Stark Industries

James Chen
Meijiao Li
Jonathan Mitchell
Christian Hurd

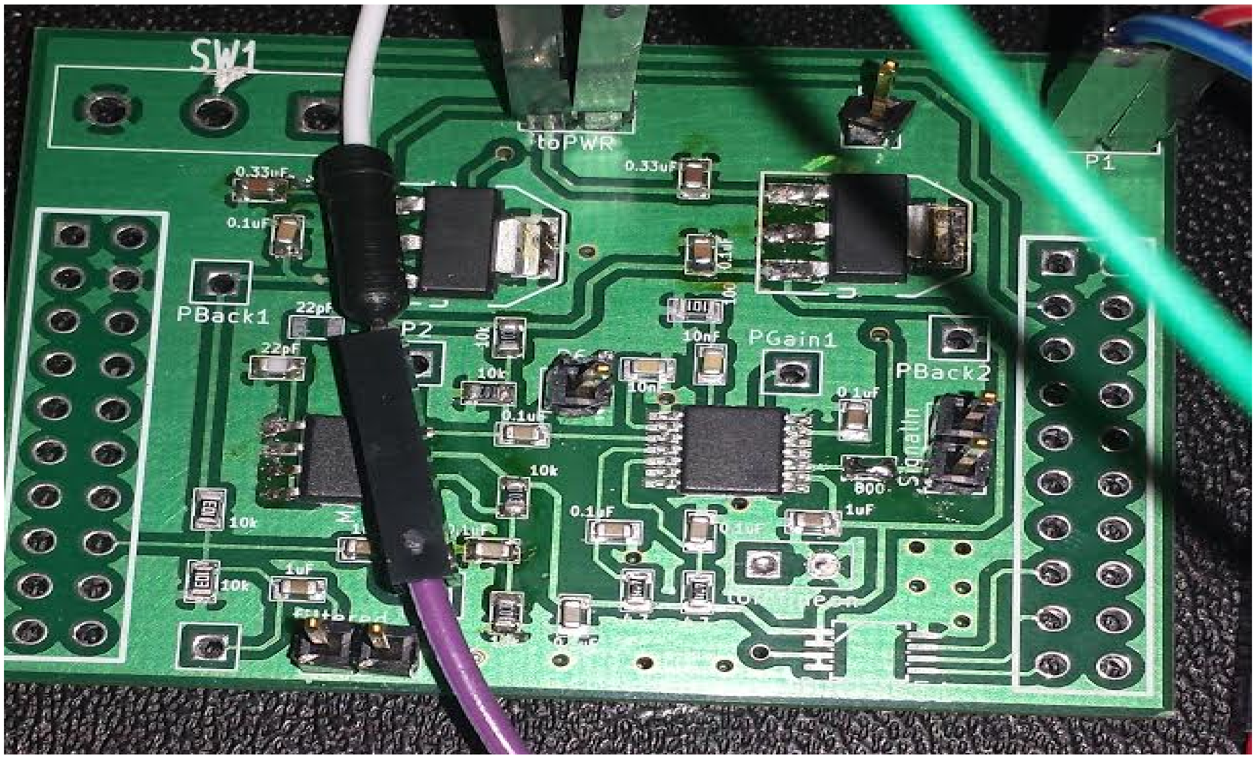
EEC 134B
Radar Report
06/05/15

I. Design Goals

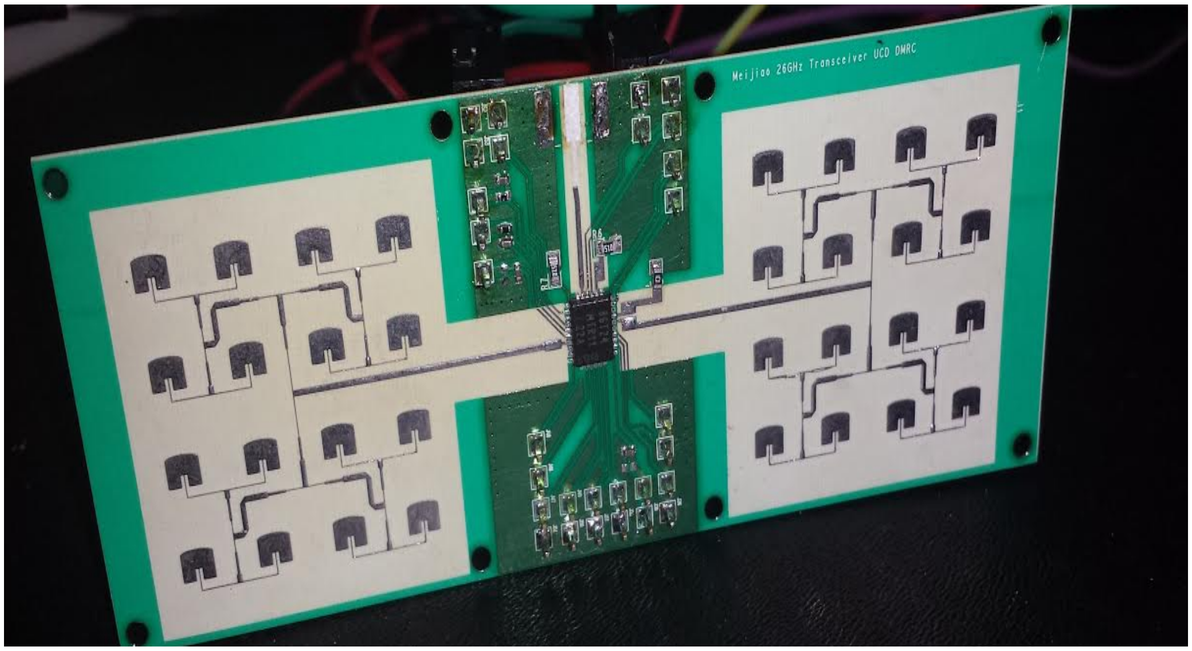
Design, Fabricate, and test a 24-GHz FMCW Doppler and range radar

II. Parts used

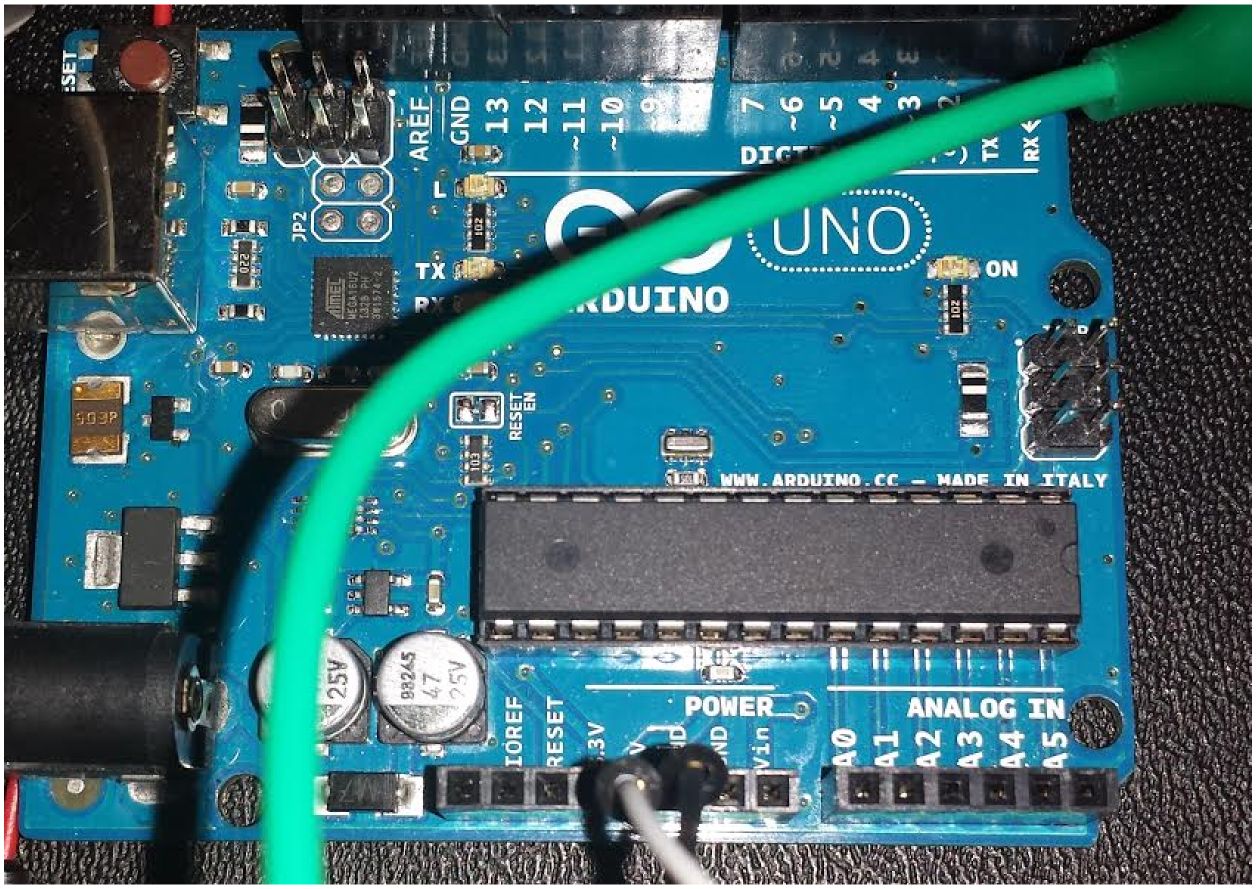
A. Broadband Filter & Voltage Regulator PCB (James)



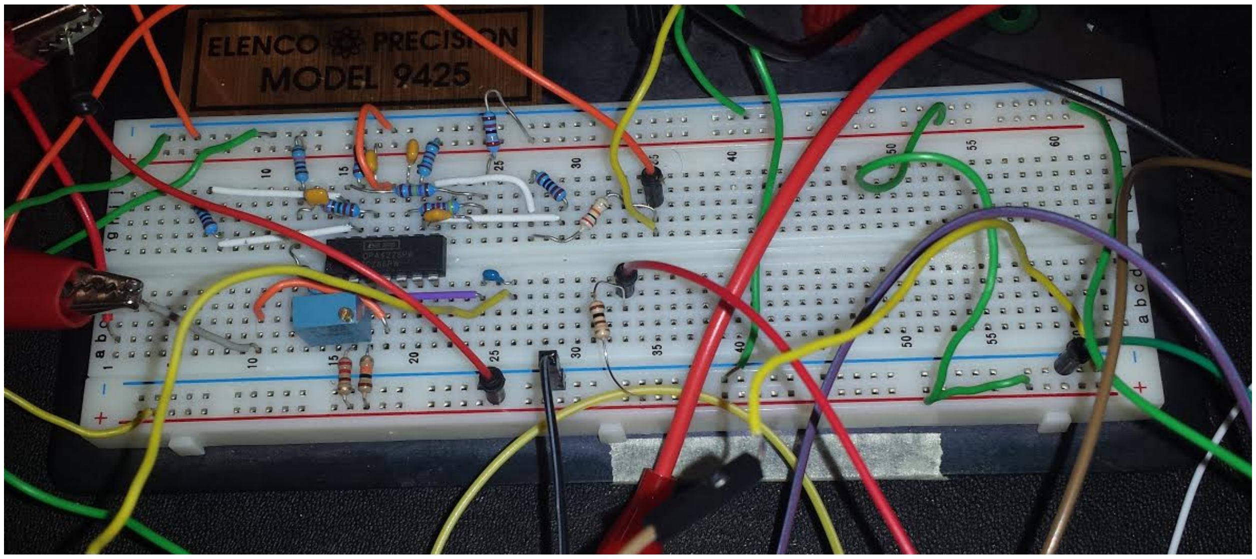
B. Infineon Transceiver and Tx/Rx antennas (Meijiao)



C. SPI, ardurino (Jonathan)



D. Breadboard Active Low Pass filter (Christian)



1. Breadboard, dual power supply, function generator, and oscilloscope
2. Texas Instrument OPA4228 quad op-amp chip
3. Potentiometer
4. C1 = 1 uF, C2-C5 = 1nF
5. R1 = 10K
   R2 = 220
   R3 = 20 K potentiometer
   R4 = 8.45 K
   R5 = 102 K
   R6 = 7.15 K
   R7 = 12.1 K
   R8 = 1K
   R9 = 17.4 K
   R10 = 28 K
   R11 = 4.12 K
   R12 = 1.62 K
   R13 = 1K
   R14 = 20 K

III. Procedures

A. Broadband Filter & Voltage Regulator PCB (James)

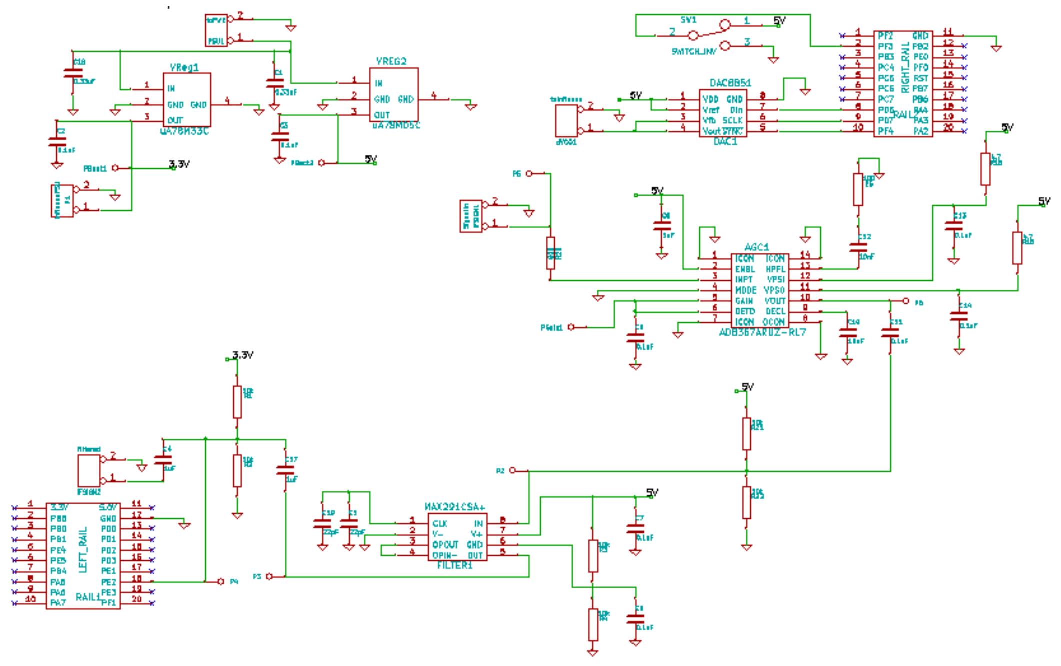
From the IF output of the Infineon transceiver, the baseband signal needs to be filtered and amplified in order for the signal to be better processed. Therefore, an AGC (Automatic Gain Control) and a filter is selected for the PCB design. Major components used are voltage regulators (7+ to 5V, 7+ to 3.3V), AGC(500 MHz, Linear-in-dB VGA with AGC Detector, AD8367ARUZ), filter(Maxim Integrated MAX291CSA+, 8th-Order, Lowpass, Switched-Capacitor Filters).

