Undergraduate Project Report
2017/18

Development of EM Signal Measurement System

Name:	Yinuo Wang
Programme:	Telecommunications with Management
Class:	2014215107
QM Student No.	140920440
BUPT Student No.	2014213051

Date 06/05/2018

Table of Contents

Abstract

Development of an Electro magnitude measurement system can help determining the characterization of EM circuits by accurately measuring the difference of phases and the difference of amplitudes of two signals up to 2.7GHz. By comparation of the reference signal and the signal reflected by device under test (DUT), the features of DUTs can be detected. My work is to improve the existing hardware system and build the software, implementing a functional EM signal measurement system. To complete the project, first, I finished literature review, referring to materials about the existing system and the improved measurement method which works with two AD8302 chips and a 90^0's phase shift of one signal to improve the measurement of phase difference. I also read data books of AD8302, 3dB hybrid and ADF4350. Based on my research, I developed a new prototype with devices including two AD8302 chips and a 3dB 90^0 hybrid. I tested the prototype to make sure that each device works well and a 90^0's phase shift can be achieved by the hybrid. Second, I improved the frequency generation process, programming a smaller and more efficient STC8051 microcontroller which controls the ADF4350 PLL synthesizer to generate frequency up to 4.4GHz. Third, to make the whole system easy to use, I designed a GUI based on C++ which allows users to set the frequency and receive results on the computer directly. Last, I validated and calibrated the system, making the software cooperate well with the hardware. The EM signal measurement system can output phase difference around 90^0 and a magnitude ratio around 0dB with no DUTs, which satisfies the requirement of the improved method. It can also be used to detect some basic EM devices.

Key words: implementation, EM measurement, signal

Development of EM Signal Measurement System

摘要 (Chinese translation of the Abstract)

电磁测量系统的开发能够通过准确测量高达 2.7GHz 的两路信号的相位差和幅度差来帮助确定电磁电路的特性。待测设备的特性可以通过比较基准信号和反射信号来测量。我的工作是改进之前已有的硬件系统并开发软件，实现一个功能完整的电磁信号测量系统。为了实现这个项目，首先，我查阅参考了之前系统以及改进方法的材料，改进的方法需要用两个 AD830 芯片以及对一路信号的 90 度相移来实现。我还参阅了 AD8302,3dB 电桥和 ADF4350 的数据手册。根据我的研究，我搭建了一个包含两个 AD8302 芯片和一个 3dB 电桥的硬件原型。我对原型进行了测试以保证每个元件能够正常工作并且 3dB 电桥能够进行 90 度的相位移动。其次，我改进了频率生成装置。我对 STC8051 系列的单片机进行编程来控制 ADF4350 频率发生器，它的频率能够达到 4.4GHz。再次，为了使整个系统便于使用，我基于 C++语言设计了一个软件。通过这个软件，用户能够直接在电脑上控制频率生成并接收测量数据。最后我完成了整个系统并对系统进行了校准，使软件和硬件能够协同工作。在没有接待测设备时，系统能够输出约 90 度相位差和约 0dB 的比率。这点满足改进方法的要求。该系统能够用于基本电磁元件的测量。

关键词：实现，电磁测量，信号

Chapter 1: Introduction

In this report, I will discuss the project in detail including the background of the existing EM measurement system, its design and implementation, the results and future work. The background mainly introduces the improved measurement method and the principle of key devices which are ADF4350, STC8051 microcontroller, AD8302 detector, 3dB power splitter and 3dB hybrid. In the implementation part, the connection of devices will be introduced. Each key step will be discussed which is related to hardware implementation and GUI design. Besides, the simple simulation of low-pass filters is shown in this part. For the Results and Discussion part, the measurement results and calibration results will be displayed.

1.1 Description of the project

The EM signal measurement system is developed to determine the features of EM circuits. A key method to achieve the system is accurately measuring the difference of phases and the difference of amplitudes of two signals up to 2.7GHz. My work is to improve the previous hardware system and build the software, implementing a complete functional EM measurement system.

The existing given system can approximately detect voltages and phases across two ports by providing a certain voltage output, without determining the direction of the phase difference. To improve the system, I use a new method, working with two AD8302 chips and a 90^0's phase shift, to identify whether the phase difference is a negative number or a positive one. Besides, to generate the signal of frequency up to 2.7GHz, I find a suitable frequency generation method (chip) and implement the method to provide wideband sinusoidal signal.

As for the software, I design a GUI to make users easily to handle the measurement and get the results directly. The GUI is a C++ program under the environment of VS 2017, which mainly has two functions. The first one is to set a computer port and generate the frequency by entering the value to the software. The second one is to receive the data sent from a certain port and display the phase differences and the amplitude differences on the screen. I also develop a Arduino program to receive data from the AD8302 chips.

To make the measurement accurately, I will make calibrations of the hardware system. In the end, the system should be an implemented one, and the software and hardware need to work harmoniously with each other to get the correct result of phase difference and amplitude difference of the two signals. Different EM circuits and devices will be needed to detecting by the system to test its accuracy.

1.2 Main tasks and work achieved

There are mainly 4 tasks in my project.

Task 1 is validation and improvement on existing measurement methods (based on AD8302).

1. Literature review
2. Finding suitable components to build an initial prototype
3. Implement the hardware circuit
4. Test the accuracy of the measurement

To achieved the task, I read papers about AD8302, ADF4350, 3dB Hybrid and the existing EM measurement system. I used two AD8302 chips, a 3dB splitters and a 3dB 900 hybrid to build an initial prototype. I have detected the phase difference and amplitude of signals from 800MHz to 2500MHz. Besides, to make sure that the 3dB splitters and the hybrid work well, I detected the two devices and the output was correct within the allowable range of error.

Task 2 is to design a circuit to provide sinusoidal signals across a band of frequencies.

1. To find a suitable frequency generation method (chip)
2. To implement the method to provide wideband sinusoidal signal
3. To test the method and validate it

For task2, I tried to use an Arduino board to control the ADF4350 to generate the frequency. However, the Arduino board is not fast enough. To improve the efficiency and make the system more portable, I found a development suit with a STC8051 microcontroller (STC15F2K60S2) and an ADF4350 which can generate frequency between 137.5MHz to 4.4GHz. Then, I read papers about ADF4350 and STC8051 chip. I programmed the suit under the Keil4 environment to generate frequency up to 2.7GHz.

Task 3 is to develop a software interface to control drive the frequency generation process and to detect signals.

1. Design a suitable graphical user interface (GUI)

2. Write the code to read data from computer port

3. Write the code to control the frequency of sinusoidal signal

4. Implement calibration code for the measurement setup

I design the GUI using C++. According to the bandwidth limits of hardware, this software allows users to set the frequency between 800MHz to 2700MHz, which sends data to a certain USB port connected to the ADF4350 development suit mentioned above. This software can also receive the data sent from a certain USB port by a microcontroller and display the phase differences and the amplitude differences on the screen.

