboarded Prototype	**19.38d?**	**03/02/2016**	**03/22/2016**		**Nora,Shashwat**
Third Board Breadboarded	1d?	03/03/2016	03/04/2016	6FS 5d,15	Nora
Ranging Protocol Implemented	5d	03/14/2016	03/19/2016	15,41	Nora
Raspi Code Implemented	4d	03/02/2016	03/06/2016	35	Shashwat
Ranging Code Tested	3d	03/19/2016	03/22/2016	42	Nora
Full BDSR PCB	**14d**	**03/22/2016**	**04/05/2016**	**10FS 7d,40**	**Nora,Clayton,Sh**
Solder PCBs	7d	03/22/2016	03/29/2016		Nora
Program PCBs	4d	03/29/2016	04/02/2016	46	Clayton
Test PCBs	3d	04/02/2016	04/05/2016	47	Nora,Clayton,Sh
Application Development	10d	03/15/2016	03/25/2016	28FS 20d	Shashwat



Team Member Responsibilities

We plan on breaking down the project into smaller tasks. Each task will have an "owner" who is in charge of it being completed. While it is the owner's responsibility to finish the task, everyone can work on it. See the schedule for finer grained breakdown of tasks.

Task	Person in Charge
Creating an SPI interface to control the UWB module	Clayton Ritcher
Creating/Implementing a protocol for Tag discovery and ranging	Nora Shoemaker
Sending data from the Microcontroller to the Raspberry Pi	Clayton Ritcher
Sending data from the Raspberry Pi to the server	Shashwat Srivastava
Making data available to an API	Shashwat Srivastava
Making application on top of data (NOTE: Stretch goal)	Shashwat Srivastava
Web Master	Nora Shoemaker
Creating the PCB Design	Nora Shoemaker
Hardware Construction	Nora Shoemaker

BOM

Below is the BOM needed to create our complete platform (two Anchors, one Tag, and a Raspberry Pi).

				Price	**$364.40**
Description	Link	Quantity	Unit Price	Shipping Charge	Total Price
Programmer	https://www.sparkfun.com/products/9825	1	$14.95		$14.95
FTDI Cable	https://www.sparkfun.com/products/9717	3	$17.95		$53.85
Raspi	https://www.sparkfun.com/products/13724	1	$49.95		$49.95
Wifi Dongle	https://www.sparkfun.com/products/13677	1	$14.95		$14.95
Headers	https://www.sparkfun.com/products/12693	2	$0.75		$1.50
Decawave	http://www.semiconductorstore.com/cart/pc/viewPrd.asp?idproduct=50013	3	$20.50		$61.50
Power Supply (for Raspi)	https://www.sparkfun.com/products/13831	1	$7.97		$7.97
MIC5209 IC REG LDO 3.3V 0.5A SOT223	http://www.digikey.com/product-detail/en/MIC5209-3.3YS-TR/576-1274-1-ND/771899	3	$1.19		$3.57

RES SMD 10K OHM 1% 1/4W 1206	http://www.digikey.com/product-detail/en/ERJ-8ENF1002V/P10.0KFCT-ND/89822	6	$0.10		$0.60
CAP CER 18PF 50V NP0 1206	http://www.digikey.com/short/3t7p88	6	$0.30		$1.80
CAP CER 1UF 25V X7R 1206	http://www.digikey.com/product-detail/en/CGA5L2X7R1E105K160AA/445-6989-1-ND/2673007	3	$0.18		$0.54
CAP CER 0.1UF (100nF) 50V X7R 1206	http://www.digikey.com/product-detail/en/C1206C104K5RAC7867/399-1249-1-ND/411524	3	$0.07		$0.21
8MHz ±30ppm Crystal 18pF 80 Ohm	http://www.digikey.com/product-detail/en/7A-8.000MAAJ-T/887-1448-1-ND/2627016	3	$0.79		$2.37
ATMEGA328P-AUR IC MCU 8BIT 32KB FLASH 32TQFP	http://www.digikey.com/product-detail/en/ATMEGA328P-AUR/ATMEGA328P-AURCT-ND/3789455	3	$3.70		$11.10
SMT Test Socket - TQFP-32 Breakout	https://www.adafruit.com/products/1240	1	$44.95		$44.95
CAP CER 22UF 25V X7R 1206	http://www.digikey.com/product-detail/en/samsung-electro-mechanics-america-inc/CL31A226MOCLNNC/1276-2728-1-ND/3890814	10	$0.29		$2.90
PCB order		1	$85.44	$6.25	$91.69