Bill of Materials (not available in lab or for sample)

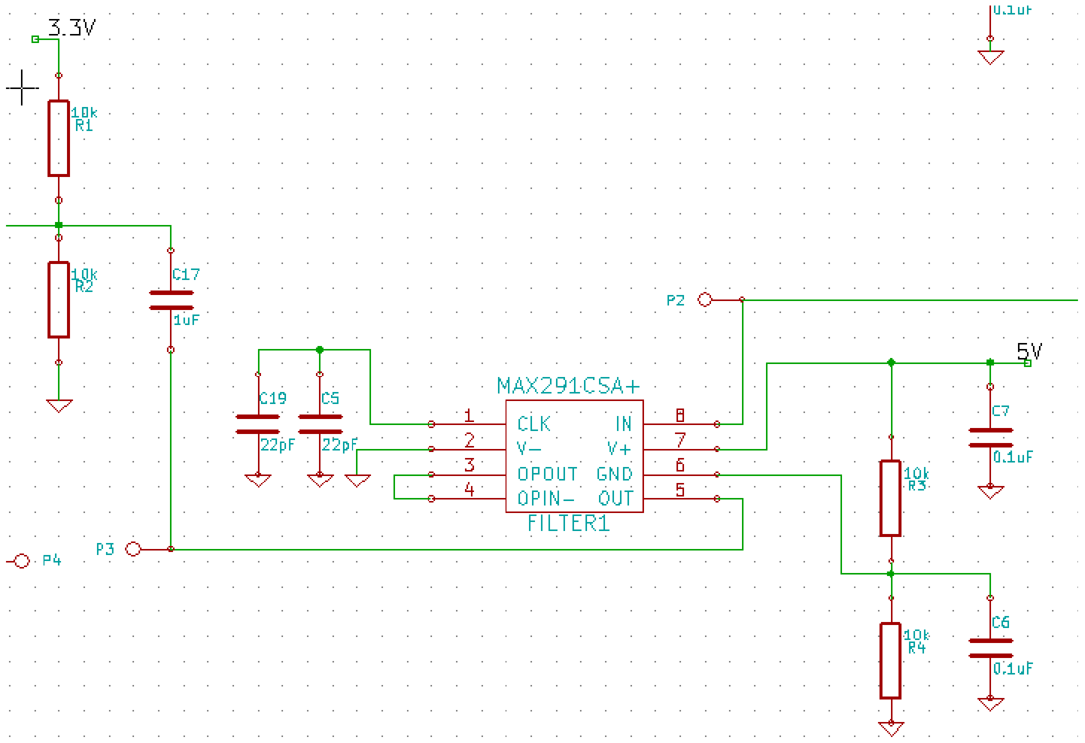
Qty	**Unit**	**Item No.**	**Description**	**Shipping Notes**	**Not to Exceed**	**Unit Cost**	**Total**
1	**EA**	**CKC5102-ND**	**SWITCH SLIDE SPDT 4A 125V HTTP://WWW.DIGIKEY.COM/PRODUCT-SEARCH/EN?KEYWORDS=CKC5102-ND&WT.Z_HEADER=SEARCH_GO**			**$2.23**	**$2.23**
1	**EA**	**MAX291CSA+-ND**	**IC FILTER LOWPASS 8-SOIC HTTP://WWW.DIGIKEY.COM/PRODUCT-DETAIL/EN/MAX291CSA%2B/MAX291CSA%2B-ND/1513229**			**$6.66**	**$6.66**

1	EA	AD8367ARUZ-ND	IC OPAMP VGA 500MHZ 14TSSOP HTTP://WWW.DIGIKEY.COM/PRODUCT-DETAIL/EN/AD8367ARUZ/AD8367ARUZ-ND/819986	$10.93	$10.93
3	EA	296-12290-1-ND	IC REG LDO 5V 0.5A SOT223 HTTP://WWW.DIGIKEY.COM/PRODUCT-DETAIL/EN/UA78M05CDCYR/296-12290-1-ND/416584	$0.66	$1.98
3	EA	296-13424-1-ND	IC REG LDO 3.3V 0.5A SOT223 HTTP://WWW.DIGIKEY.COM/PRODUCT-DETAIL/EN/UA78M33CDCYR/296-13424-1-ND/484495	$0.62	$1.86
5	EA	445-5142-1-ND	CAP CER 0.33UF 25V 10% X5R 0603 HTTP://WWW.DIGIKEY.COM/PRODUCT-DETAIL/EN/C1608X5R1E334K080AC/445-5142-1-ND/2093757	$0.11	$0.55
5	EA	P4.7GCT-ND	RES SMD 4.7 OHM 5% 1/10W 0603 HTTP://WWW.DIGIKEY.COM/PRODUCT-DETAIL/EN/ERJ-3GEYJ4R7V/P4.7GCT-ND/282421	$0.10	$0.50

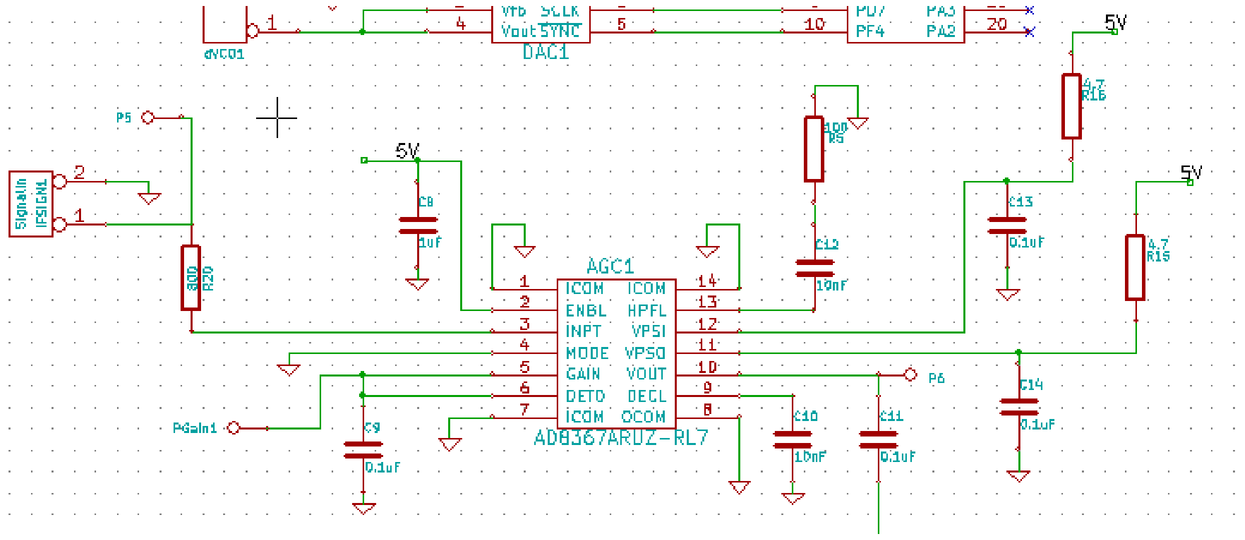
Complete Schematic



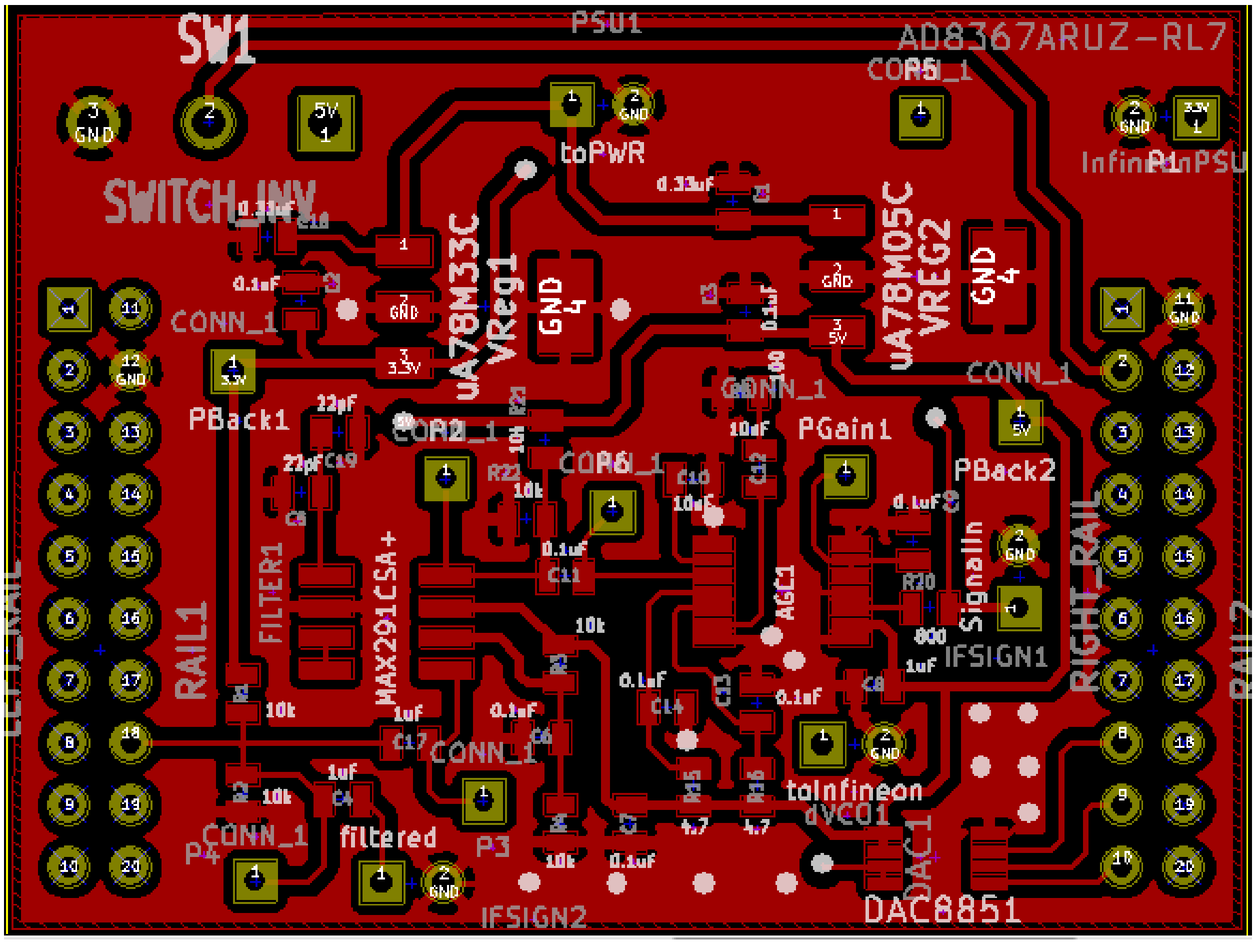
Filter Portion (Zoomed In)



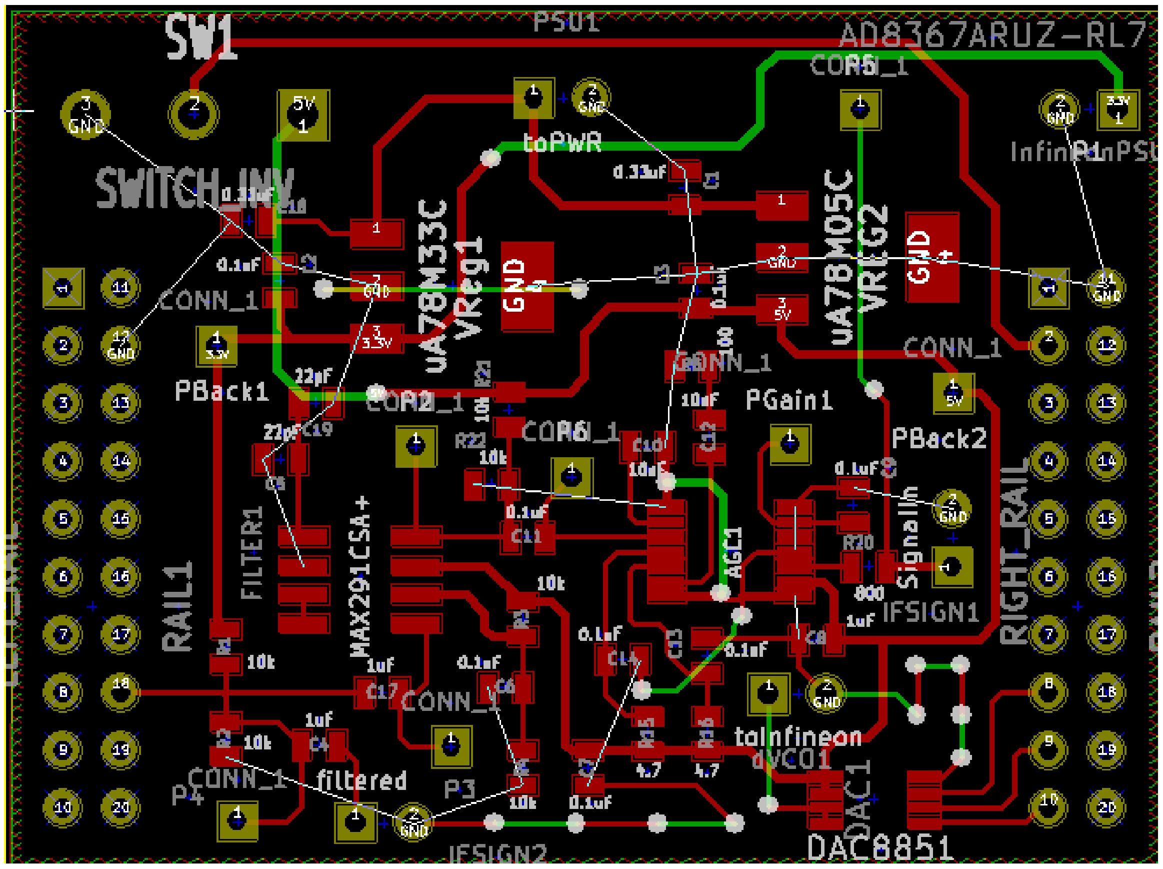
AGC Portion (Zoomed In)



Final PCB Design (Top Layer):



Final PCB Design (Connections, without Copper Fills):

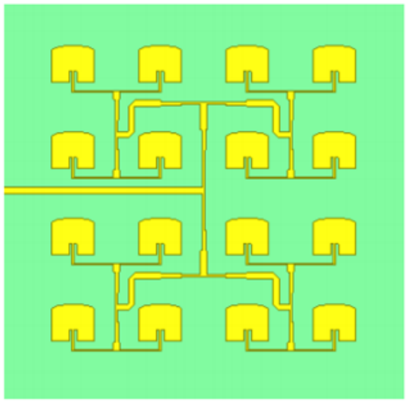


B. Infineon Transceiver and Tx/Rx antennas (Meijiao)

(1) Tx/Rx antennas design

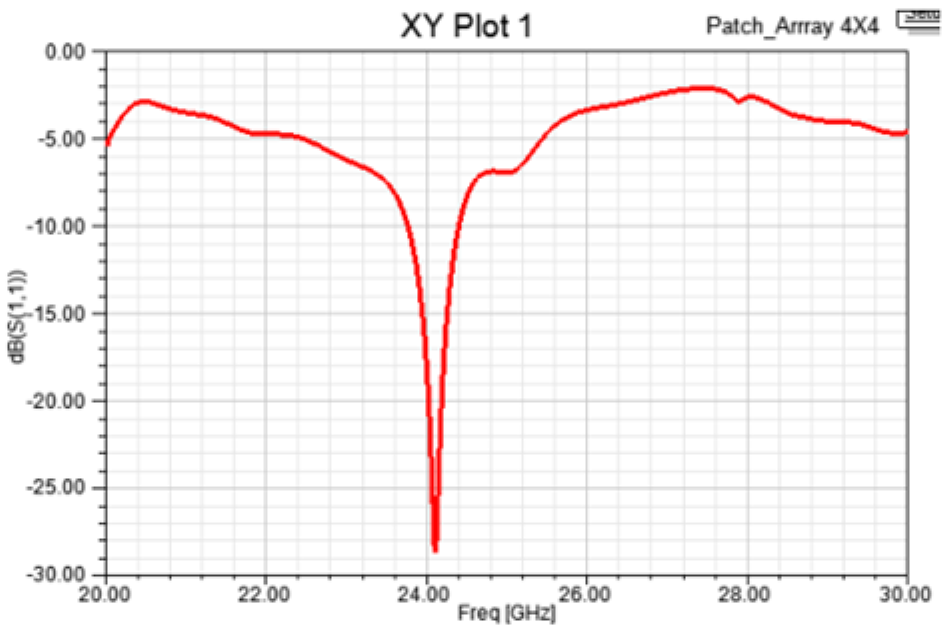
The antenna is a very important part of the radar system. In this project, we chose to use infineon BGT24MTR12 chip. The working bandwidth for the chip is from 24 GHz to 26 GHz. We chose to use a patch antenna for the system because it has the simplest structure and easy circuit implementation. Normally, the patch antenna will have about 4 dB directivity. In our design we chose to use patch antenna array to increase the gain.

a. The patch antenna array structure is shown below.