Task 4 is to validation the final design.

1. Integrate the software with hardware

2. Test the system

In the task 4, I mainly connected the hardware and the software to make the whole system work well. To make the measurement more accurate, I also tested the devices separately and make calibrations. A model of the completed EM signal measurement system is shown as follows.

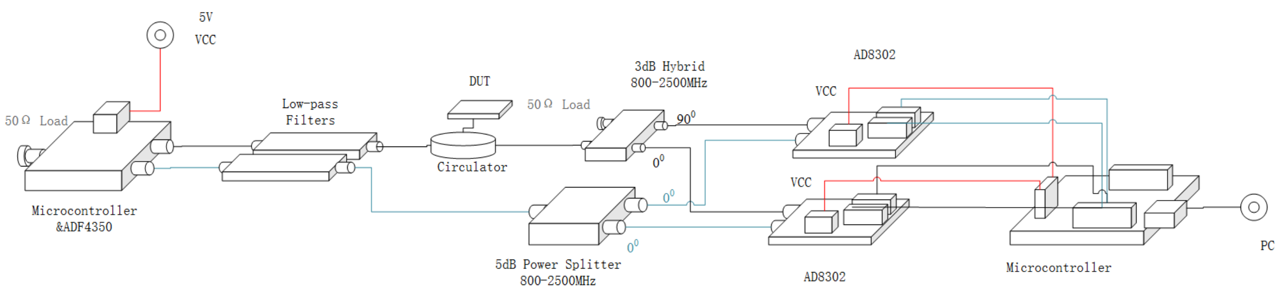

Figure 1 Model of improved EM signal measurement system

1.3 Challenges

1. Programme the STC8051 and ADF4350 development suit to generate frequency up to 2.7GHz.

2. Read real-time data and from computer port and separate it.

3. Make a 90^0's phase shift by using 3dB hybrid.

4. Test and calibrate the system.

Chapter 2: Background

In chapter 2, I mainly discuss the improved measurement method in detail which is critical to implement this project. The relevant principles of key devices including ADF4350, STC8051 microcontroller, AD8302 detector, 3dB power splitter and 3dB hybrid are also introduced in this part.

2.1 Improvement for EM measurement System

The existing EM measurement system is based on the block diagram shown as follows. For frequency generation, the microwave source is designed as a PLL synthesizer ADF4350 and a microcontroller, which can generate frequency up to 2.7GHz. A LCD1602 screen is used to display the frequency. A switch is used to change the value of frequency. For the phase and amplitude detection, in order to detect the differences between the signals, a microwave receiver is proposed, based on a gain and phase detector AD8302.

The detection of the system is inaccurate because the existing given system can only approximately detect amplitude and phase difference across two ports by providing a certain voltage output, without determining the direction of the phase difference. [2][3] Moreover, the frequency generation process, which is based on the microcontroller, cannot cooperate with a software on computer. Thus, an improved detection method and a new frequency generation method need to be used.

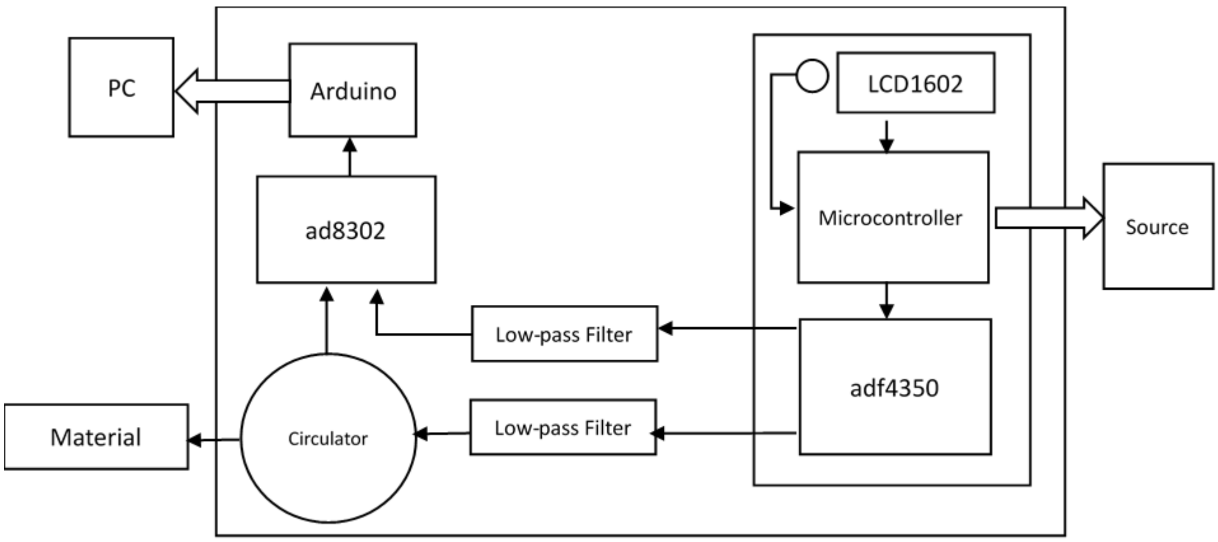

Figure 2 Block diagram of existing EM measurement system

To improve the accuracy of the system, a new method should be used. In this method, there is a 90^0 3dB hybrid and two AD8302 chip identifying whether the phase difference is a negative number or a positive one, so that the ambiguity in phase of the AD8302 circuit can be solved. [4]

The measurement system is connected to a PC through two USB ports. One port is used to set the value of desired frequency and the other is to receive the outputs into the desired display. The detailed functions will be achieved within a software.