Risk Management

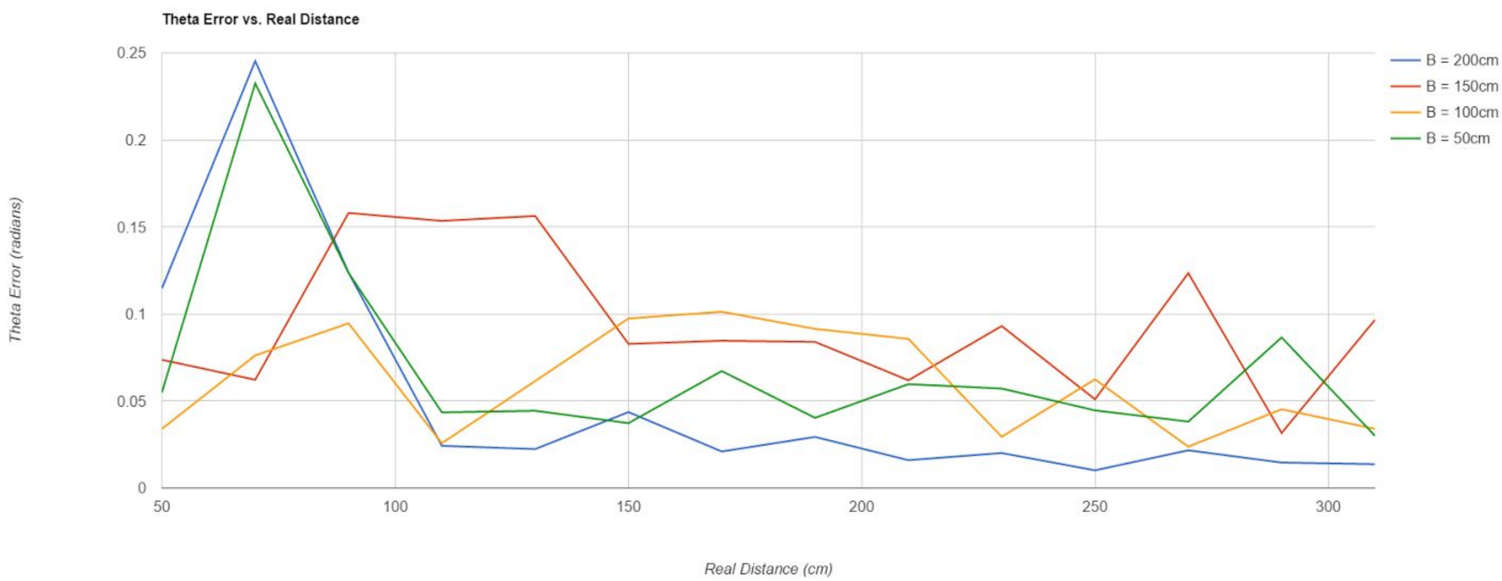
We have identified the use of the Decawave DWM1000 chip technology as a significant risk we are choosing to take on with this project. The use of this new technology, along with the fact that we are using it to find angle of arrival information, something that no one has done publicly yet, is a risk we have identified and thought a lot about. Our first mitigation strategy thus far has been to reduce the scope of the project. This doesn't mitigate the risk of working with the technology entirely, but it does help by allowing us to budget more time towards getting things working in the likely chance that this will be a very hard, time consuming task. Our second mitigation strategy has been to add a significant amount of padding to our schedule and essentially frontload much of the work with the hope of gaining information as quickly as possible about the problems we will incur.

Another risk we have identified is our team's relative lack of experience with embedded programming, especially when it comes to interfacing with real hardware. Every member of our team took 18-349 in Fall 2014. Because of this, we have plenty of experience with simulated hardware, but less experience with real, physical hardware at the embedded level. We all have a background in hobby electronics using Arduino and Raspberry Pi, but we understand that this project will be different and for that reason we expect a significant learning curve. In order to mitigate this risk, we have chosen a project that will involve the use of a single sensor and a single microcontroller. We scale by creating multiples copies of the same setup. Another mitigation strategy we would like to put in place is that we will require a fully working, breadboarded prototype before we fab our PCBs.