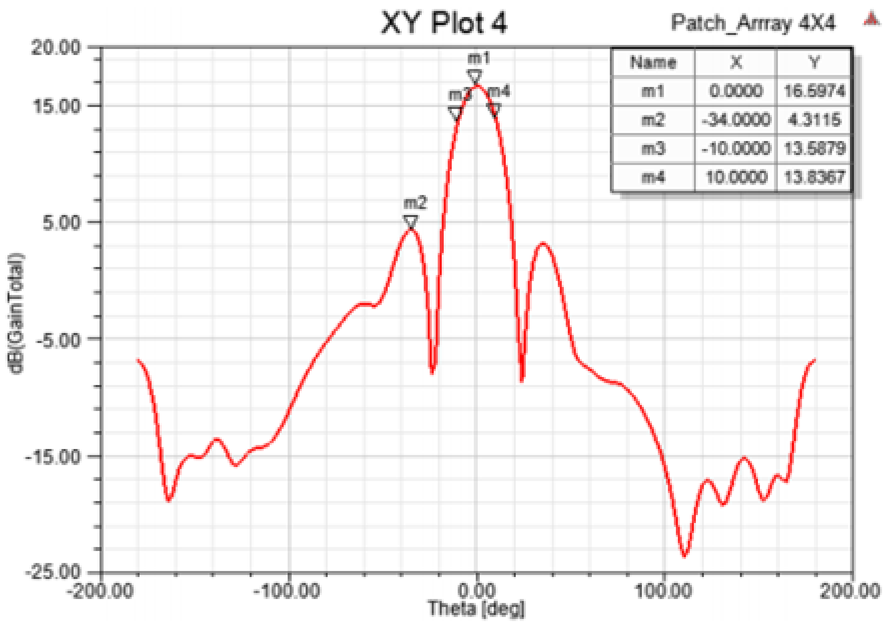
We used a four times four patch antenna array for both the transmitter and receiver part.

b. Patch antenna array return loss simulation results
The figure below shows the return loss of the antenna array design



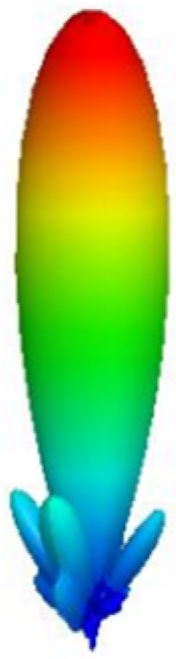
From the plot, we can find that the return loss of antenna array is about 28 dB near 24 GHz. This means that the antenna transmits a lot of power near 24 GHz. Therefore, the array design is sufficient. We considered the differences between the simulation and fabrication results when tuning the array. We also did some tuning for our array to get several different design among the 24 GHz to 26 GHz. When we send out our design to the fabrication company, we chose four array designs. In this way, we can make sure at least one of the array will work perfectly in 24 GHz to 26 GHz system band.

c. patch antenna array farfield pattern simulation results



Name	X	Y
m1	0.0000	16.5974
m2	-34.0000	4.3115
m3	-10.0000	13.5879
m4	10.0000	13.8367

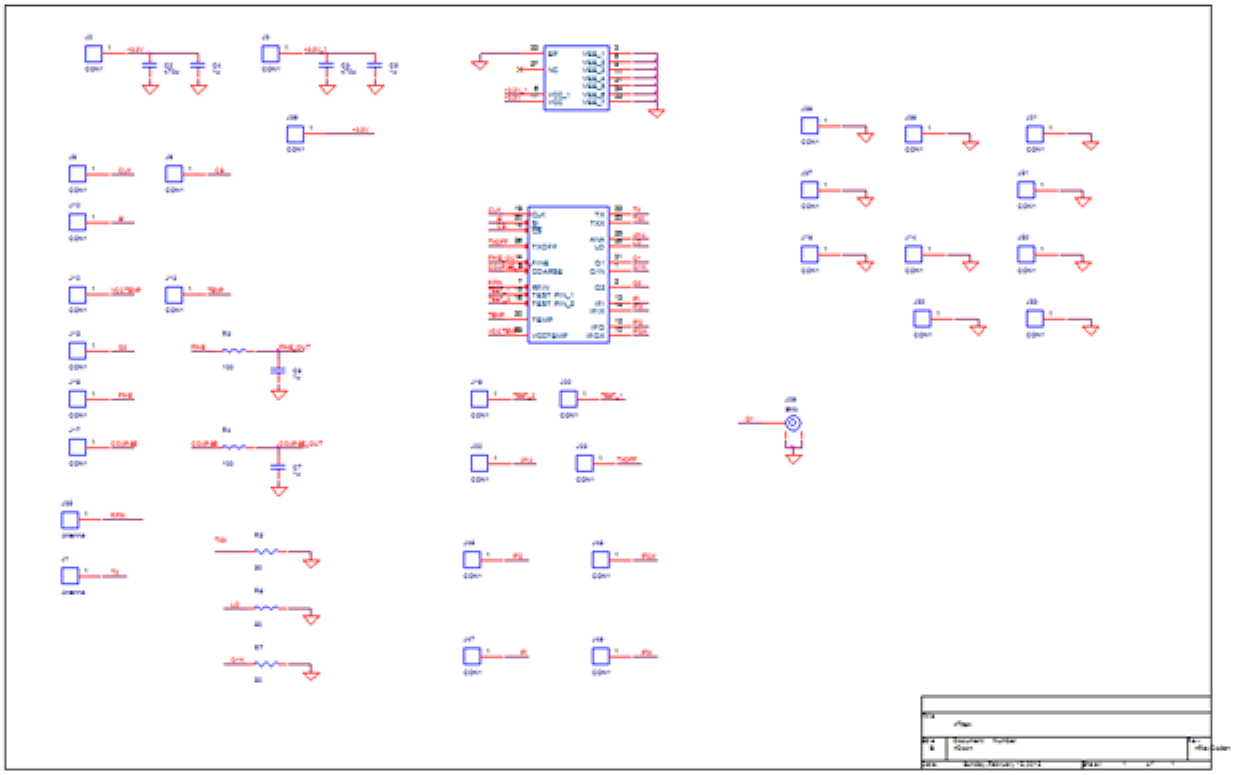
The plot above is the E-plane far field antenna simulation results.



The figure above showed the 3-D far field pattern of the antenna array.
From both the figures, we can find that, the antenna array has a high 16.6 dB gain. And the main lobe is about 12 dB higher than the side lobe.

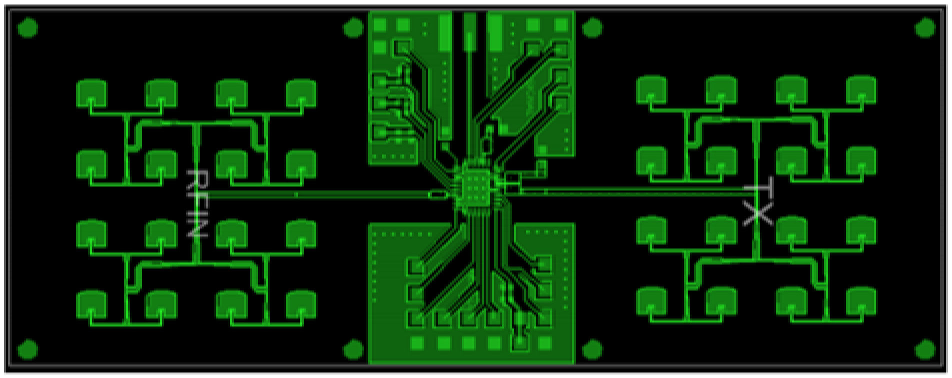
(2) Infineon Transceiver PCB design

a. The schematic of the infineon transceiver

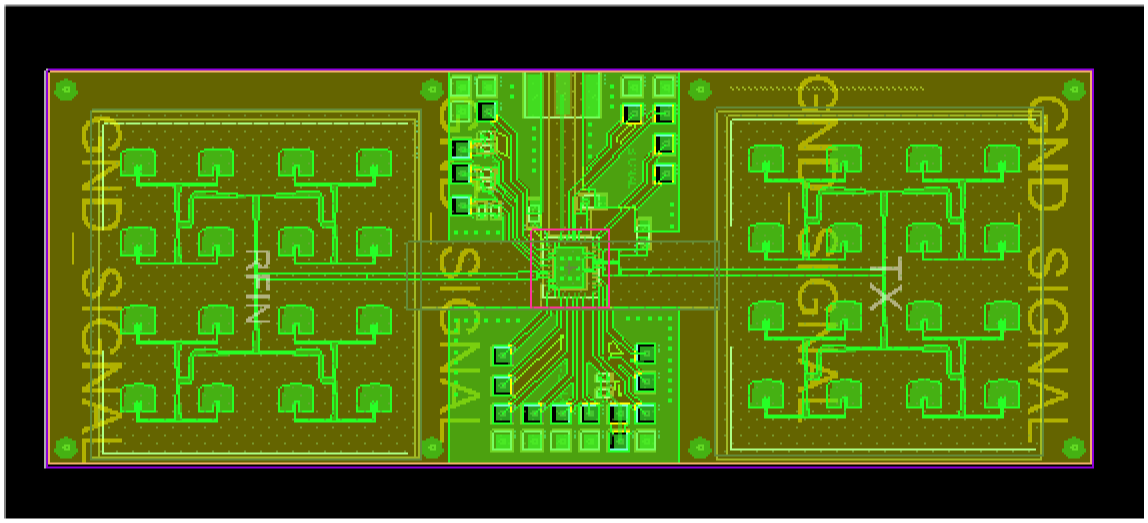


b. The layout of the infineon transceiver

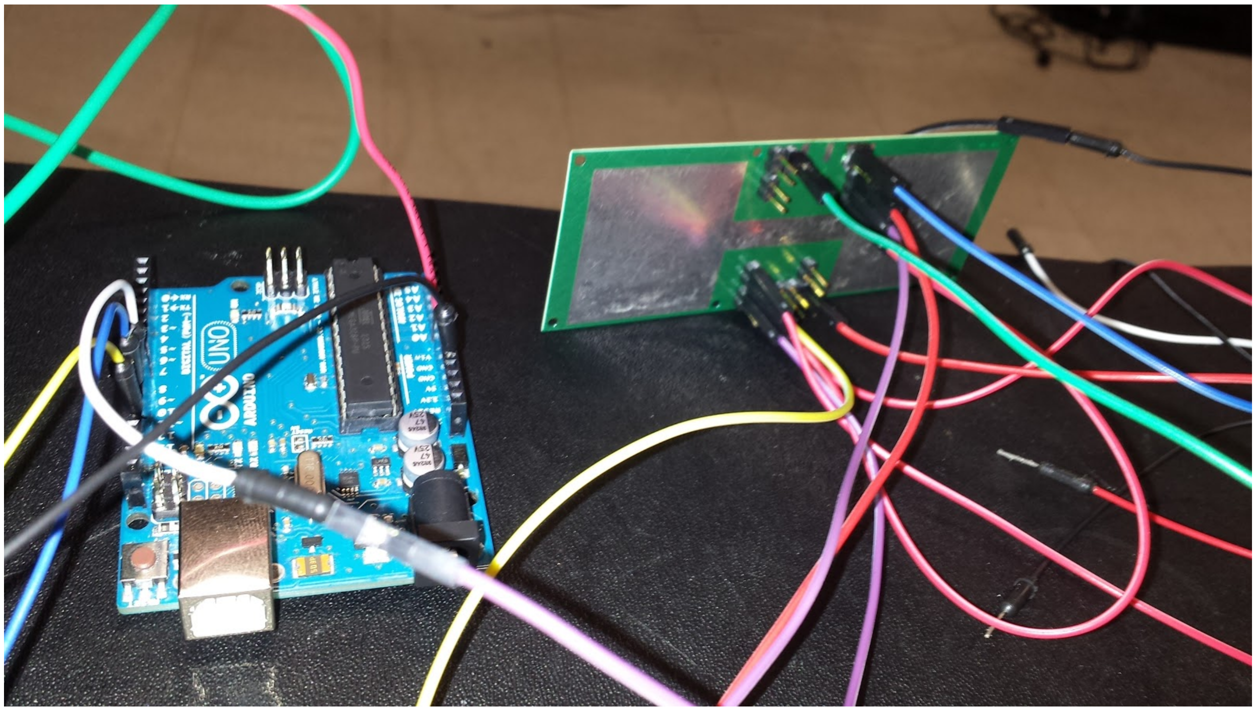
1. Top layer



2. Top layer with the bottom layer



C. SPI, arduino (Jonathan)



In order to change certain parameters on the Infineon chip I used an Arduino Uno to communicate via SPI. The Arduino functions as the Master, while the Infineon chip functions as the Slave.

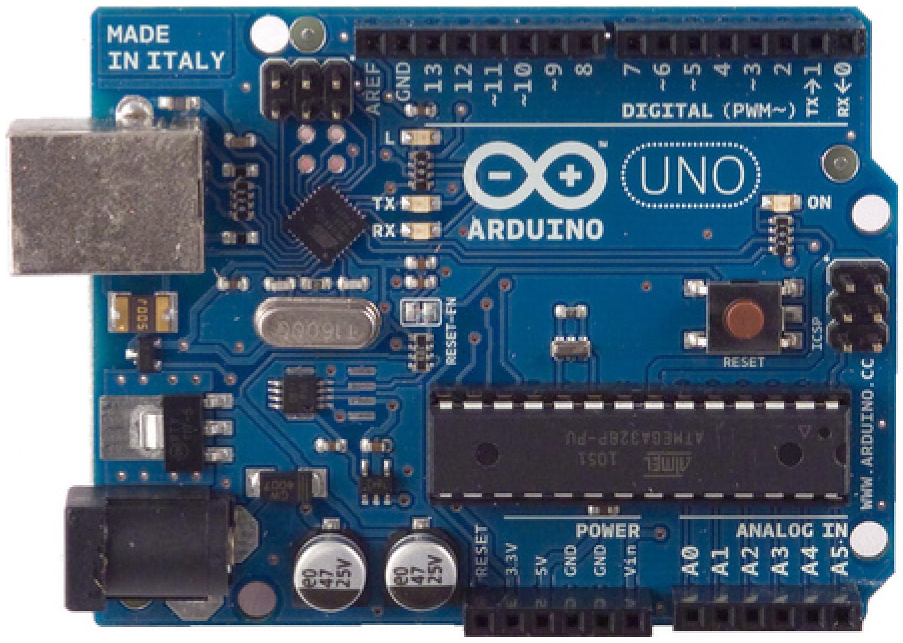


Fig C1.

On the Arduino Uno Pins 10,11,12, and 13 support the SPI.h Library.

SS/ CS = pin 10

MOSI= pin 11

MISO = pin12

SCK = pin 13

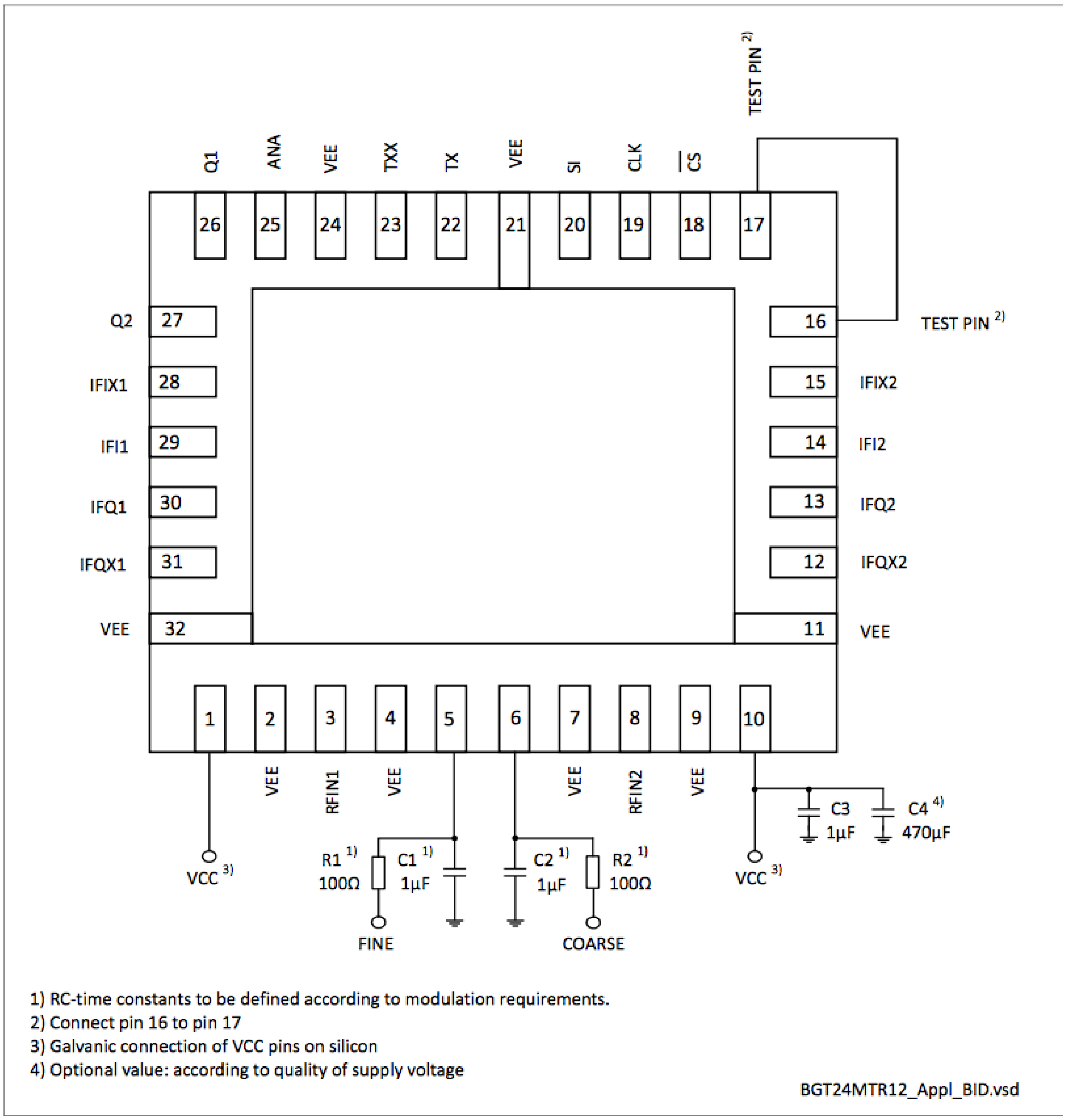


Fig C2.