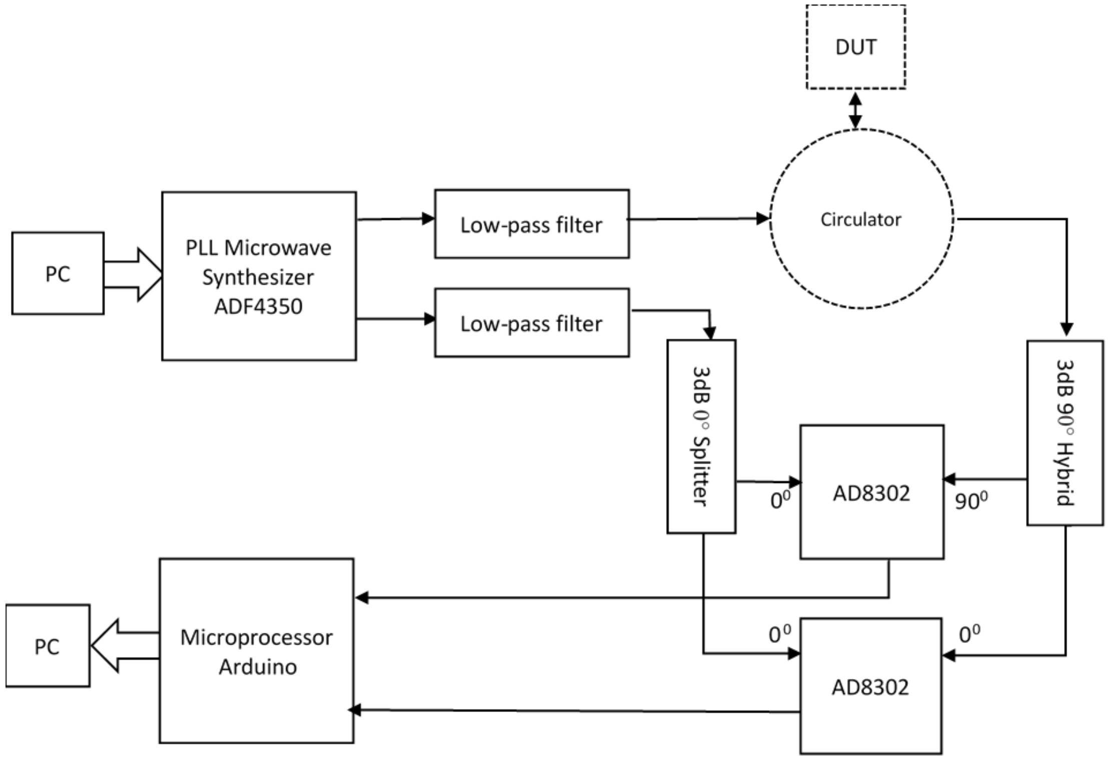

Figure 3 Block diagram of improved EM signal measurement system

2.2 PLL synthesizer-ADF4350

The ADF4350 allows implementation of fractional-N or integer-N phase-locked loop (PLL) frequency synthesizers if used with an external loop filter and external reference frequency. [6]

There is an integrated voltage-controlled oscillator (VCO) in the ADF4350 chip, which outputs a fundamental frequency ranging from 2200 MHz to 4400 MHz. However, because of the inside divide-by-1/2/4/8/16 circuits, the RF output frequencies can be as low as 137.5 MHz. If isolation is required in applications, the RF output stage can be muted. The mute function is both pin- and software-controllable. Besides, an auxiliary RF output is also available, which can be powered down if not in use. A simple 3-wire interface is used to control all the on-chip registers. The device operates with a power supply ranging from 3.0 V to 3.6 V. [6] The functional block of ADF4350 chip is shown as follows.

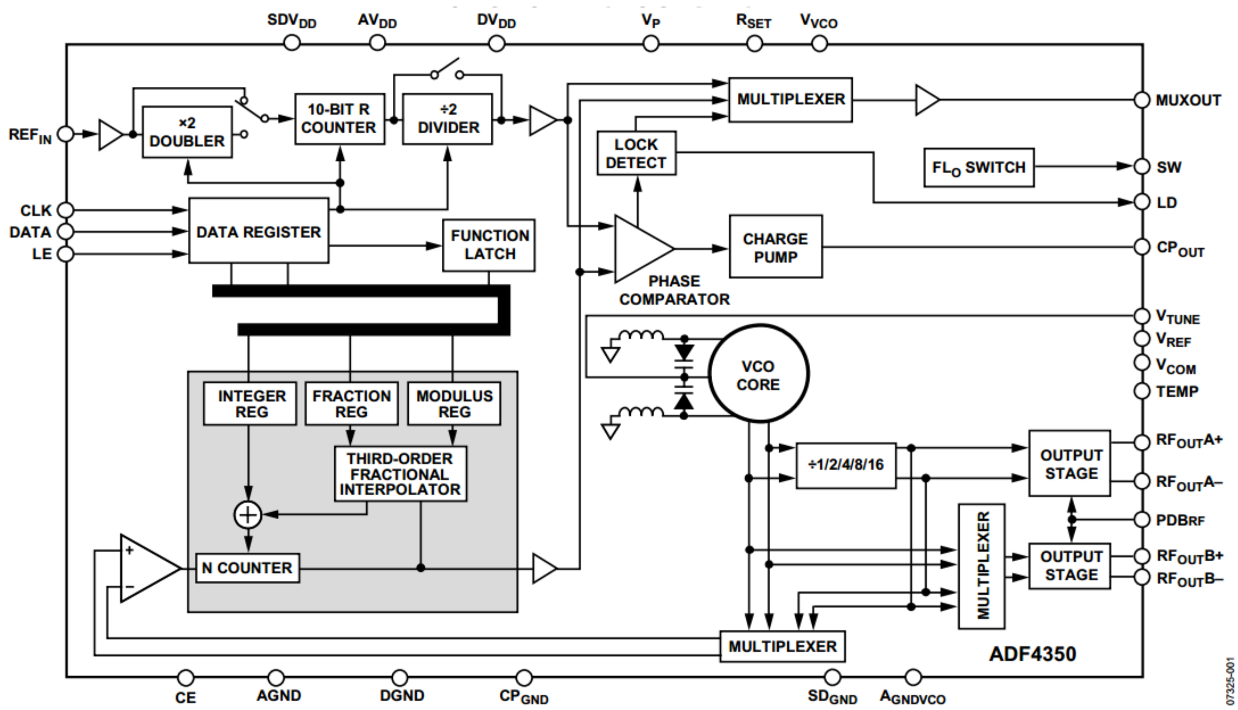

Figure 4 Functional block of ADF4350 [11]

Figure 5 shows the reference input stage. SW1 and SW2 are normally closed switches. SW3 is normally open. SW3 is closed, and SW1 and SW2 are opened when power-down is initiated, which ensures that the loading of the REF_{IN} pin is shut off during power-down.

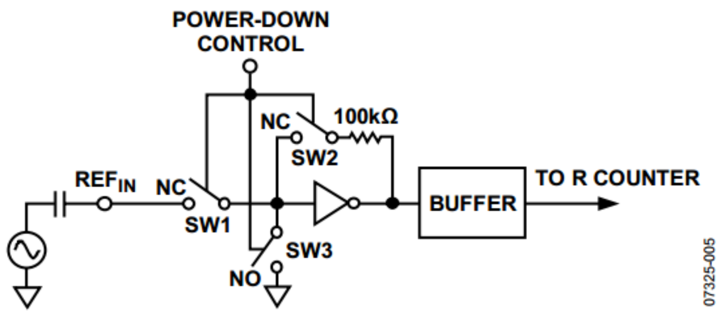

Figure 5 Reference input stage [11]

There is a RF N divider allowing a division ratio in the PLL feedback path. The division ratio can be determined by the values of INT, FRAC and MOD, which build up this divider.