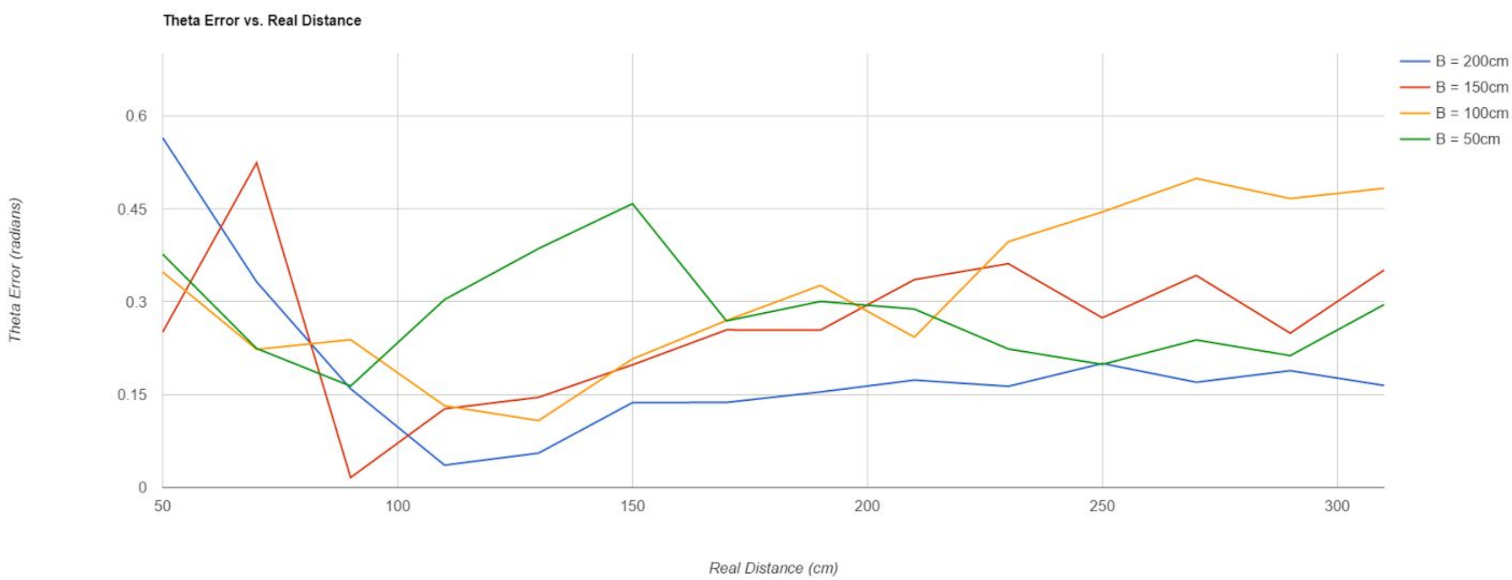
Evaluation

A large part of our project involved research into how baseline distances (between Anchors) affect the accuracy of the angle and range measurements. In order to assess how different baselines affect accuracy, we needed to setup some tests. We decided to sweep out a line at 20 cm increments both at 0 degree (directly perpendicular) and 45 degree angles, using a script that collected 50 readings per increment. We repeated this process for baselines of 50, 100, 150, and 200 cm. Below are graphs of the data we collected.