On the Infineon BGT24MTR:

MOSI = pin 20

CLK = pin 19

CS/SS = pin 18

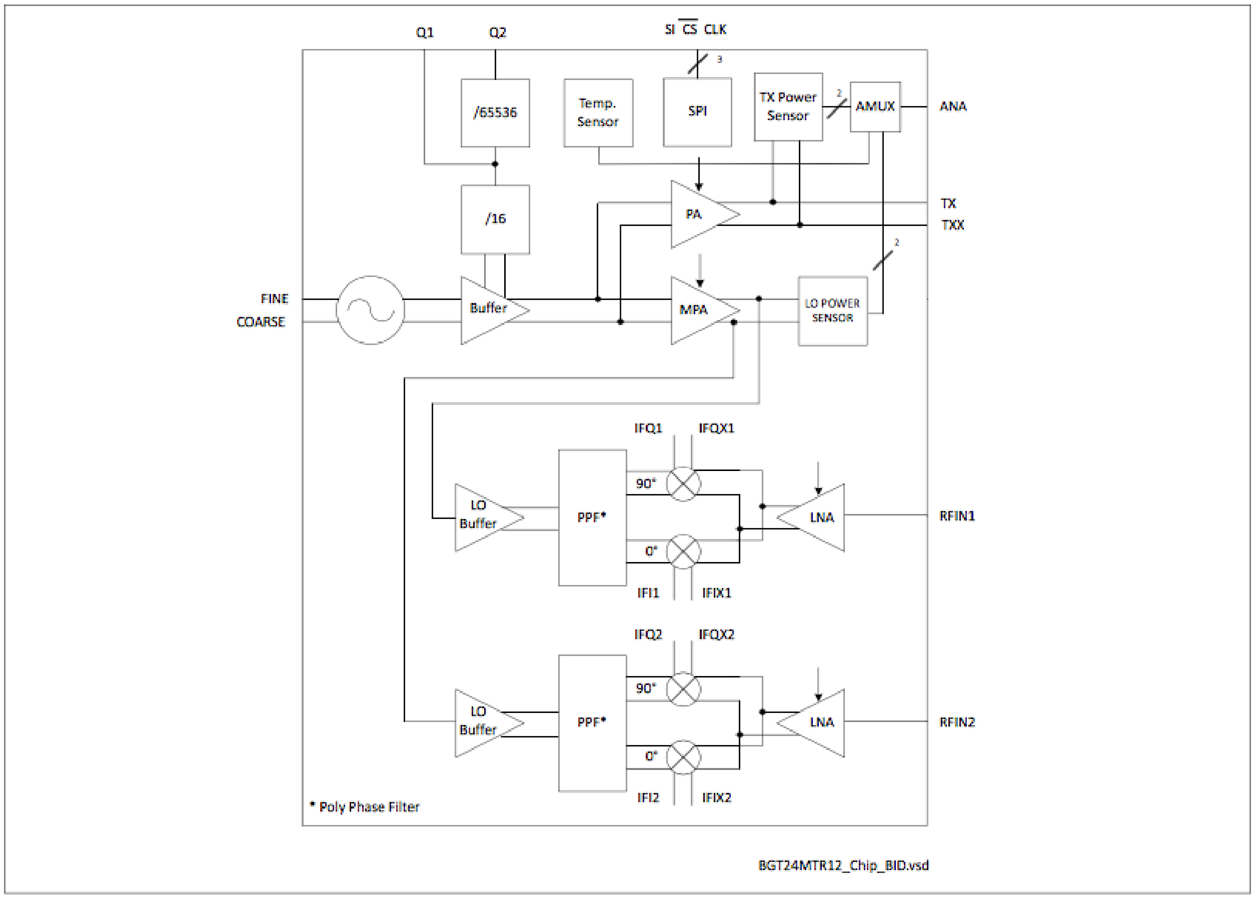


Fig C3

My goal was to turn on the power amplifier, and to turn off the power reduction bits so we have greater power output. However, during testing I decided to run multiple tests with the power reduction bits turned on and with the power reduction bits turned off. It turns out that having the power reduction bits turned on actually allows me to obtain a more accurate signal. This is probably due to the inherent noise from the PA. I have so much noise when I turn those bits off, this can be shown in the Testing and Results section.

Table 11 SPI Block Data Bit Description

Data Bit	Name	Description (Logic High)	Power ON State
15	GS	LNA Gain reduction	low
14	–	Not used	low
13	AMUX2	Analog multiplexer control bit 2	high
12	DIS_PA	Disable Power Amplifier	high
11	Test Bit	Test bit, must be low otherwise malfunction	low
10	Test Bit	Test bit, must be low otherwise malfunction	low
9	Test Bit	Test bit, must be low otherwise malfunction	low
8	AMUX1	Analog multiplexer control bit 1	low
7	AMUX0	Analog multiplexer control bit 0	low
6	DIS_DIV64k	Disable 64k divider	low
5	DIS_DIV16	Disable 16 divider	low
4	PC2_BUF	High LO buffer output power, need to be low otherwise increased current consumption	low
3	PC1_BUF	High TX buffer output power	low
2	PC2_PA	TX power reduction bit 2	high
1	PC1_PA	TX power reduction bit 1	high
0	PC0_PA	TX power reduction bit 0	high

Fig C4.
I sent my data through from the MSB to the LSB in two stages that are separated in time by my master clock frequency. To do this there are is a timing requirement that must be met for the infineon chip.

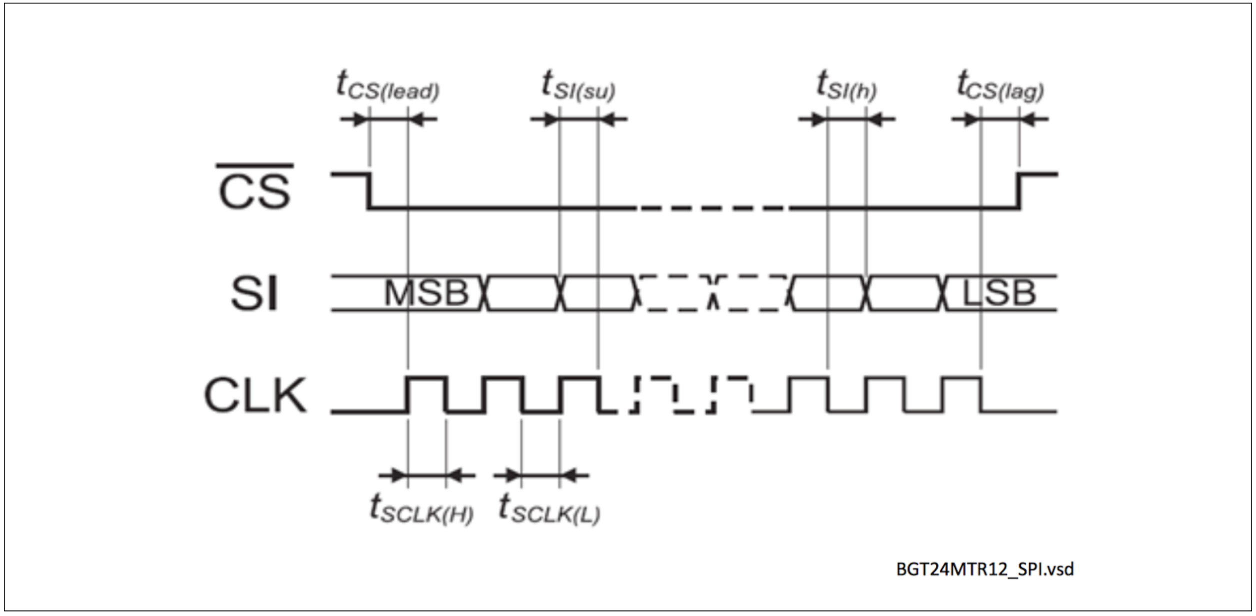


Fig C5
Requirements for Infineon Chip SPI:

“At least 20 ns before first rising edge of the first CLK signal CS needs to be in "low" state. “
This requirement means that my master clock signal must not exceed 100Ghz, because then it would interfere with my SPI Clock transfer. TCS(lead) = 10ns, there are two of them so.

$2Pulses \div 2Tcs \ = \ 2 \ \div 20ns \ = \ 100$ Ghz

Another requirement is that I can only send 8 bits at a time, and they have to be front the MSB to the LSB. This is accomplished in the code as you will see in Appendix A and B.

<u>My SPI Programs are listed in Appendix B and Appendix C.</u>

D). Minimizing noise due to the Arduino

When adding more components to the system I started to receive a ton of noise. Each component you add, increases your noise. The addition of noise would offset the advantage of enabling the power amplifier. When using the power strip to power the arduino I was getting noise from other items plugged into the same power strip, so I decided to use a battery.

I only had 4 AA 1.5V batteries available, so my goal was to put these batteries together in series to give me 6V and then create a voltage divider circuit to bring that 6V down to 5V. I put together the following circuit.

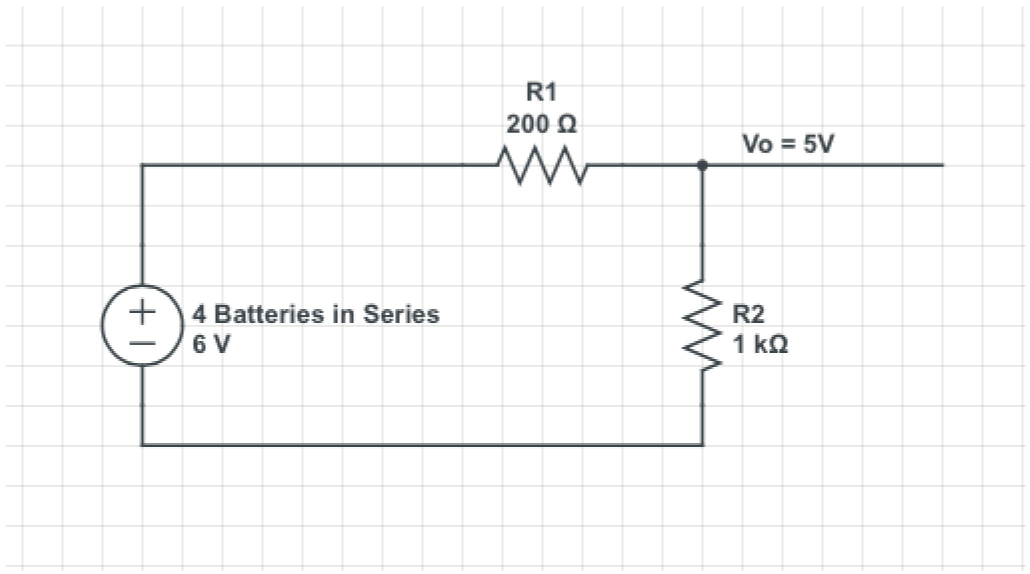


Fig C6

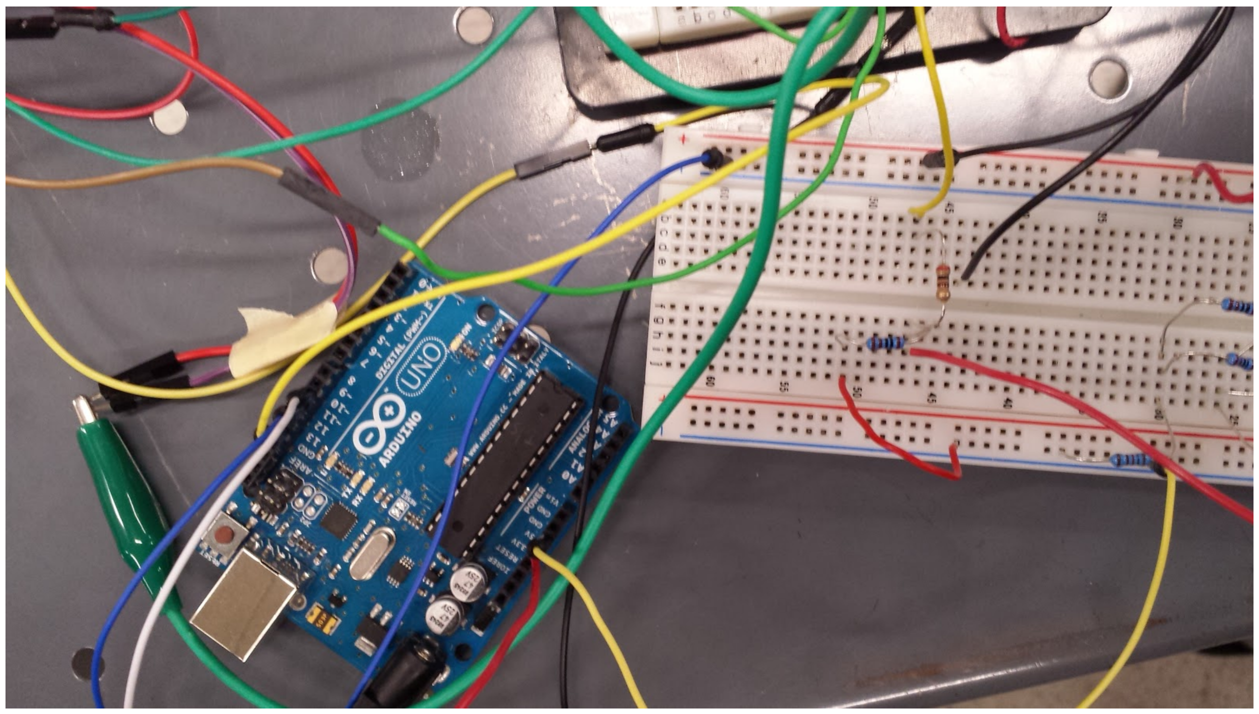


Fig C7.

Vo is the input to my arduino uno 5V bus, then I just grounded the GND port on the arduino with my ground on the breadboard associated with the low pass filter.

I am still receiving a ton of noise, in fact there is a thermal noise associated with each added resistor. According to Pozar, when evaluating the performance of a microwave system, the effect of noise is one of the most important considerations. To account for my value of thermal noise, which is associated with random motion of electrons in my resistors, I use the Rayleigh-Jeans approximation as follows. Note that this approximation is valid for frequencies up to the microwave band.

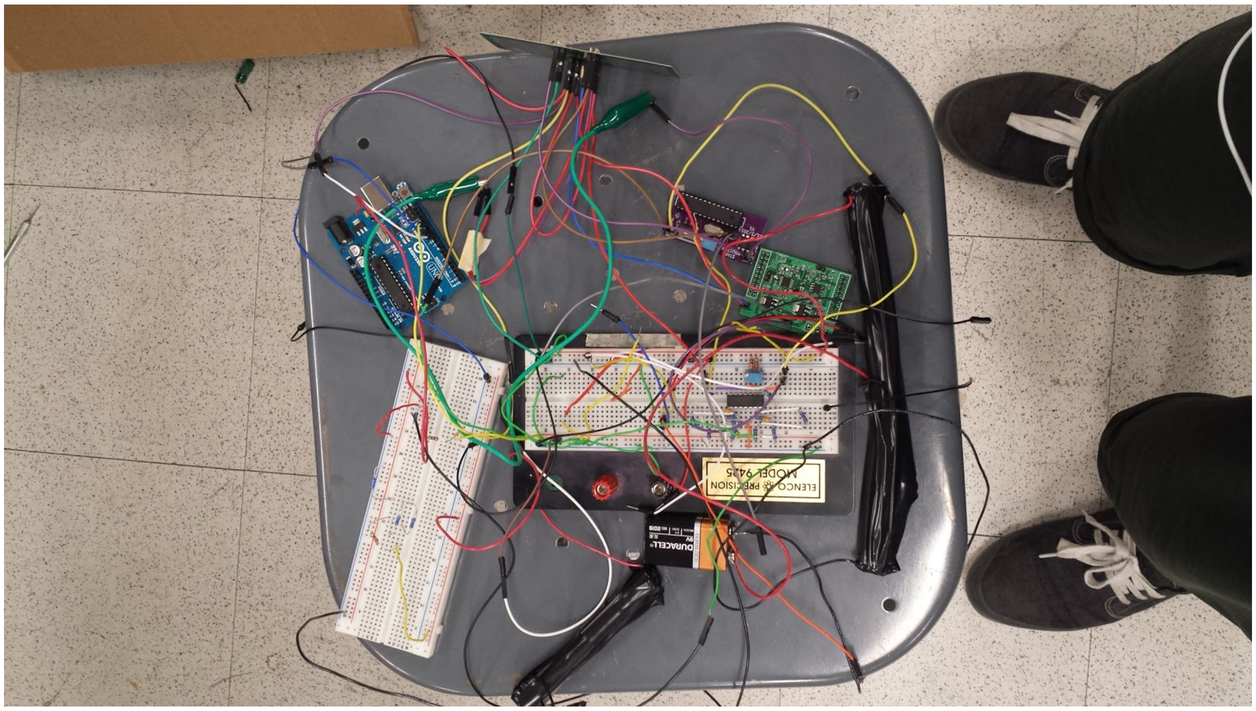
$$Vn = \sqrt{4kTB(R1 + R2)}$$

k = 1.380 x 10^-23J/K is Boltzmann's constant
T = absolute temperature in Kelvin = 300
B = the bandwidth in Hz = 2Ghz (difference between 24Ghz and 26Ghz, all other bands are attenuated by the filter).
R is the resistance in Ohm (R1 = 1000 Ohm, R2 = 200 Ohm)

Vn = 9.9679 x 10^-5 /Hz

Then I simply connected Vo into the 5V pin on the Arduino board in the section labeled POWER, and ground the arduino on the pin next to it labeled GND by connecting the GND pin to a common ground, in this case I used the ground on the Low Pass Filter.

D. Error Analysis (Jonathan)



Many issues arise when you use bare components instead of having all components on one PCB. We are using wires and breadboards to connect our components together. Wires have an inherent parasitic inductances associated with them while breadboards have an inherent parasitic capacitances. These parasitics play a huge role in the overall system. They cause insertion loss for the low pass filter, and gain reduction for the amplifier. These parasitic effects make it very difficult to have an accurate system. Each component may seem to work ideally when tested separately, but when you connect them together to become a system as a whole it will not function properly. We are operating at 24Ghz, rather than 2.4Ghz which is the normal radar operating frequency, and at such a high frequency having accurate device functionality is essential.