The following equations are used to programme an ADF4350 synthesizer: [7]

$$RF_{out} = \left[INT + \left(FRAC/MOD \right) \right] \times \left[f_{PFD} \right] / RF\ divider \tag{1}$$

where:

RF_{out} is the RF frequency output.

INT is the integer division factor.

FRAC is the fractionality.

MOD is the modulus.

RF divider is the output divider that divides down the VCO frequency. [11]

$$f_{PFD} = REF_{IN} \times \left[\left(1 + D\right) / \left(R \times \left(1 + T\right) \right) \right] \tag{2}$$

where:

REF_{IN} is the reference frequency input.

D is the RF REF_{IN} doubler bit.

T is the reference divide-by-2 bit (0 or 1).

R is the RF reference division factor.

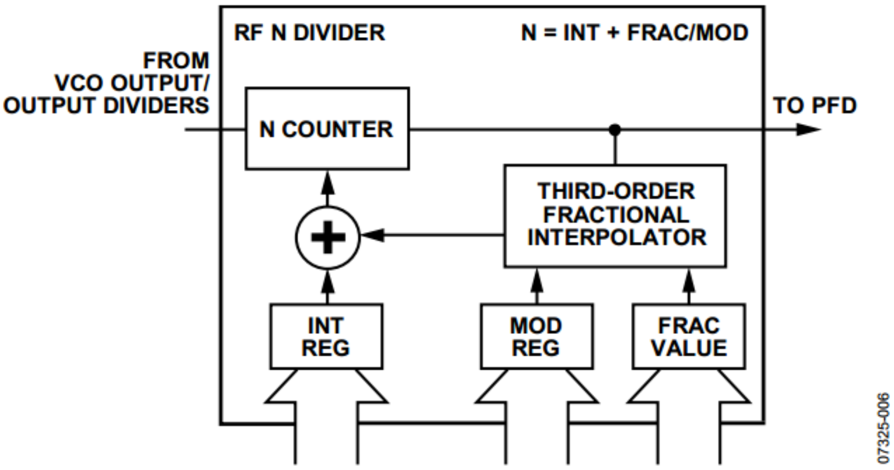

Figure 6 RF INT divider [11]

When the FRAC = 0 and DB8 in Register 2 (LDF) is set to 1, the integer-N mode is triggered in the synthesizer. The DB8 in Register 2 (LDF) must be set to 1 to get integer-N digital lock detect.

The 10–bit R counter is used to divide down the input reference frequency (REF_{IN}) to produce the reference clock to the PFD. The ratios of division is allowed to set from 1 to 1023. [11]

The RFOUTA+ and RFOUTA− pins of the ADF4350 are connected to the collectors of an NPN differential pair, which is driven by buffered outputs of the VCO. In order to allow users to optimize the power dissipation vs. the output power requirements, the tail current of the differential pair is programmable by Bits [D2:D1] in Register 4 (R4). Four current levels can be set. [11]

These levels give output power levels of −4 dBm, −1 dBm, +2 dBm, and +5 dBm, respectively, where a 50 Ω resistor to AVDD and ac coupling into a 50 Ω load are used. Alternatively, both outputs can be combined in a 1 + 1:1 transformer or a 180° microstrip coupler. If the outputs are individually used, the optimum output stage consists of a shunt inductor to VVCO. The unused complementary output must be terminated with a similar circuit to the used output. [11]

An auxiliary output stage exists on Pins RFOUTB+ and RFOUTB− providing a second set of differential outputs which can drive another circuit, or which can be powered down if unused. The auxiliary output can be used only in conjunction with the main RF output, which cannot be used when the main output is powered down. Another feature of the ADF4350 is that the supply current to the RF output stage can be shut down until the device achieves lock as measured by the digital lock detect circuitry. This is enabled by the mute till lock detect (MTLD) bit in Register 4 (R4).

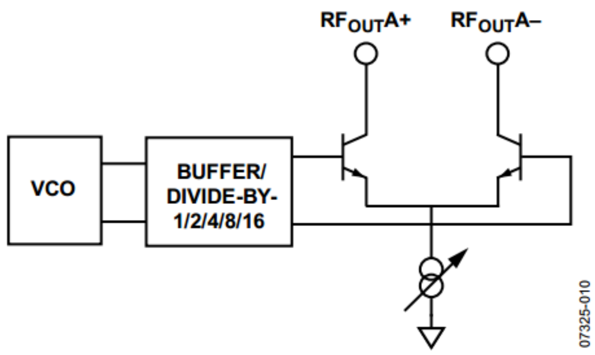

Figure 7 Output stage [11]

Register maps: [7] There are six registers in an ADF4350 chip. By referring to its register map, parameters can be set and written into relevant registers through programming.

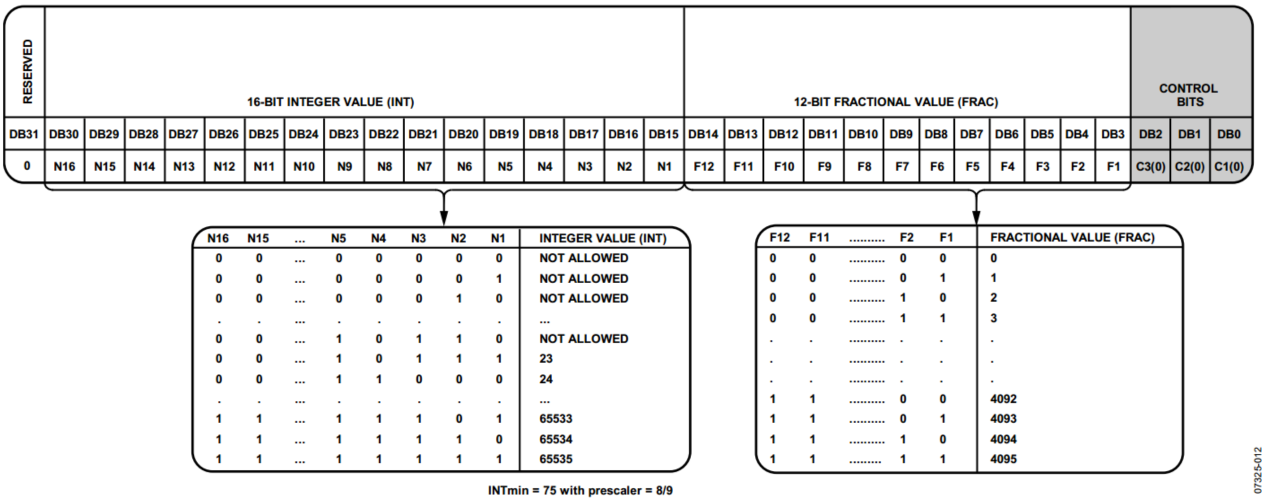

Figure 8 Register 0 (R0) of ADF4350 [11]