Theta Error vs. Real Distance at 0 Degrees

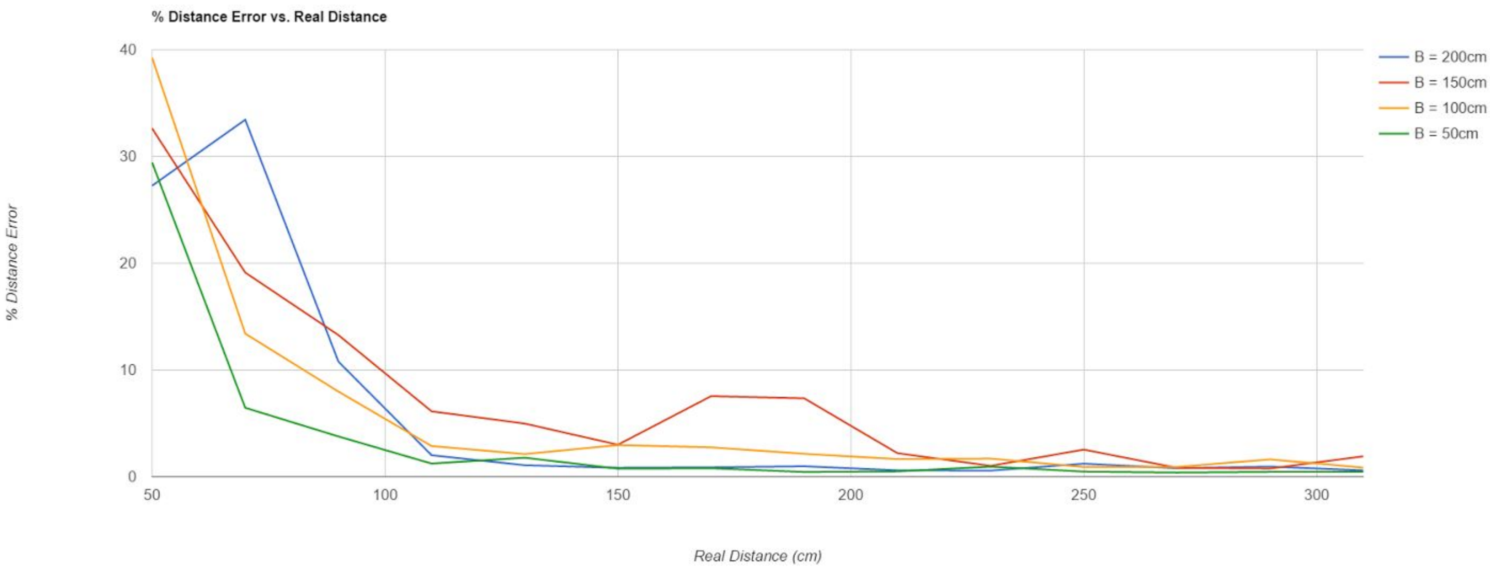


Theta Error vs. Real Distance at 45 Degrees

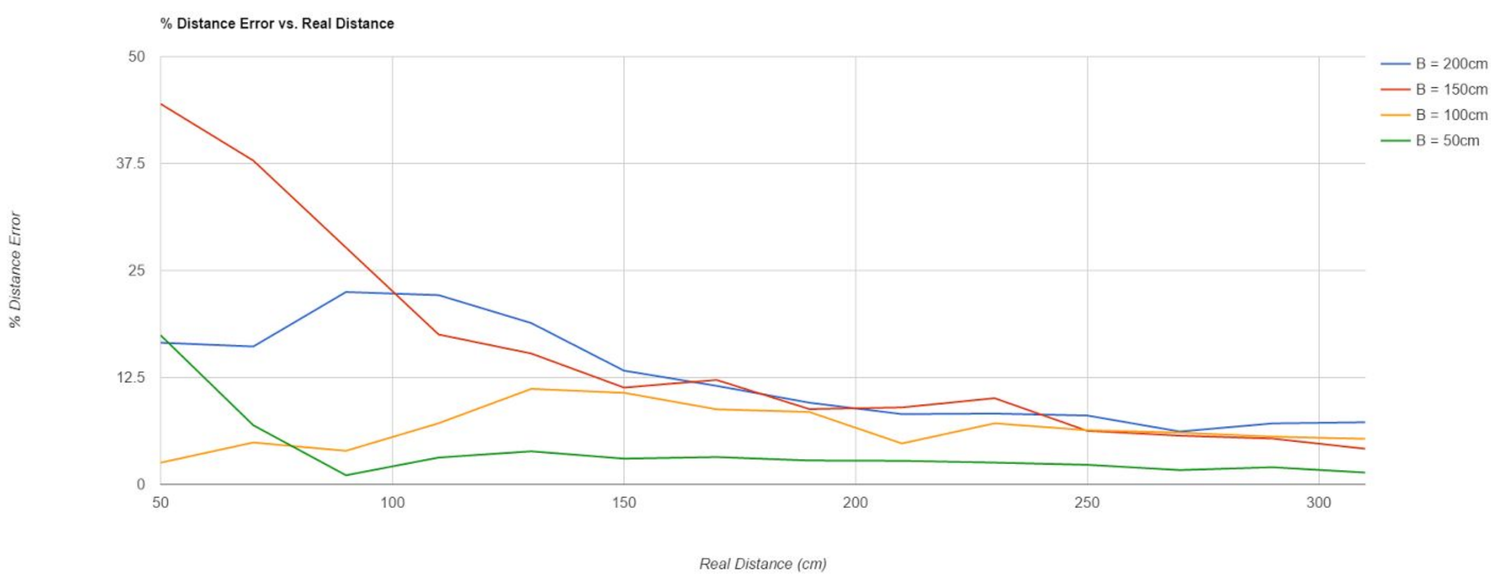


The derivative of the trilateration equations, with respect to measurement error, shows that theta calculations are more robust to noise as the baseline increases. This can be seen in the graphs above -- the theta error for larger baselines is consistently lower than the theta error for smaller baselines.

Range Error vs. Real Distance at 0 Degrees



Range Error vs. Real Distance at 45 Degrees



The derivative of the trilateration equations, with respect to measurement error, shows that range calculations are less robust to noise as the difference between r1 and r2 (distance to each of the two anchors) grows. This can be seen in the graphs above -- the range error for for all baselines is very low when at 0 degrees (r1 = r2), but larger baselines have larger range error when at an angle (the larger baseline forces r1 and r2 to be more different at the same angle).

Conclusions

We learned quite a few things from this project. It was one of our first ventures into anything hardware-related, and the hardware choices we made were relatively risky in nature (heavily reliant on an uncommon chip).

The Only Thing Worse than No Documentation is Wrong/Disorganized Documentation

We ran into lots of problems surrounding the DWM1000 whitepages while writing the SPI drivers. We always identified this as a source of risk in our project, but we perhaps underestimated how much of a source of frustration it would be. The whitepages we had access to were poorly organized, and sometimes solutions to our problems were only mentioned once in a footnote. This wouldn't be so bad if we were using a popular technology with plenty of online documentation and support. However, the most popular existing open source libraries for the DWM1000 were broken, or "still in development" as they prefer to say. This left us quite alone with the Decawave-provided documentation.

Appearances Matter