Our long range measurements are unlikely to be accurate because we are at such a high frequency. 2.4Ghz would be a great frequency for large range measurements, what we lose in range. However, what what we lose in range we gain in accuracy for short range measurements.

We decided not to use the built in low pass filter on the PCB because it was receiving too much noise. Having a bare Low Pass filter and connecting it with wires was not an efficient method because of the associated insertion loss.

Powering up our system with batteries is not efficient because batteries discharge. Every sequential test will run with a lower voltage, and our results will change from test to test, so there is a limited amount of time we can leave the system plugged in. Therefore, after each test we unplug the system, which gives more opportunities for human error, (if we bump a wire out of its slot).

E. Breadboard Active Low Pass filter (Christian)
   Assemble the circuit seen in the figure below

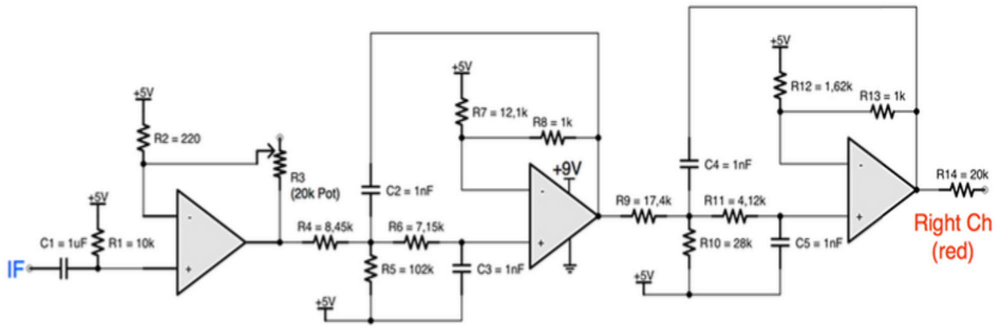
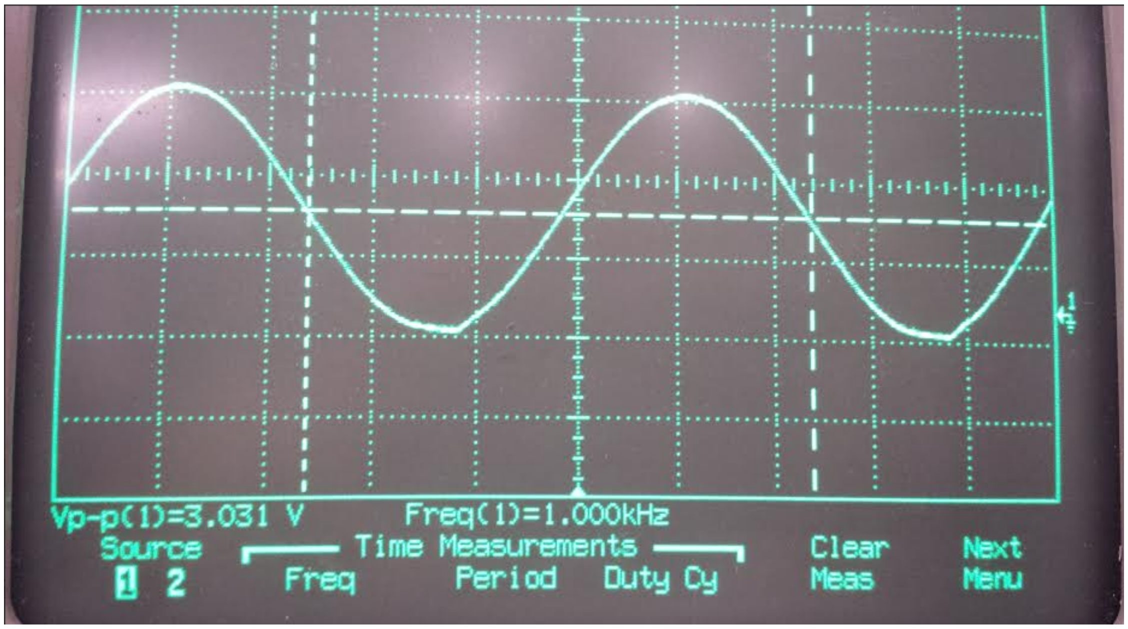


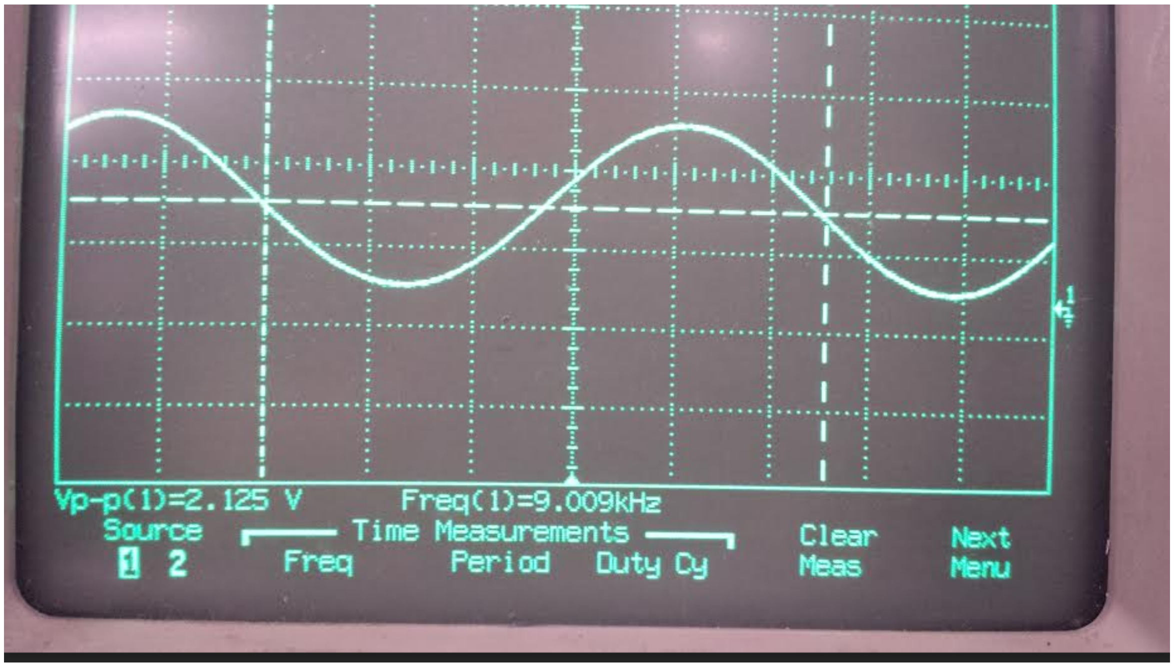
Fig. 6.10.1 Schematic of the active low-pass filter

Input from the function generator a sine wave signal with 100 mV Vpp at 1 KHz. Adjust to potentiometer so that the output voltage reads 3 Vpp.
Raise the frequency and analyze the output Vpp. When the output voltage reaches 0.707 of 3V, 2.121V, record the cutoff frequency. Below are figures of what the oscilloscope reading looked like during the testing stages of the LPF.

Oscilloscope reading of LPF output at 1 KHz and input Vpp = 100 mV



Oscilloscope reading of LPF output at 9 KHz,, output Vpp (f = 9 KHz) = 0.707*Vpp (f = 1 KHz)



Once the LPF is designed wired and tested apply the filter to the IF output of the infineon PCB. This filter will serve as the IF low pass filter of the mixer in the receive path of the infineon transceiver. When connected to the system the output of the filter will either be analyzed on an oscilloscope or the audacity Wav file software.

IV. Testing and Results:

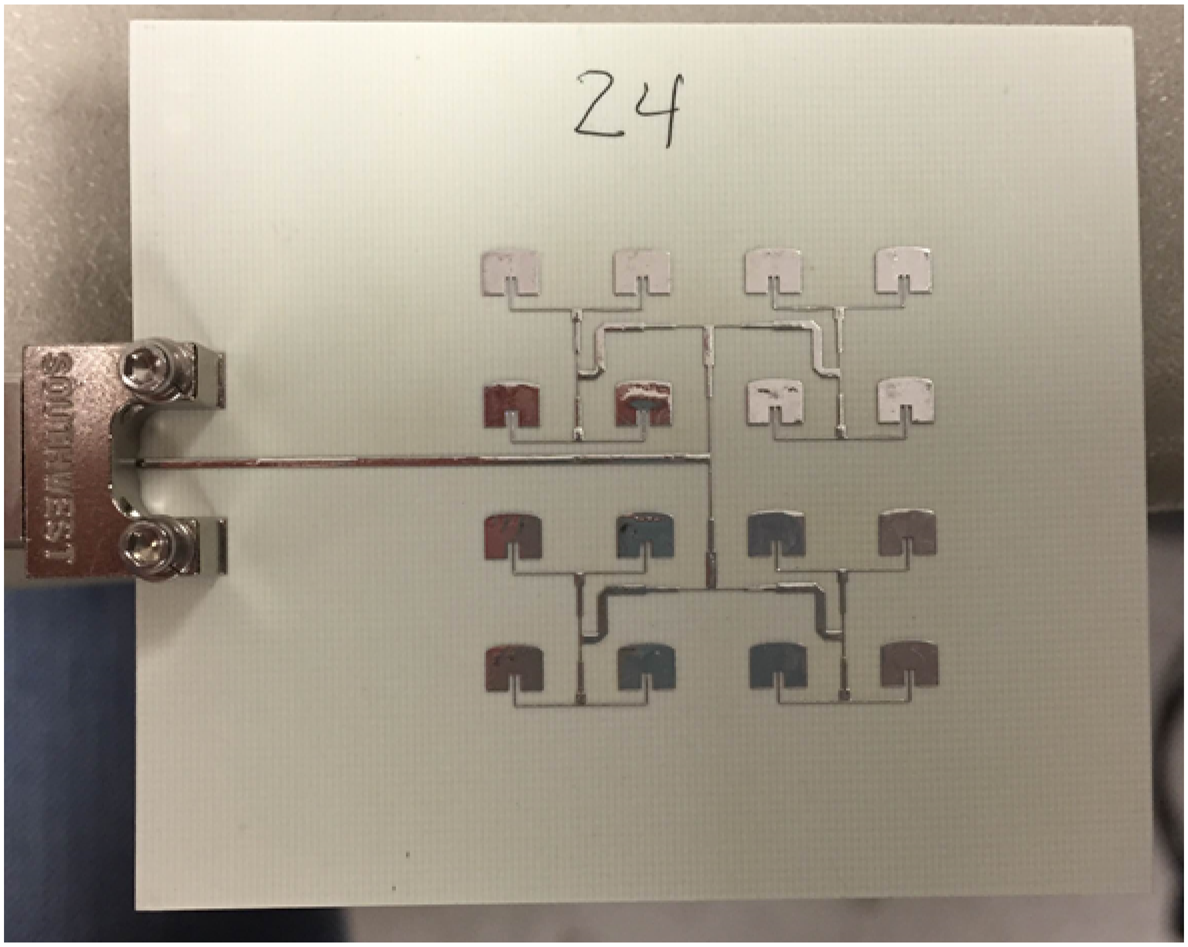
Part 1 Antenna Testing Results (Meijiao):

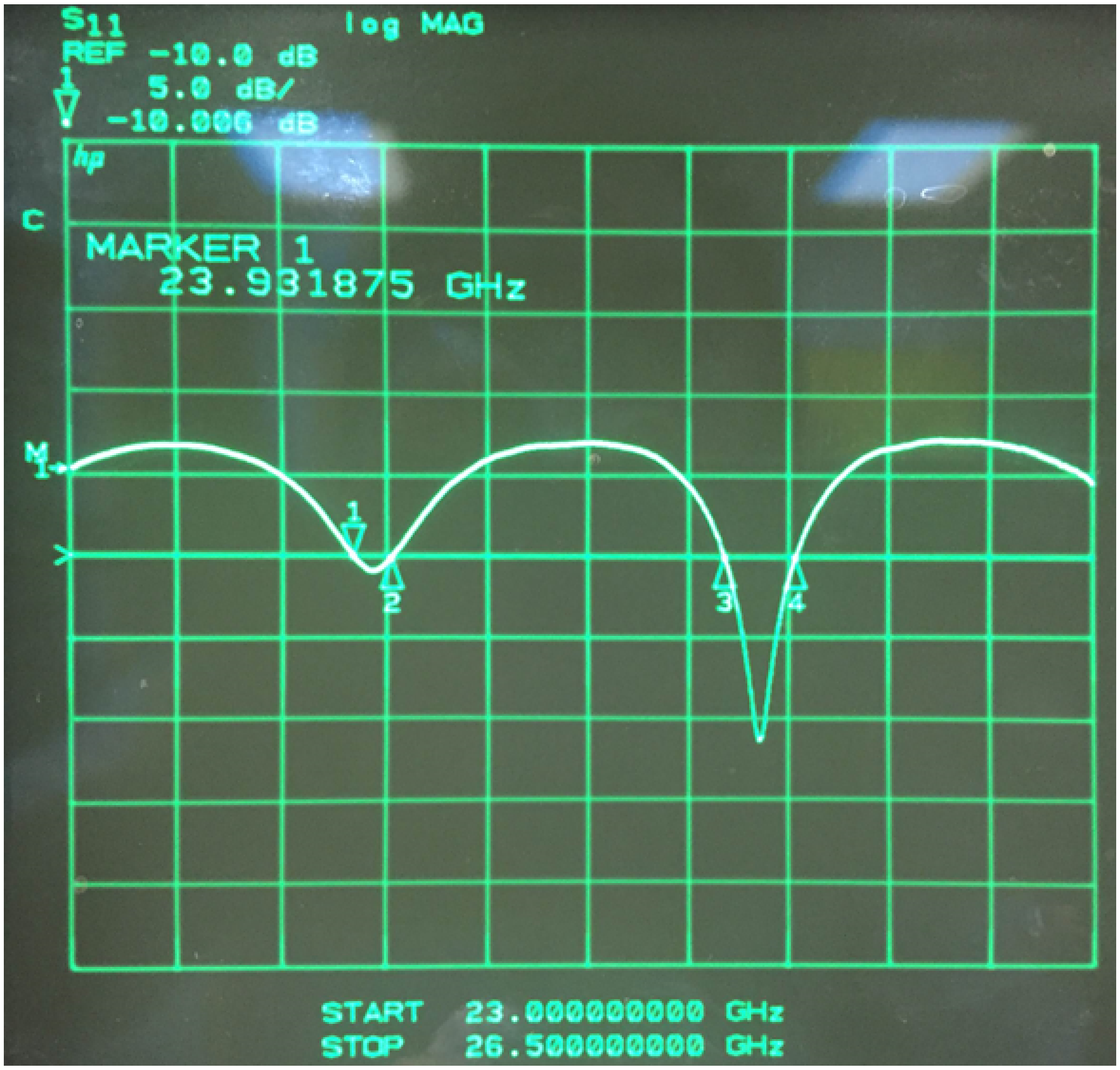
(1) Antenna Return Loss test

The antenna array return loss testing is very important in the whole radar system testing, because it decided what frequency will work in our transceiver and what voltage value we need to send to the VCO in the infineon.