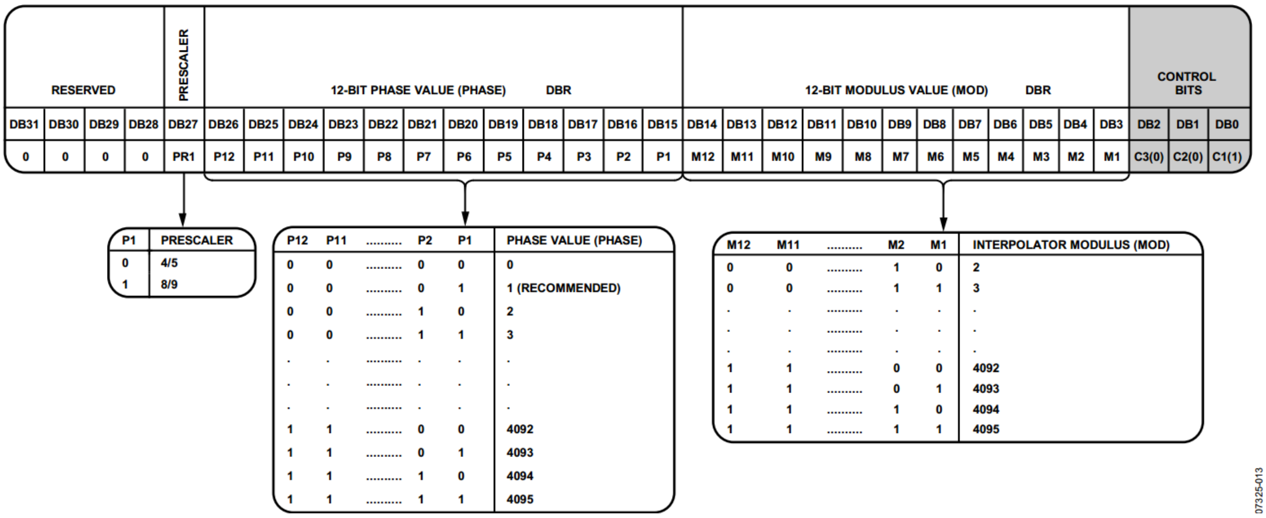

Figure 9 Register 1 (R1) of ADF4350 [11]

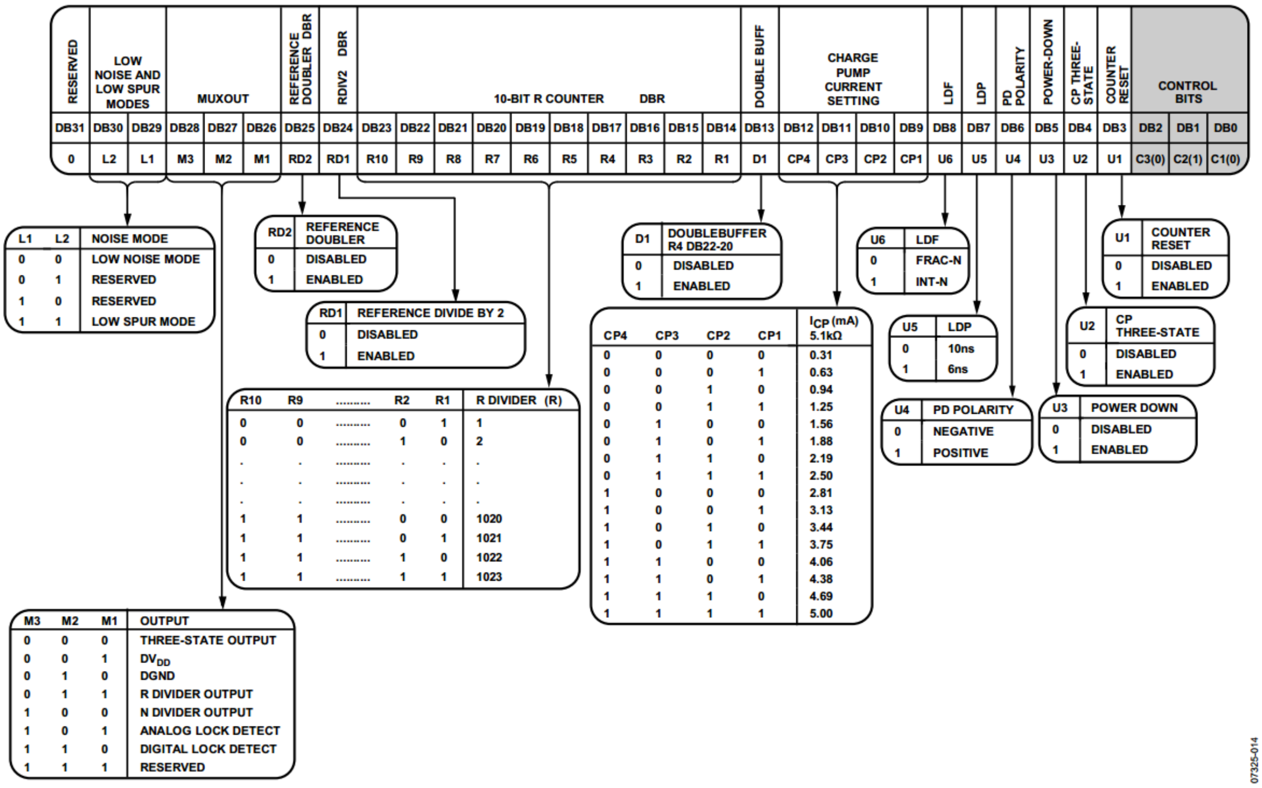

Figure 10 Register 2 (R2) of ADF4350 [11]

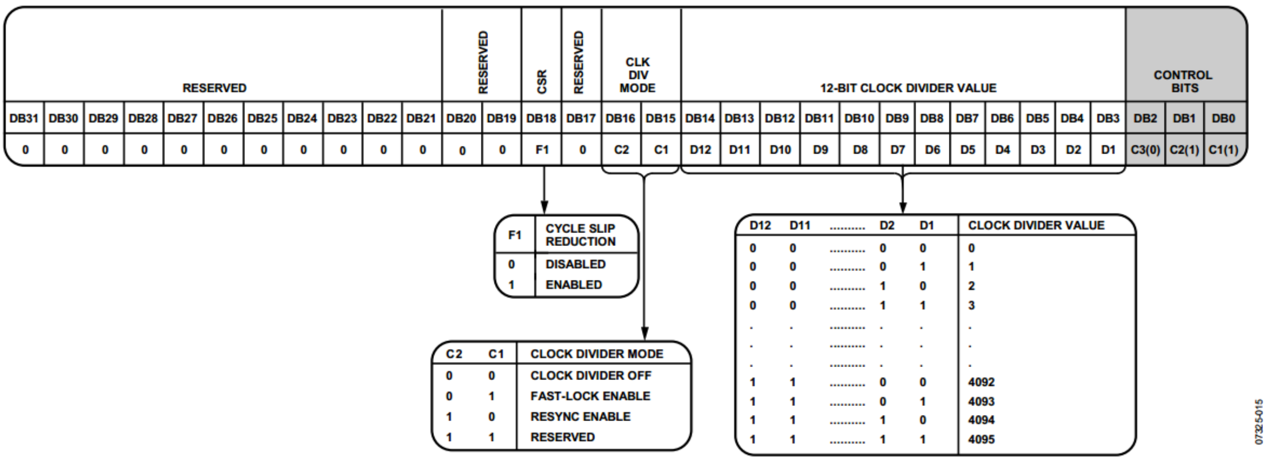

Figure 11 Register 3 (R3) of ADF4350 [11]