Most of our project involved building a platform -- a backend, something people hopefully would never need to think about. While the underlying technology was cool enough for us to be wooed, we did not have enough time to also build a really flashy application on top of the framework. In other words, it was difficult for us to communicate the full complexity and potential of our framework in the demo. We took the time to develop a few example applications on top of our platform, but our efforts were much more focused on the platform itself. Initially we didn't think this was an issue, but we hadn't been thinking about the importance of the demo.

No Risk, No Reward

This should probably be retitled to "We Got Lucky" or perhaps "We Invested Quite a Bit of Time Into Those Drivers". We took a big risk deciding to have our project rely completely on a single, new module, the DWM1000. We tried to mitigate the risk by breadboarding and creating a prototype as soon as possible, but in the end if we had not succeeded, things would be looking very bad for us. This was a scary possibility hanging over our heads for much of the initial development. For this reason, we almost always were secretly a week ahead of the schedule Aaron was setting for us. This allowed us to guarantee we never missed a weekly demo.

Related Work

One of the primary reasons we chose our project was to research the feasibility of making angle of arrival calculation using ultra-wideband transceivers, as it has not been publicly demonstrated as of yet. There are other solutions that tackle similar issues, though.

For instance, indoor localization systems using wifi[2], BLE[3], or even UWB[4] could all be used for similar applications as our system. Wifi and BLE systems are less accurate than UWB, though, and UWB indoor-localization systems still require installation of at least three Anchors in each area, on separate walls. Our system would be able to support many similar applications (such as those listed in earlier sections) with the installation of a single device (with two contained Anchors) on a wall in the room. The self-contained property of our Anchors device also means that it could be mounted to something other than a wall, such as a robot, and used to calculate relative range and angle information to a tag from what it is mounted to.

References

1. http://www.decawave.com/support/download/file/nojs/725
2. http://redpin.org/
3. http://estimote.com/
4. http://www.timedomain.com/
5. https://www.adafruit.com/products/1240