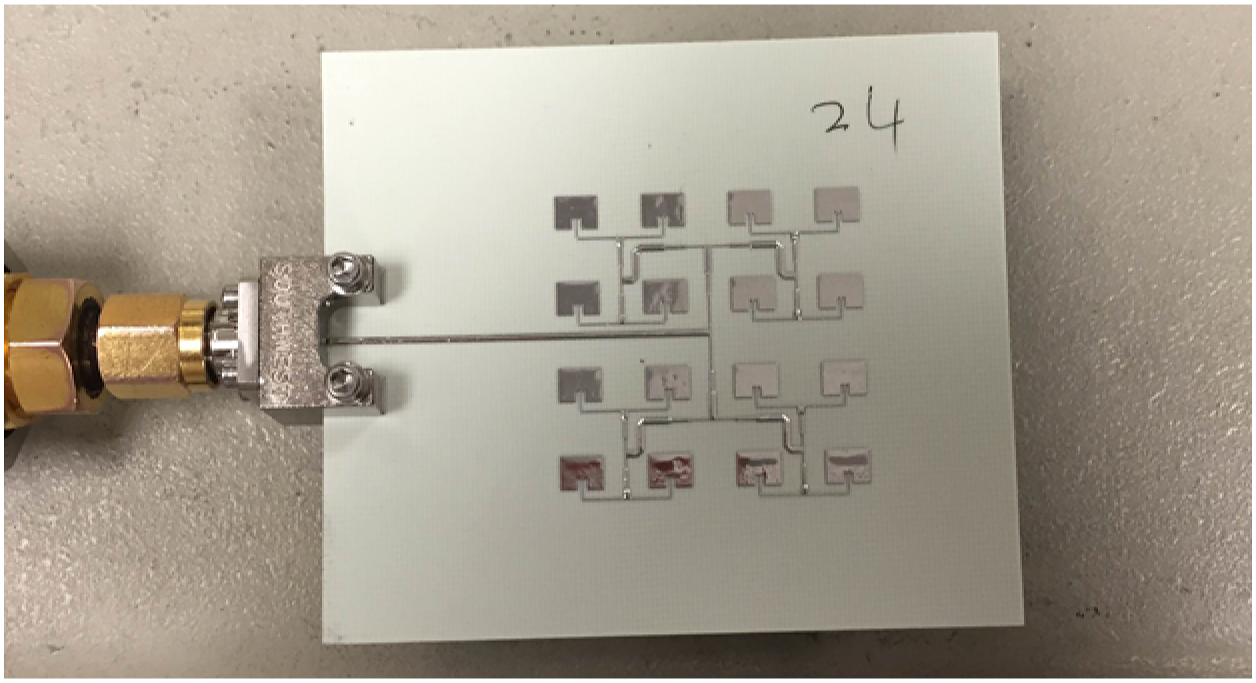
As we designed four arrays and fabricated them, we test all of them to decide which antenna we will use in our radar system

Array A:

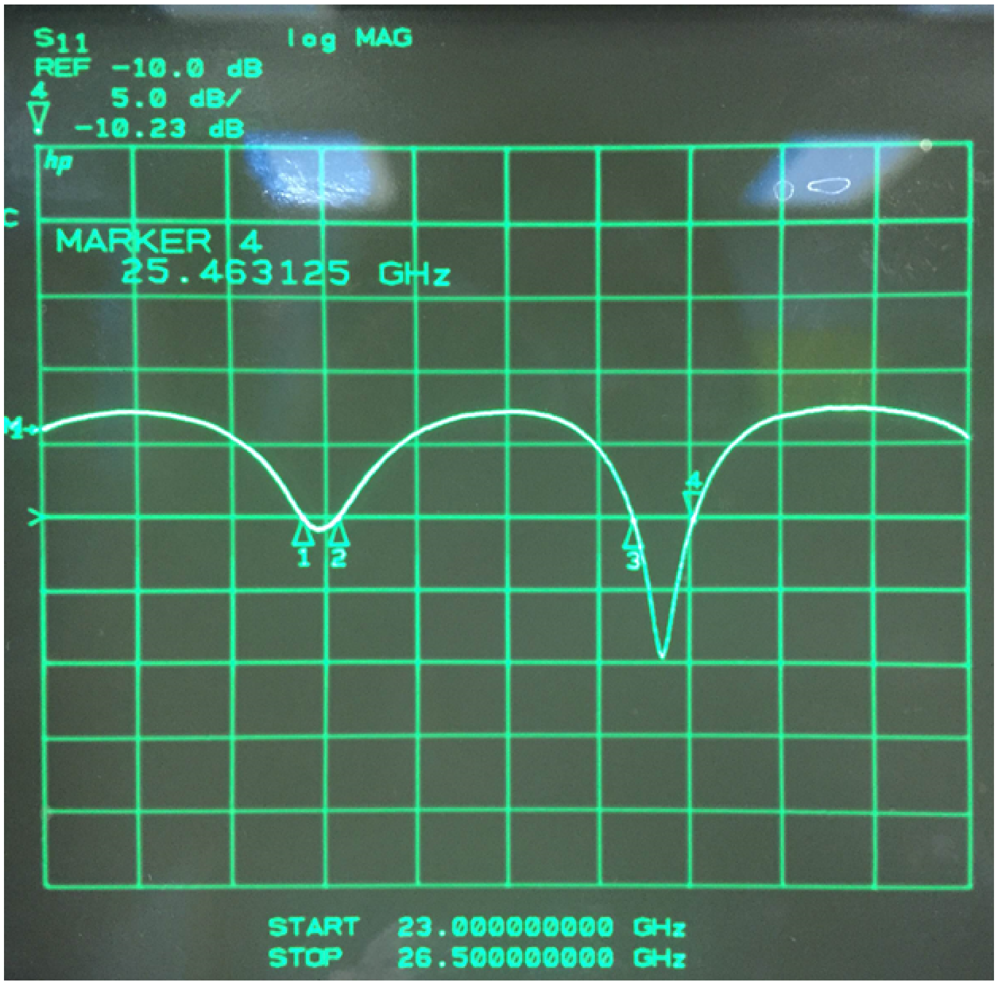



S11
log MAG
REF -10.0 dB
5.0 dB/
-10.006 dB
hp
C
MARKER 1
23.931875 GHz
M
1
1
2
3
4
START 23.000000000 GHz
STOP 26.500000000 GHz

Array B:

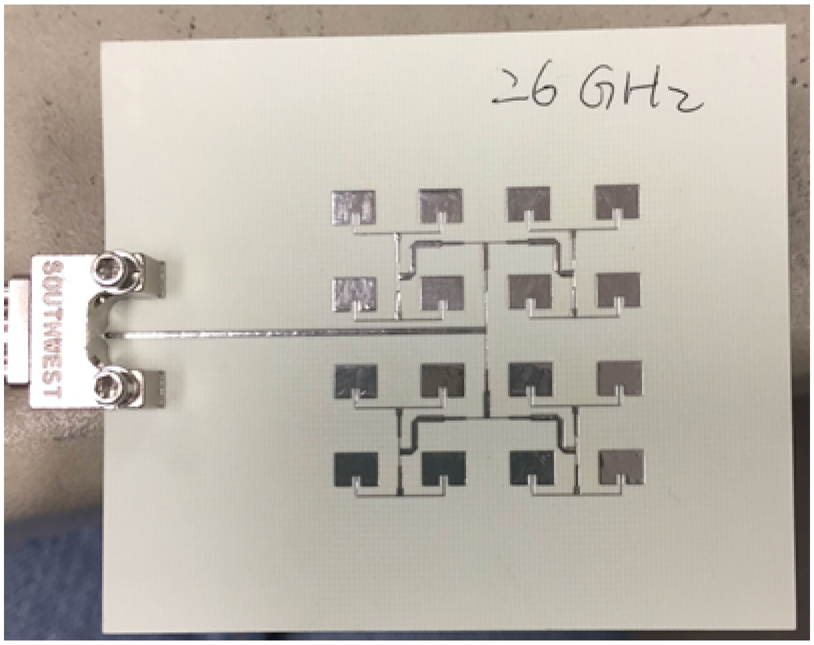


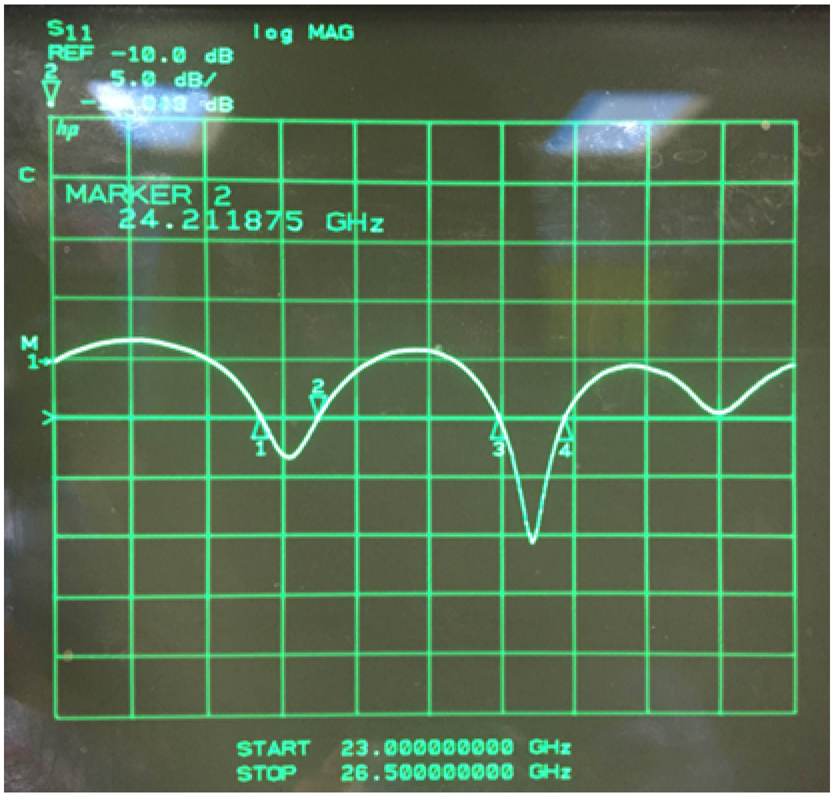
24
SOUTHWEST



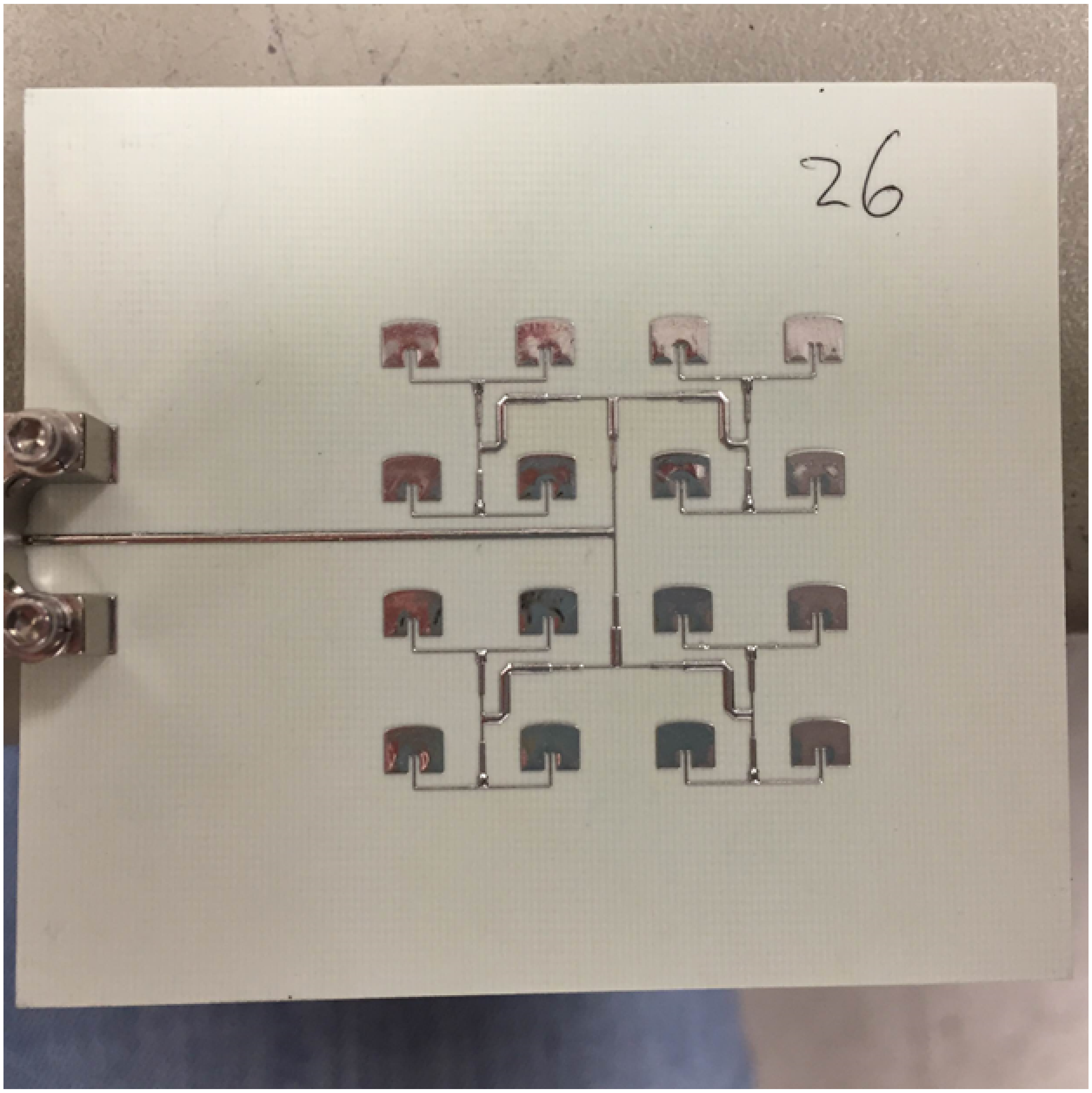
S11
log MAG
REF -10.0 dB
5.0 dB/
4
-10.23 dB
hp
MARKER 4
25.463125 GHz
1
2
3
4
START 23.000000000 GHz
STOP 26.500000000 GHz

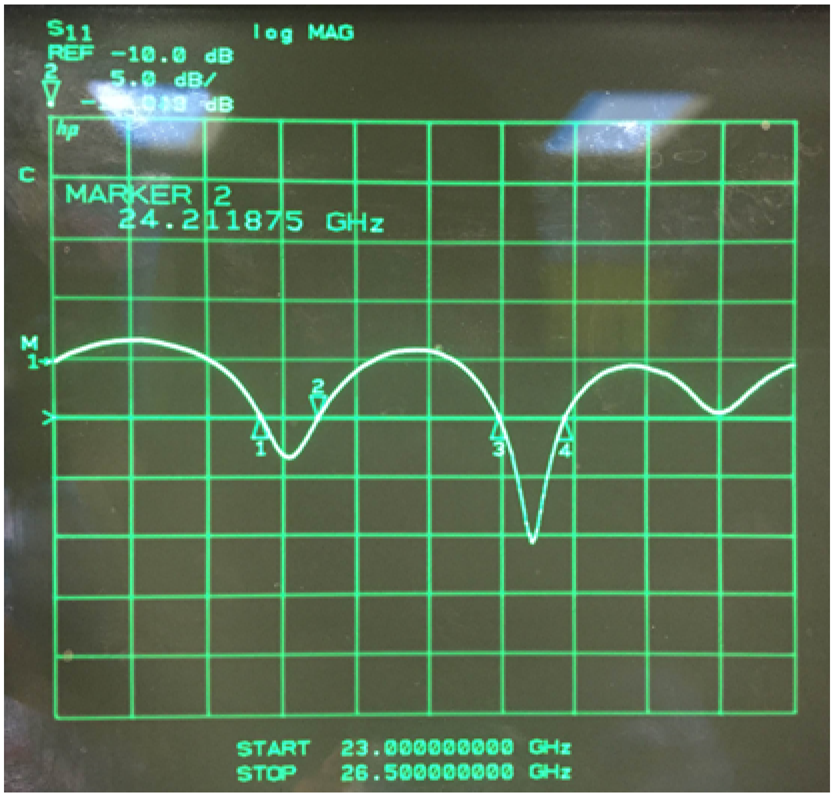
Array C:


26 GHz
SOUTHWEST


S11
log MAG
REF -10.0 dB
5.0 dB/
hp
C
MARKER 2
24.211875 GHz
M
1
2
1
3
4
START 23.000000000 GHz
STOP 26.500000000 GHz

Array D:

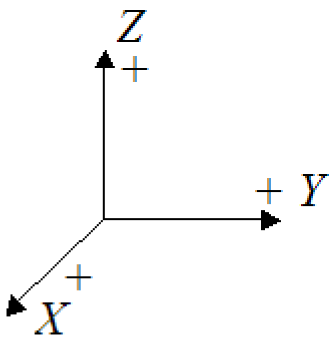




At last, we chosed the Array D for our radar as it has the best return loss and bandwidth.

Part 2) System Testing Results (Jonathan):

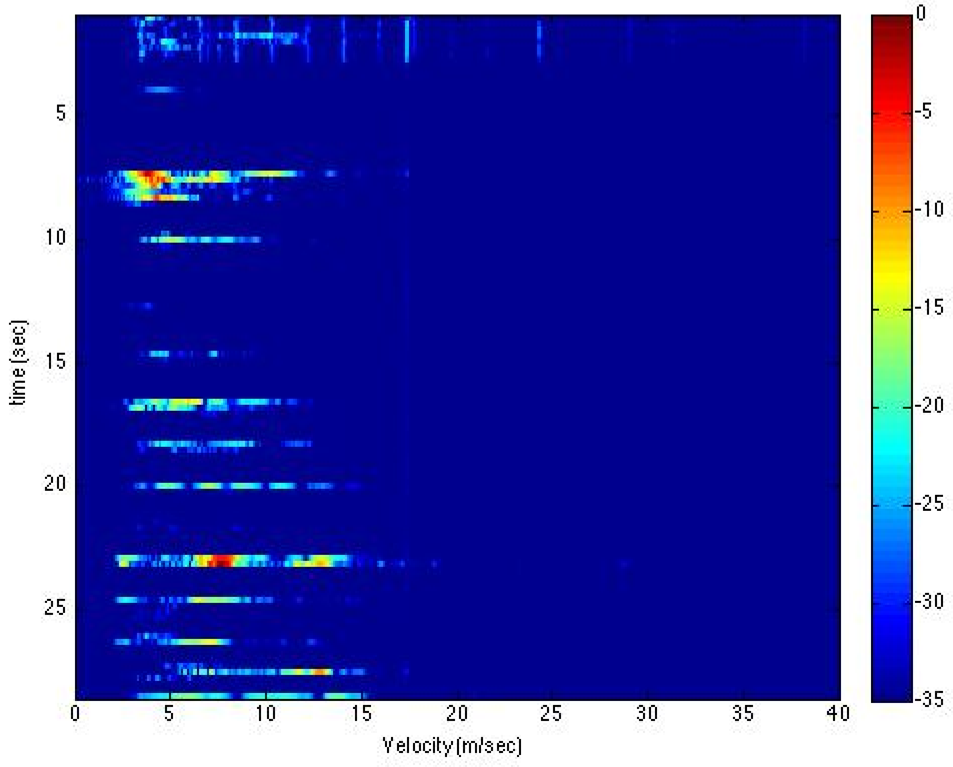
For simplicity in my testing explanations consider the following coordinate system.