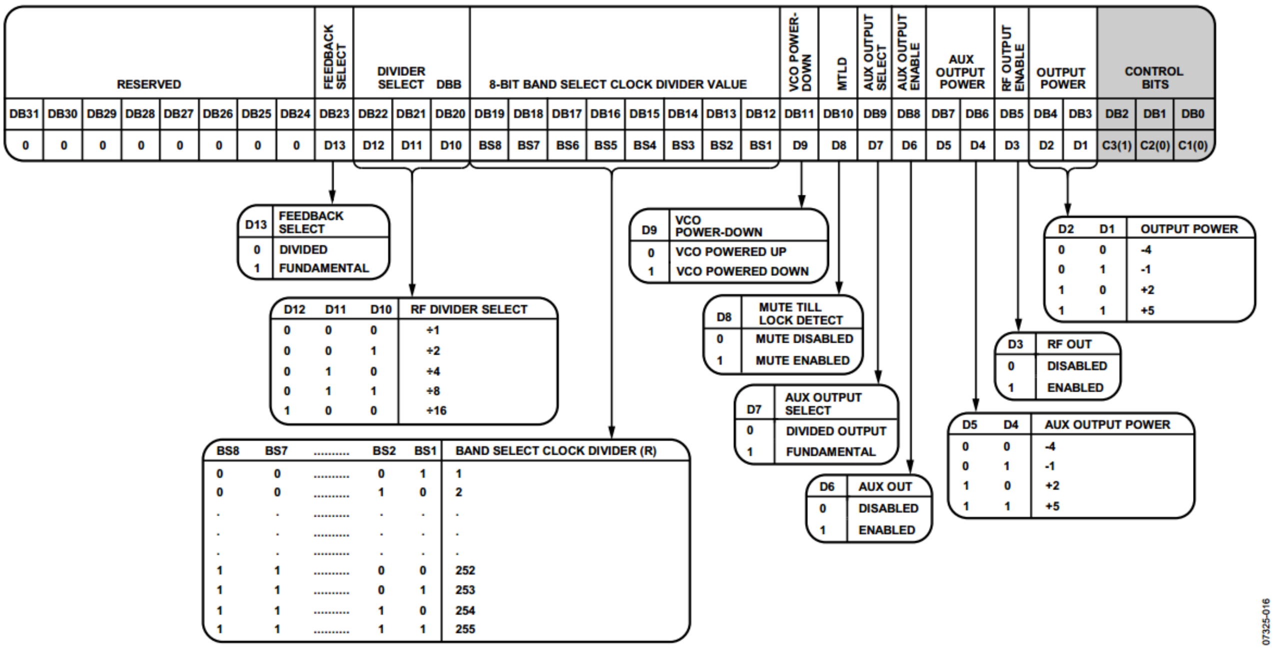

Figure 12 Register 4 (R4) of ADF4350 [11]

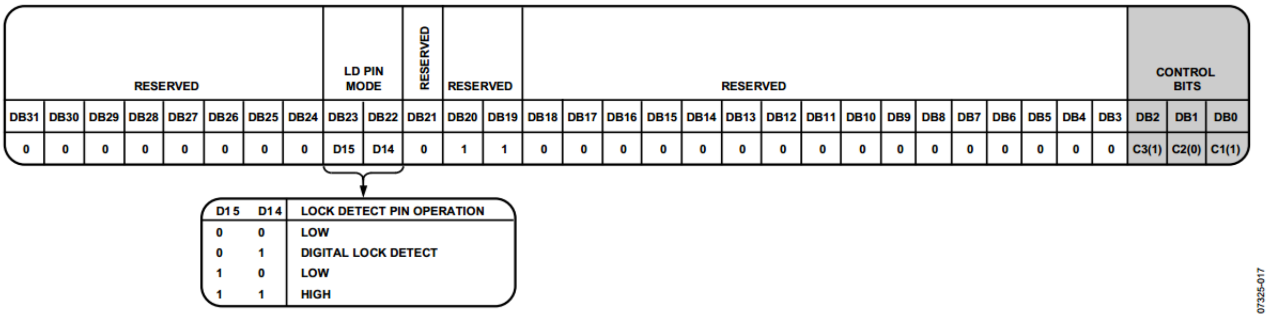

Figure 13 Register 5 (R5) of ADF4350 [11]