The Antenna will be positioned in the ZY plane and taking measurements in the X direction.
In all of our experiments the test subject held a large metal sheet so it would reflect the incident wave. It is more effective that using our body.

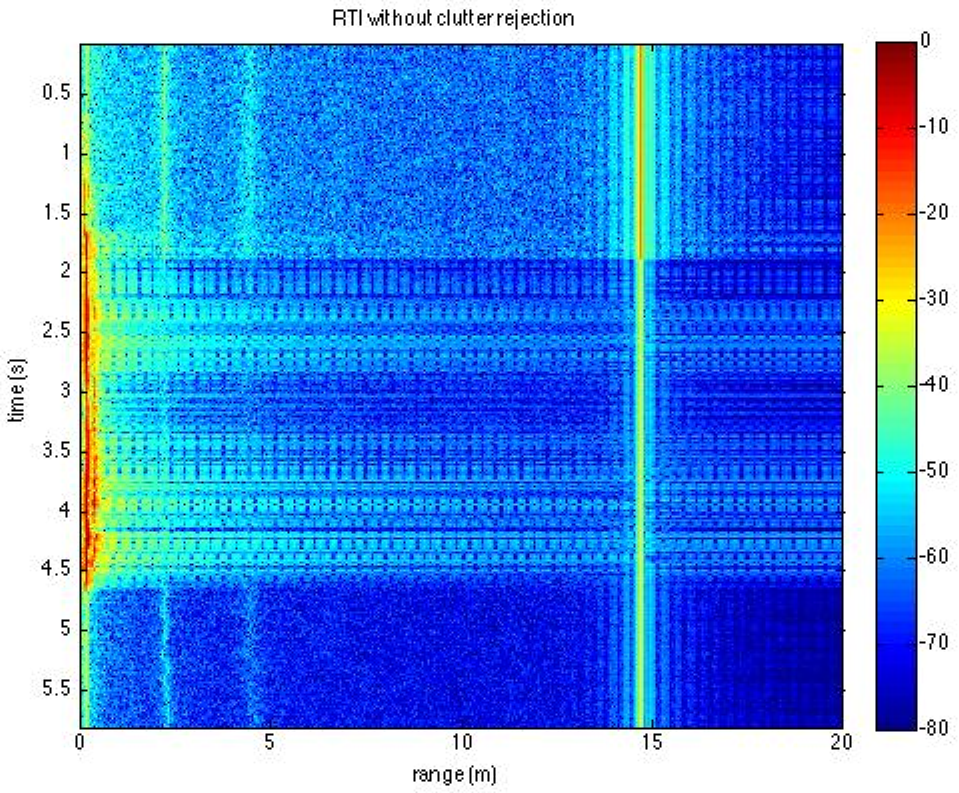
I will run you through a series of tests that we performed in order to attempt to gather the best possible data.

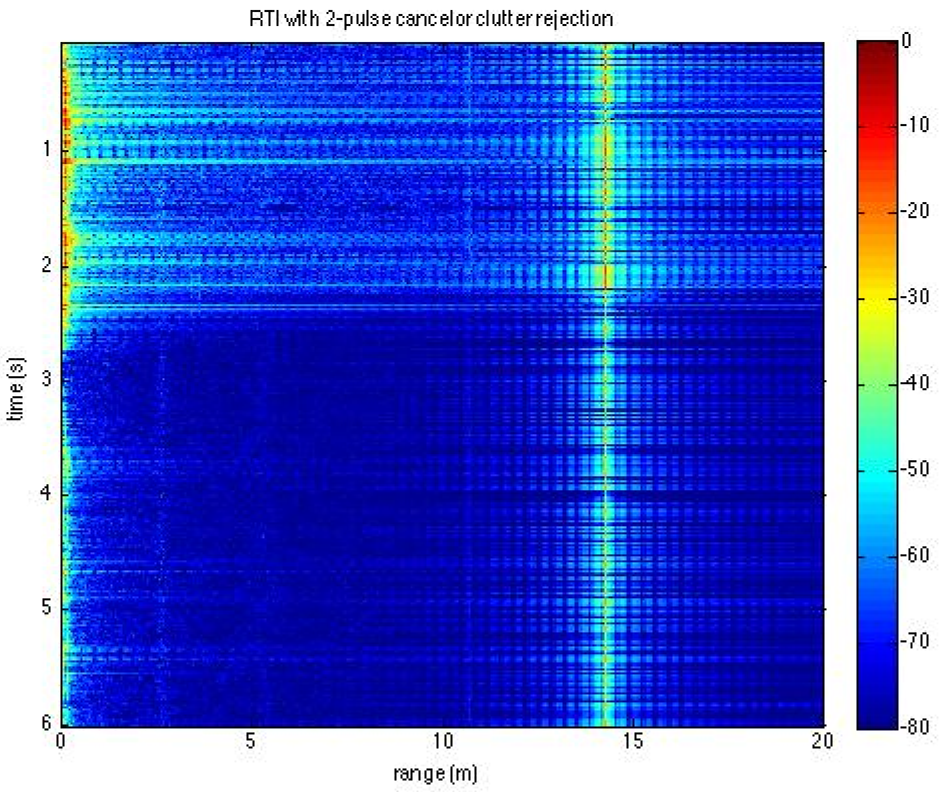
<u>2a) Test Method One (W/ Power Reduction Bits ON)</u>:
In Lab Testing:



This experiment was done in the lab room. In this experiment Christian held a large metal plate, at a distance of 118 inches, he walked in a straight line from the desk area of the lab to the yellow double doors in the lab. The antenna was angled pointing towards the chamber. Christian walked by the antenna a couple times at a distance x = 118 inches ~ 3m. He walked about 10 feet in 5 seconds which is 2 ft/sec or 0.6906 m/s.

<u>2b) Kemper Hallway Testing:</u>
<u>Range:</u>
Video: https://youtu.be/HwhnhoFssBY
We set two distances, the start distance and the stop distance.
Start distance: 16 ft = 4.5 meters
Stop distance: 9ft 7in ~ 3 meters
There is also the distance from the hallway to the end of the hallway, which happens to be 83 feet. = 25 meters. Christian walked for 4 seconds from the start distance to the stop distance, from 16ft to 9.6ft or 4.87 meters to 3 meters. We then used audacity in order to record the wav file, and then exported it and ran it on matlab using the code we obtained via the course github which is listed in appendix A.


RTI without clutter rejection
time (s)
range (m)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
0
5
10
15
20
0
-10
-20
-30
-40
-50
-60
-70
-80

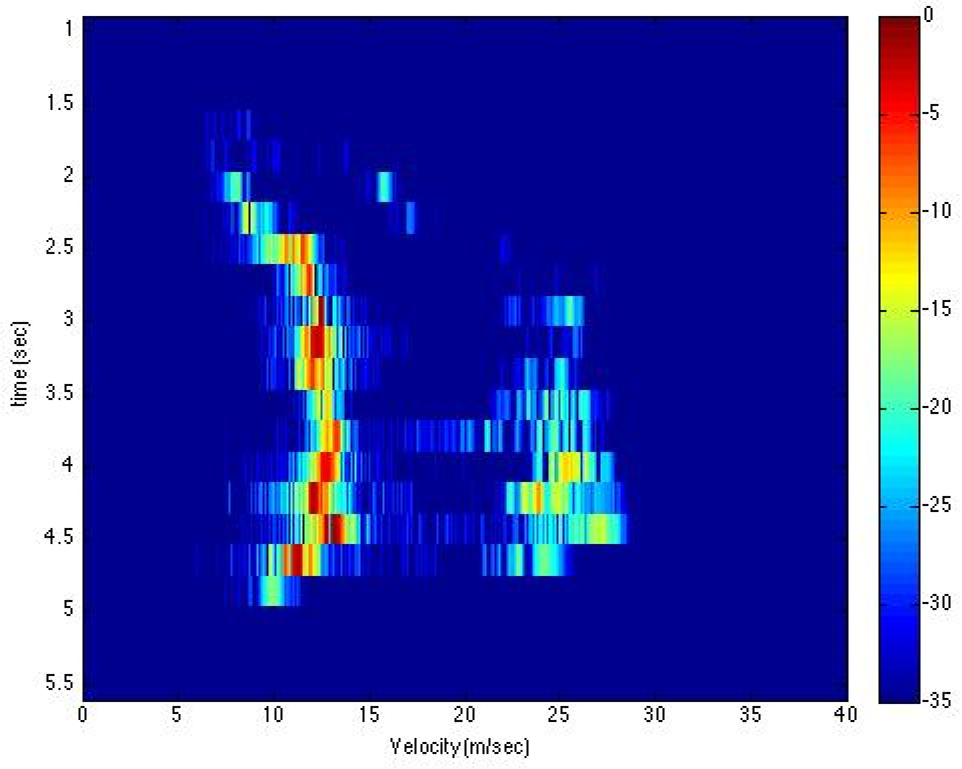

RTI with 2-pulse cancelor clutter rejection
time (s)
range (m)
1
2
3
4
5
6
0
5
10
15
20
0
-10
-20
-30
-40
-50
-60
-70
-80

In this figure we see one very solid line at ~14.5 meters (Line 3) another at ~5 meters (line 2), and one at ~3.5meters (line1). Line 1 corresponds to our stopping distance, Line 2 corresponds to our starting distance, and Line 3 corresponds to a unique distance.

So here is what I think happened. The distance from the point of measurement to the end of the hallway is 25 meters. However, after about 14 meters the signal starts to decay and is interrupted because it is bouncing off the the inner walls of the hallway. The point at which the signal starts to bounce off the walls is around 14 meters, therefore our range measurement is measuring ~14meters because that is what it views as the absolute distance.
We accurately measure the starting and stopping points in our controlled measurement. However, these points were observed at the same time intervals, meaning that the radar measured Line 1 and Line 2 at the same time, therefore our results were incorrect.

When performing this measurement we used the SPI code in appendix A, which turned on the power reduction bits. Error might be attributed to reflections from the inner walls of the kemper hallways.

Doppler:
Video link: https://youtu.be/MRX6Wb8iPaQ

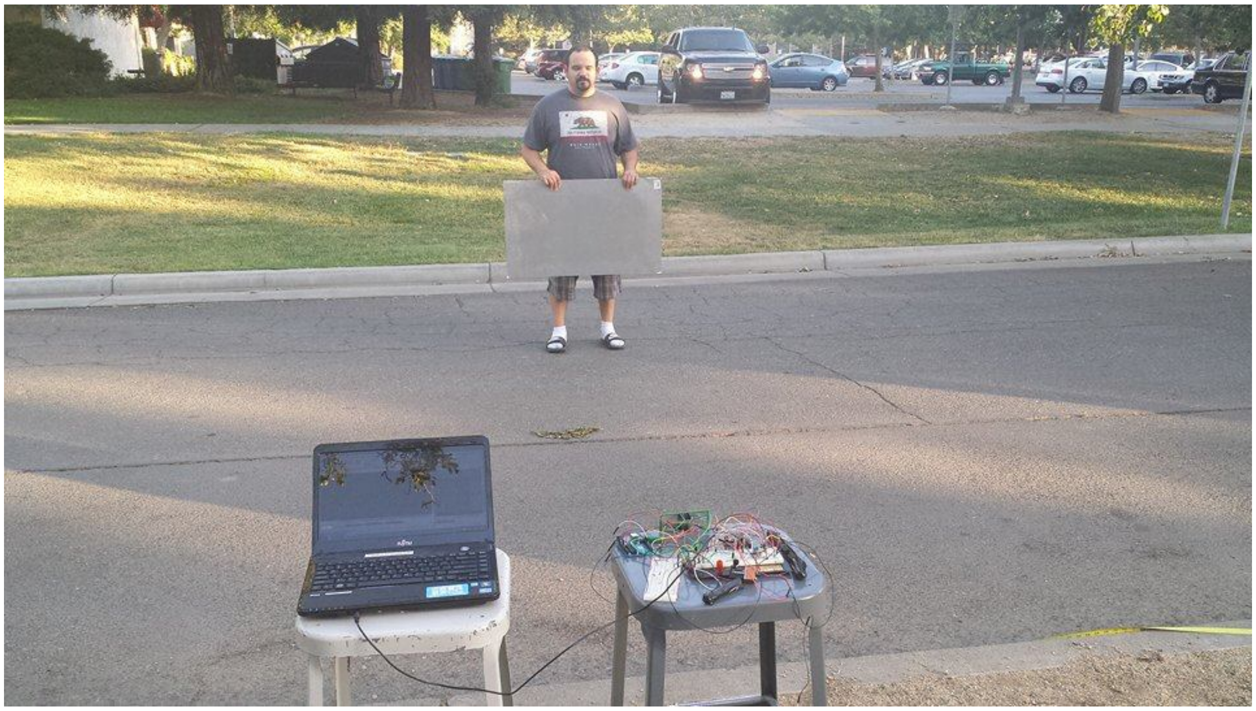


In this test, Christian started at a distance X = 15ft and walked to X=-2ft, passing by the Antenna. The point at which the antenna interacted in a direct line of sight with Christian was roughly 11 feet and the antenna was roughly at a 42 degree angle. Christian must have been walking around 4mph. Our results suggest the speed was around 11 meters/second, which corresponds to 24 mph. There is no way that christian actually walked that fast in the experiment. Something in our system must have been wrong, so we decided to test it again, this time turning off the power reduction bits in order to transmit higher power.

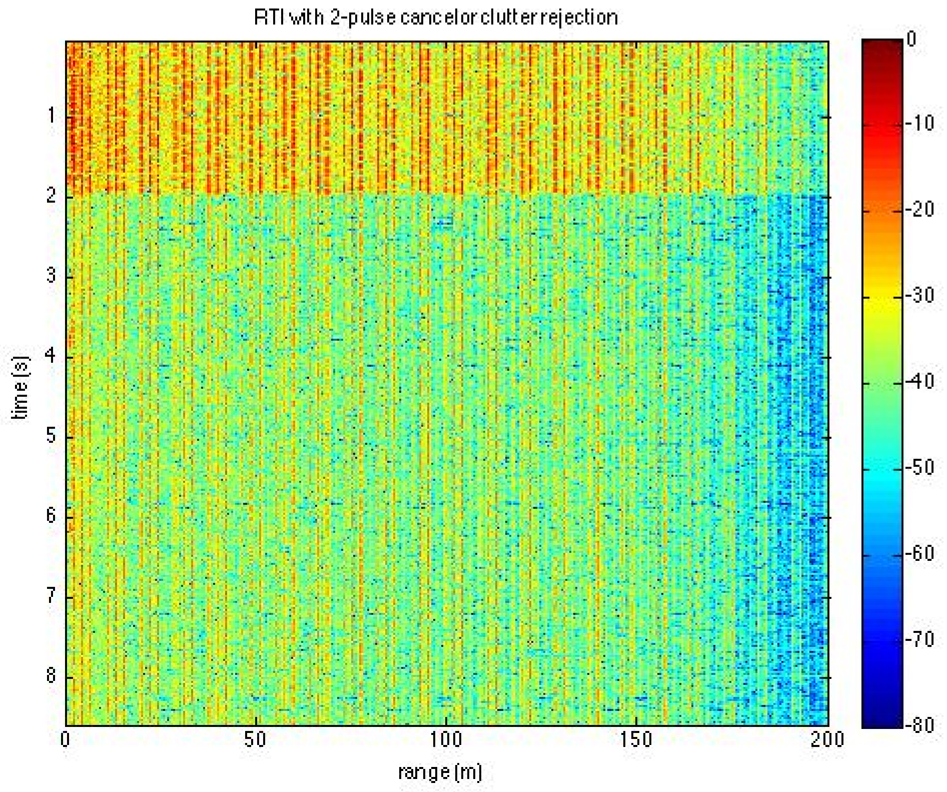
2C) Testing Method Two(Power Reduction Bits Off):

There is too much interference measuring indoors. The signal could be reflecting on the walls in the Kemper hallway. To gather better data, and put our system to the test Christian and I decided to go outside into an open area. We walked outside of Kemper past the courtyard to the bike path to perform further measurements. I decided to turn off the power reduction bits (bit 2,1 and 0) on the Infineon chip using the SPI code in Appendix B.

Range:

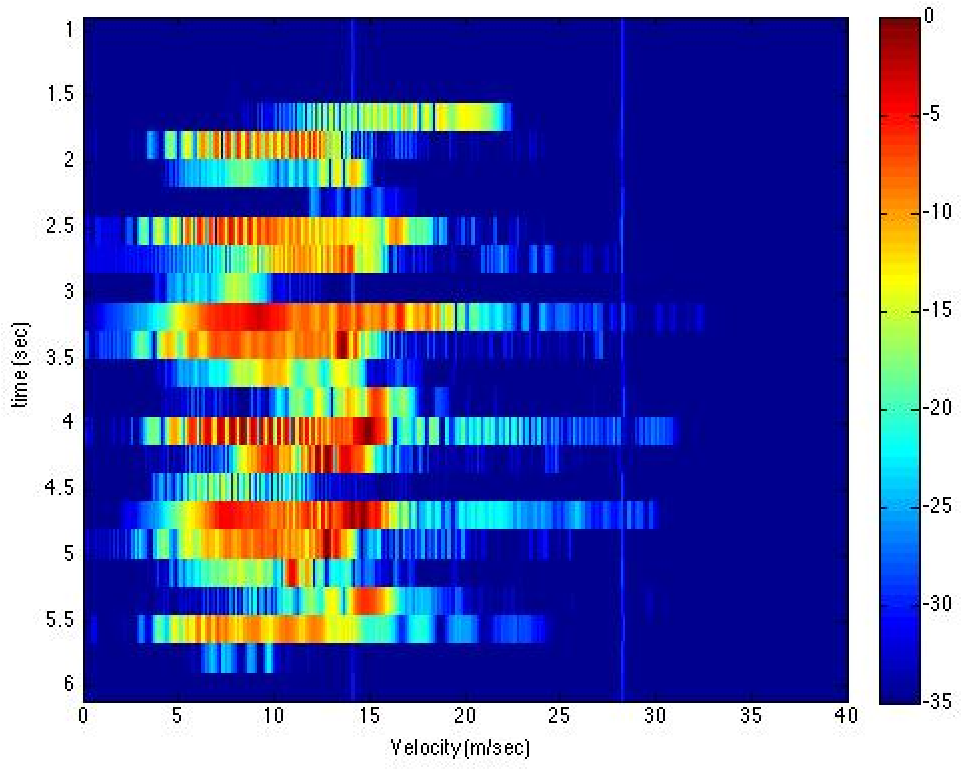


To do this we set up measured 15 feet from the Antenna and measured 10 feet from the antenna. Where Christian is standing in the image is 15 ft. (X1=15ft = 4.5m) and the branch on the ground is 10 feet (X2=5ft = 1.524m). Christian walked from X1 to X2, however our readings have so much noise that they are essentially unreadable. This is due to the fact that we are using a 24 Ghz transceiver. At such long ranges, our high frequency actually hinders our results, giving us no little to no accuracy. The inaccuracy of these results can be attributed to turning the power reduction bits off (bits 2-0) on the Power Amplifier. The power reduction bits actually reduce the noise from the PA, and by turning those bits off I have let the PA maintain more noise, therefore my overall system is carrying so much noise my results will be unreadable.



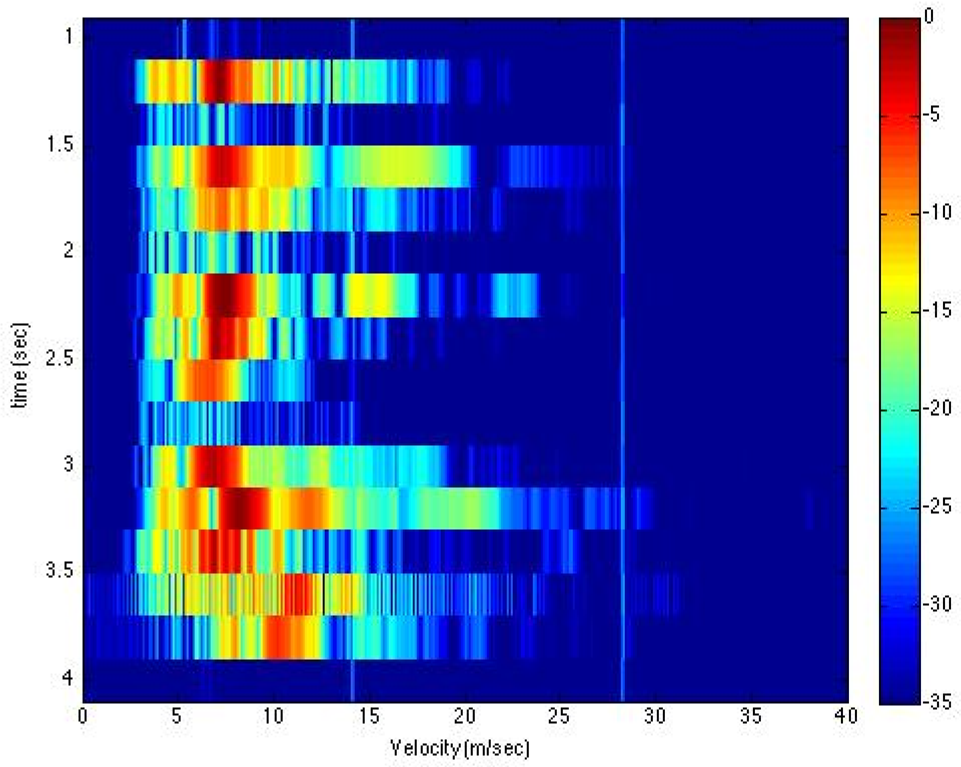
<u>Doppler:</u>

First off, here is what happens when we wave our hand in front of the Antenna:



The device is definitely measuring the change in speed, but when we increase the distance our measurement drops off dramatically. This distance was measured at x = 0.5m.

Experiment:
https://youtu.be/ZxgrkR4H6g4



Christian walked past the antenna at around 8 miler per hour, which is around 4 meters per second. This reading shows a much higher value than that. The error can be attributed to the low transmitted power of the radar, which results in short range accuracy, and possibly distortion in the system due to the mixer or other RF components. A possible cause of low power is the mismatch at the transmitted end with the antenna, as well as the the 50 ohm termination at the other differential output. Therefore, less than half of the power is actually transmitted. Another cause is that the SPI is not fully functional, and thus the transmitted output is actually the leakage power. As for the signal distortion, it can be because the transmitted signal and returned signals are at high frequencies (24-26 GHz range), so the difference of the frequency is low.

V. Conclusion

“Now the general who wins a battle makes many calculations in his temple ere the battle is fought.” - Sun Tzu (The Art of War)

The general that wins, is the one who follows his plan. We had an initial plan for how to build, design, and implement our system. That plan changed as we progressed through the process of design and implementation. We kept making changed during the process after gather results from testing, our strategy was to test and modify, and after confirming that a component worked, we moved onto the next one. We kept fixing little bits and pieces of the system as we progressed through the process but now looking back we should have come up with a full proof 0 to 100 plan, that took into account all possibilities for error and how we would treat those issues when we encounter them.

We decided not to use the low pass filter on James’ PCB because it had too much noise associated with it. When conducting a measurement of the radar system using the oscilloscope as a monitor, the distance radar showed a sine wave but with a lot of noise. The shape on the oscilloscope looked like a sine wave but the signal was distorted and blurry and kept bouncing. It was this point, we sought guidance from our instructor and ended up trying using the active LPF from fall quarter.

Midway through Winter quarter Christian and I were spending a lot of time trying to configure the Kentec display to process the signal in real time, but our efforts were needed elsewhere. There would be little point in having a working display if our radar system did not work. We had a group meeting and concluded that in light of the competition, we would gain more points for having a working system than getting the point divider for having the display unit. Team Stark reassigned our roles.

Our approach to making adjustments to the system was to “add more.” Thinking back on it, we should have attempted to reduce the amount of components in the system instead of adding them,this approach was not intentional and there was no strategy behind it. When we added more we simply me that we added some additional redundant wires that were only on the breadboard to make circuit connections a lot easier and more accessible. For example the PCB that had the 7 to 3.3 V voltage regulator and the PCB that had the triangle and SYNC functions both required >7 V Vcc supply. So for this we had a wire that came out of the node that powered the dual op amp chip used for the active LPF. This connection utilized another wire. Also when making connections between the active LPF and the voltage divider breadboards, a common ground had to be established which required another wire connection.

Our system ended up having so many wires going in so many directions because we kept adding to the system, we took an “add more approach” instead of a “reduce the system” approach.

We should have tried to make this project easier on ourselves by working with a lower frequency so our antenna would be able to measure values at further distances. At high frequencies we are prone to more noise, so minimizing noise is such an essential part of high frequency design. Our system does not work perfectly, and we can attribute that to the fact that we are operating at such a high frequency and have too much noise and signal distortion. If we operate at a lower frequency would be able to measure longer distances and noise would not have such a great effect. Higher frequency means a smaller wavelength. When an object is being measured from a radar system, if the radar is operating at higher frequencies, than covering the entire distance between the radar and the object of interest requires more wavelengths of the radar’s incident wave disturbance. This higher frequency and increased amount of wavelengths between the radar and the object, will lead to a better measurement resolution. If the system is not designed with the right amount of precision, then it is prone to more noise compared to a system that operates at a lower frequency (where the wavelengths of the incident wave between the radar and the object of interest are longer).

Our team is a cohesive unit, James designed the PCB and Meijiao designed the antenna. Christian designed the low pass filter and I configured the SPI communication. James and Meijiao contributed a lot towards the design phase during the first and second quarter and then Christian and I tested and debugged the system the entire third quarter to make sure it works. At every point during the project, someone was working on a task, there was no dead time. We were in constant communication with one another, working together, solving problems for each other and discussing concepts related to the project. We learned so much about electronics not only from each other but also asking questions to graduate students in the third floor lab. At times we came to a point where we did not know what to do, thankfully professors and graduate students were willing to lend a hand and provide insight to further our knowledge on specific issues so that we could push through those barriers. Graduate students and professors extended so much help to us throughout this entire process. There is a cohesive nature in the engineering department that fosters learning. When we needed help it was there, willing and able. Team Stark Industries was a success in that we learned how to efficiently work and manage our project together.

Special thanks to:
Dr. Xiaoguang Leo Liu
Dr. Lance Halsted

Naimul Hasan
Jeff Tan
Dr. Bernard Levy
Dr. Rick Branner

APPENDIX:

Appendix A: SPI Arduino code to turn on Power Amplifier and turn on Power reduction bits, less power

```
/*
Jonathan Mitchell
Team Stark Industries
SPI Data Transfer to turn on PA on the Infineon BGT24MTR12

The circuit
* On the Infineon
* MOSI = pin 20
* CLK = pin 19
* ChipSelect = pin 18
* MISO - where you receive from slave

On the Arduino
* ChipSelect or Slave Select = digital pin 10 (CS or SS pin)
* MOSI = digital pin 11 (MOSI pin)
* CLK = digital pin 13 (SCK pin)

B - connect this to ground
*/
#include <SPI.h>
```

```
const int slaveSelectPin = 10;

void setup() {
  pinMode (slaveSelectPin, OUTPUT); //configure pin to behave as output pin
  //initialize SPI
    SPI.setClockDivider(SPI_CLOCK_DIV2); //clock set to 8Mhz
  SPI.begin(); //Initializes the SPI bus by setting SCK, MOSI, and SS to outputs, pulling SCK and
MOSI low, and SS high
  SPI.setBitOrder(MSBFIRST);
}


void loop()
{
  digitalWrite(10,LOW); //low means 0V
  SPI.transfer(0b00100000); //transfers one byte over SPI bus, sending and receiving
  //this part sends in the first half of the bits to the register, bits 15-8
  SPI.transfer(0b00000111); //this part sends in the second half of the bits to the register as low,
bits 7-0  //clear the register, send 0's through
  //digitalWrite(10,HIGH); //new
  digitalWrite(10,HIGH);

}
```

Appendix B: SPI code to turn Power Amplifier On and turn off Power reduction bits, so we have more power:

```
/*
Jonathan Mitchell
Team Stark Industries
SPI Data Transfer to turn on PA on the Infineon BGT24MTR12

The circuit
* On the Infineon
* MOSI = pin 20
* CLK = pin 19
* ChipSelect = pin 18
* MISO - where you receive from slave
```

```
On the Arduino
* ChipSelect or Slave Select = digital pin 10 (CS or SS pin)
* MOSI = digital pin 11 (MOSI pin)
* CLK = digital pin 13 (SCK pin)

B - connect this to ground
*/
#include <SPI.h>

const int slaveSelectPin = 10;

void setup() {
  pinMode (slaveSelectPin, OUTPUT); //configure pin to behave as output pin
  //initialize SPI
    SPI.setClockDivider(SPI_CLOCK_DIV2); //clock set to 8Mhz
  SPI.begin(); //Initializes the SPI bus by setting SCK, MOSI, and SS to outputs, pulling SCK and
MOSI low, and SS high
  SPI.setBitOrder(MSBFIRST);
}


void loop()
{
  digitalWrite(10,LOW); //low means 0V
  SPI.transfer(0b00100000); //transfers one byte over SPI bus, sending and receiving
  //this part sends in the first half of the bits to the register, bits 15-8
  SPI.transfer(0b00000000); //this part sends in the second half of the bits to the register as low,
bits 7-0 turning off the power power reduction so we have more power output.
  //digitalWrite(10,HIGH); //new
  digitalWrite(10,HIGH);

}
```

Appendix C

Ardurino code for 100 Hz triangle wave:

```
/*
Triangle wave and sync pulse generator to control a (0-5V input range)
VCO for FMCW radar.
The MPC4921 DAC is used to generate a triangle wave with a period of
40ms.
PWM of the Arduino UNO is use to simultaneously generate the sync
pulse,
used for signal processing.
*/

#include "SPI.h" // SPI library for communicating with the DAC
byte data = 0;// A byte is an 8-bit number
word outputValue = 0;// A word is a 16-bit number

void setup()
{
   // Set pins for input and output
   pinMode(8, OUTPUT);                    // SYNC pin
   pinMode(10, OUTPUT);                   // Slave-select (SS) pin
   SPI.setClockDivider(SPI_CLOCK_DIV2);    // Set the SPI clock to 8MHz
   SPI.begin();                           // Activate the SPI bus
   SPI.setBitOrder(MSBFIRST);             // Most significant bit first for
MCP4921
}

void loop()
{
   digitalWrite(8, HIGH);             // SYNC pulse high

   // Rising edge of the triangle wave
   for (int a = 1715; a <= 1960; a=a+1)
   {
      outputValue = a;
```

```
        digitalWrite(10, LOW);          // Activate the the SPI transmission
        data =highByte(outputValue);    // Take the upper byte
        data = 0b00001111 & data;       // Shift in the four upper bits (12 bit
total)
        data = 0b00110000 | data;       // Keep the Gain at 1 and the
Shutdown(active low) pin off
         SPI.transfer(data);            // Send the upper byte
        data =lowByte(outputValue);     // Shift in the 8 lower bits
        SPI.transfer(data);             // Send the lower byte
        digitalWrite(10, HIGH);         // Turn off the SPI transmission
    }

    digitalWrite(8, LOW);               // Sync pulse low

    // Falling edge of the triangle wave, very similar as above
    for (int a = 1960; a >= 1716; a = a-1)
    {
        outputValue = a;
        digitalWrite(10, LOW);
        data =highByte(outputValue);
        data = 0b00001111 & data;
        data = 0b00110000 | data;
        SPI.transfer(data);
        data =lowByte(outputValue);
        SPI.transfer(data);
        digitalWrite(10, HIGH);
    }
}
```

Ardurino code for triangle wave at 16 Hz

```
/*
```

```
Triangle wave and sync pulse generator to control a (0-5V input range)
VCO for FMCW radar.
The MPC4921 DAC is used to generate a triangle wave with a period of
40ms.
PWM of the Arduino UNO is use to simultaneously generate the sync
pulse,
used for signal processing.
*/

#include "SPI.h" // SPI library for communicating with the DAC
byte data = 0;// A byte is an 8-bit number
word outputValue = 0;// A word is a 16-bit number

void setup()
{
  // Set pins for input and output
  pinMode(8, OUTPUT);                    // SYNC pin
  pinMode(10, OUTPUT);                   // Slave-select (SS) pin
  SPI.setClockDivider(SPI_CLOCK_DIV2);   // Set the SPI clock to 8MHz
  SPI.begin();                           // Activate the SPI bus
  SPI.setBitOrder(MSBFIRST);             // Most significant bit first for
MCP4921
}

void loop()
{
  digitalWrite(8, HIGH);              // SYNC pulse high

  // Rising edge of the triangle wave
  for (float a = 1715; a <= 1960; a=a+0.25) // int-->float and
                                           //a=a+1→a=a+0.25
  {
    outputValue = a;
```

```
        digitalWrite(10, LOW);          // Activate the the SPI transmission
        data =highByte(outputValue);    // Take the upper byte
        data = 0b00001111 & data;       // Shift in the four upper bits (12 bit
total)
        data = 0b00110000 | data;       // Keep the Gain at 1 and the
Shutdown(active low) pin off
         SPI.transfer(data);            // Send the upper byte
        data =lowByte(outputValue);     // Shift in the 8 lower bits
        SPI.transfer(data);             // Send the lower byte
        digitalWrite(10, HIGH);         // Turn off the SPI transmission
    }

    digitalWrite(8, LOW);               // Sync pulse low

    // Falling edge of the triangle wave, very similar as above
    for (float a = 1960; a >= 1716; a = a-0.25) /// int-->float and
                                                    //a=a+1→a=a-0.25

    {
        outputValue = a;
        digitalWrite(10, LOW);
        data =highByte(outputValue);
        data = 0b00001111 & data;
        data = 0b00110000 | data;
        SPI.transfer(data);
        data =lowByte(outputValue);
        SPI.transfer(data);
        digitalWrite(10, HIGH);
    }
}
```