2.3 STC8051 microcontroller-STC15F2K60S2

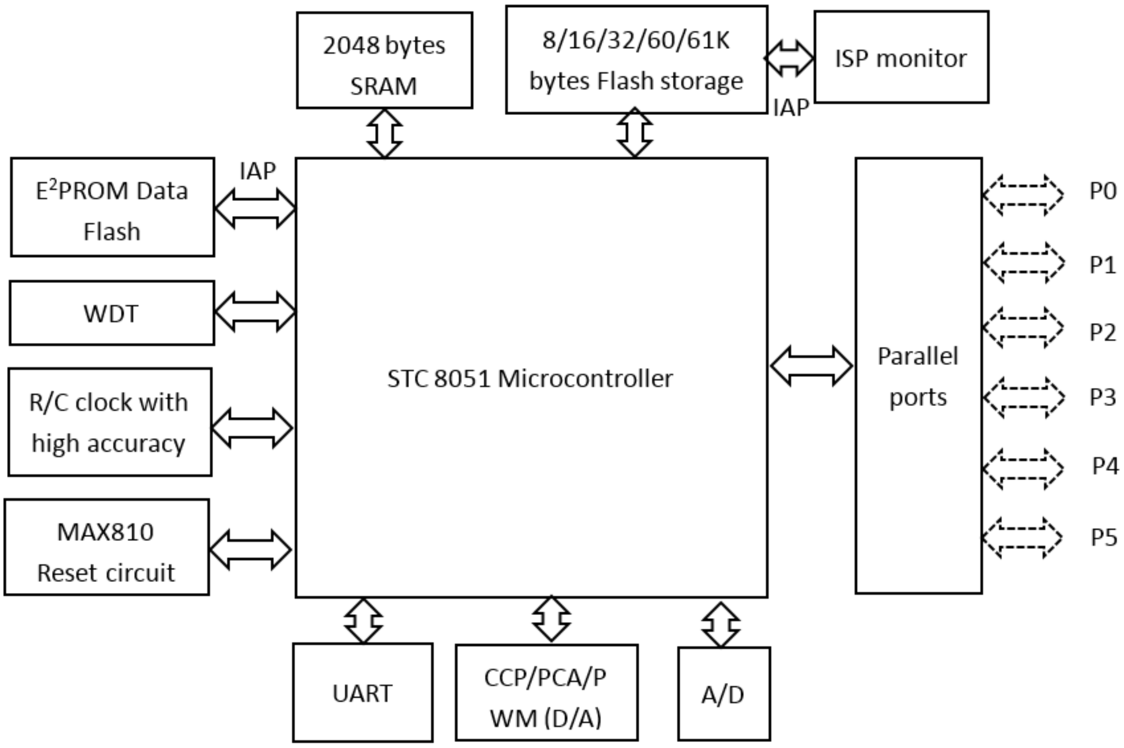

Figure 14 Features of STC8051 microcontroller [13]

To make the frequency generation process more efficient, a STC8051 microcontroller is used which is STC15F2K60S2. The speed of this STC8051 microcontroller is five or eight times higher than traditional STC8051 chips. Besides, it contains all the circuit units that are required in data collection and data processing. [13]

2.4 Phase and amplitude detector-AD8302

2.4.1 Introduction of AD8302 chips

The AD8302 can be used as a phase and amplitude detector by sending it a reference signal and a phase-shifted signal. The phase shift is caused by a 3dB 90^0 hybrid and a Device Under Test (DUT) by reflection or transmission, and the goal is to determine the exact effect of the DUT. [5]

The inputs of the two signals are single-ended, meaning that they can be matched and connected by the directional coupler. In low frequency range, the input impedance is normally 3000Ω. Moreover, the chip includes a multiplier phase detector. However, it is the outputs of the two logarithmic amplifiers that show the precise phase balance which is driven by the fully limited signals. Thus, the phase accuracy measurement depends on signal level over a wide frequency range.

TheAD8302 features the following main points:

1. In low frequency to 2.7GHz frequency range measured within two input signal amplitude ratio and phase difference.
2. For 50-ohm measurement system, the input range of -62dBm ~-2dBm.
3. Accurate amplitude measurement scale factor of 30mV/dB
4. Exact typically less than 0.5dB
5. Accurate phase measurement scale factor of 10mV/degree
6. Precision typically less than1degree
7. The device is in operation, with a comparison of three work measurement, control and level mode
8. Withastable1.8Vreferencevoltagebiasoutput
9. Responseto30MHzvideobandwidth
10. A small pin TSSOP package
11. Operating temperature -40degree~85degree

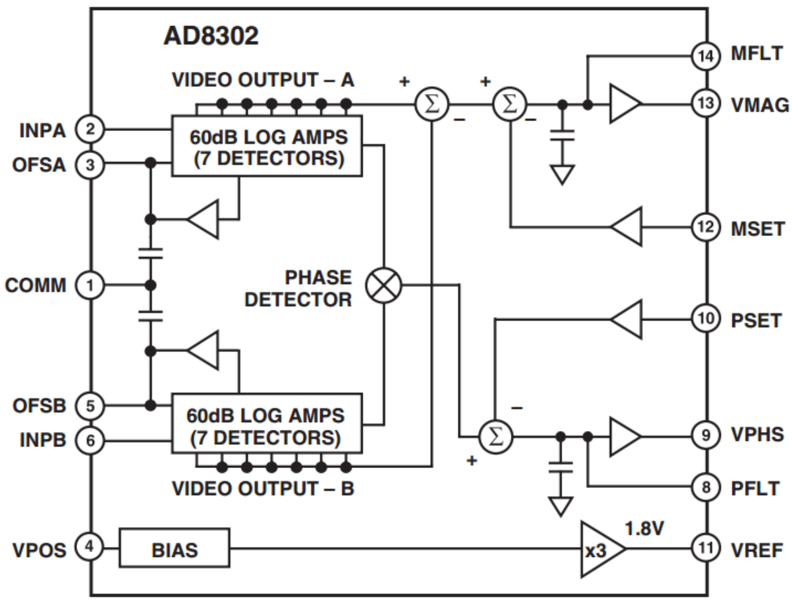

Figure 15 Inside structure of AD8302 [1]

The phase difference can be in the range from –180° to +180°. In the measurement mode, the output functions are as follows. [1]

$$V_{OUT} = \left(I_{IN} R_F + V_{CP}\right) / \left(1 + sT\right) \tag{3}$$

$$V_{MAG} = R_F I_{SLP} \log\left(V_{INA} / V_{INB}\right) + V_{CP} \quad \text{or} \tag{4}$$

$$V_{MAG} = \left(R_F L_{SLP} / 20\right)\left(P_{INA} - P_{INB}\right) + V_{CP} \quad \text{or} \tag{5}$$

$$V_{MAG} = -R_F I_{\Phi} \left(\left|\Phi\left(V_{INA}\right) - \Phi\left(V_{INB}\right)\right| - 90^{\circ}\right) + V_{CP} \tag{6}$$

For the gain function, the slope represented by RF ISLP is 600 mV/decade or, dividing by 20 dB/decade, 30 mV/dB. With a centre point of 900 mV for 0 dB gain, a range of –30 dB to +30 dB covers the full-scale swing from 0 V to 1.8 V. For the phase function, the slope represented by RFIΦ is 10 mV/degree. With a centre point of 900 mV for 90°, a range of 0° to 180° covers the full-scale swing from 1.8 V to 0 V. The range of 0° to –180° covers the same full-scale swing but with the opposite slope. [1] The following figures show the relationship between voltage output and the real value.

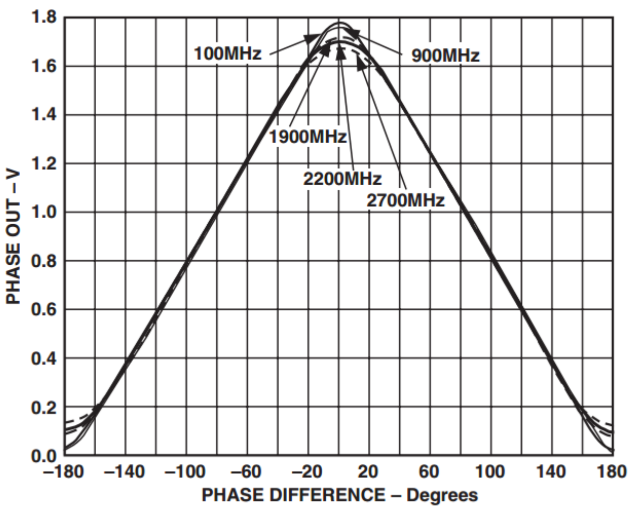

Figure 16 Phase output of AD8302 [1]

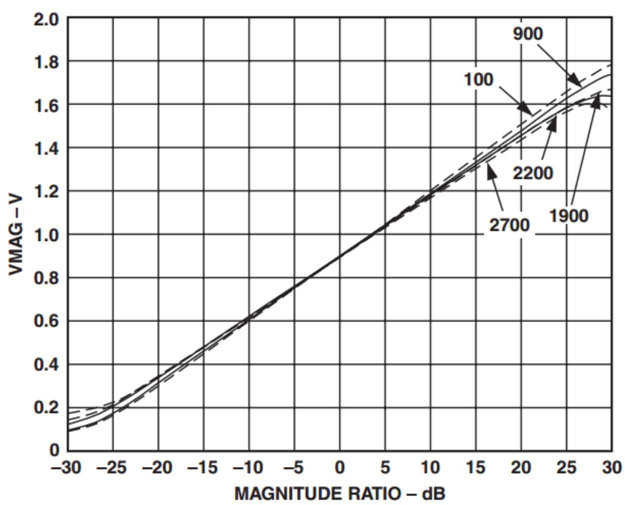

Figure 17 Magnitude output of AD8302 [1]

Ideally

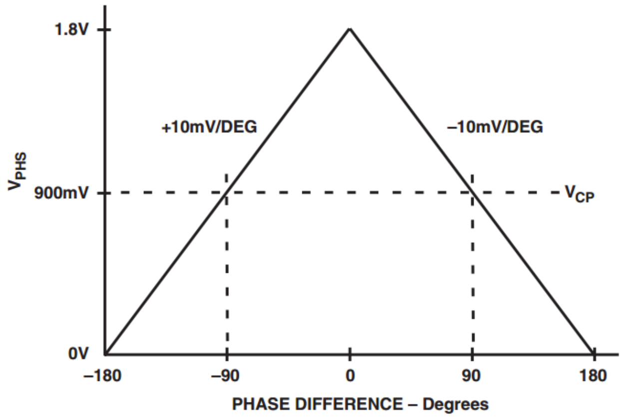

Figure 18 Phase difference of AD8302 [1]

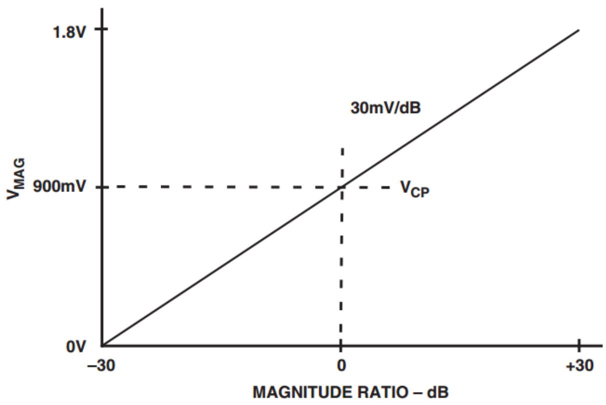

Figure 19 Magnitude ratio of AD8302 [1]

2.4.2 Principle of improved method using two AD8302 chips

AD8302 can provide an output corresponding to 0^0-180^0, but it cannot distinguish the sign of the phase, which means that if the output voltage is 1.2V, then the phase can be both -60^0 and 60^0. As a result, it is necessary to use an additional phase shifting of the reference signal to identify the sign of the phase. [4]

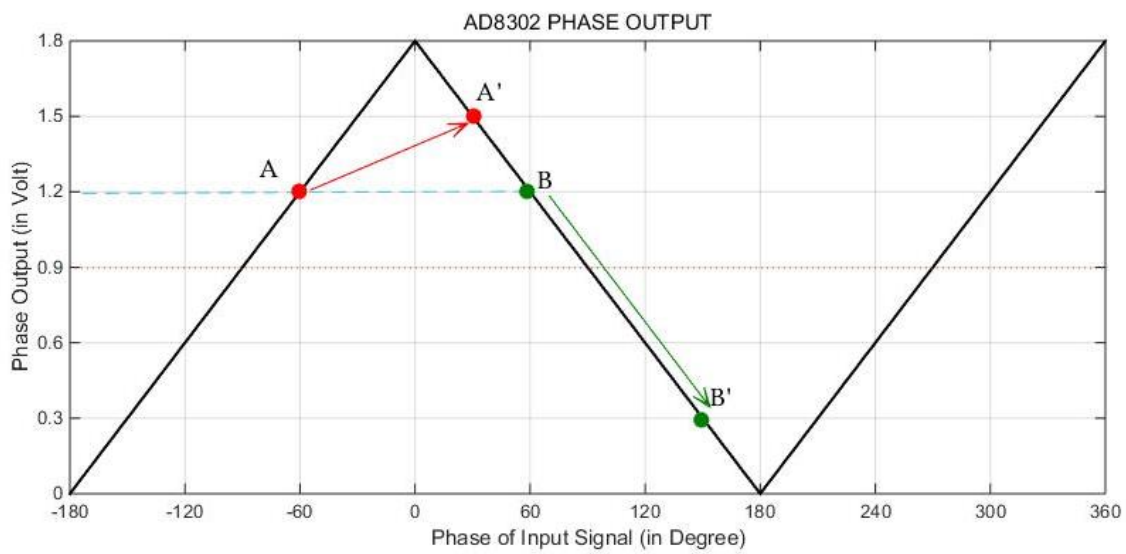

Figure 20 Example of a 90^0's phase shift [4]

In the figure above, every possible output value can be produced by two different phases, which are negatives of each other. For example, points A and B, representing ±60^0, both produce an output of 1.2V. So, by one AD8302, the result can be at A or B (±60^0), but the two points cannot be distinguished. To solve the problem, a 90^0 shift needs to be achieved. With the phase shift, point A will move as shown by the arrow to 30^0, and point B will move as shown by the arrow to 150^0. By analyse the results with and without the phase shift, the A and B can be distinguished and I can obtain the accurate result. [5]

If the original measured phase is PM (a positive number) then the actual phase PA is either – PM or + PM. The measured shifted phase will be PS, whose sign is readily determined as describe above, so it will be the actual signed phase. Ideally, PS will equal either S-PM or S+PM, where S is the amount of the shift (S=90^0 in Figure 20). [5]

To allow for measurement and shift errors, we make the decision as follows: [5]

1. S is the midpoint between the two possible shifted values.

2. If PS >S then PA = PM

3. Otherwise PA = -PM

2.5 3dB splitter and 3dB hybrid

2.5.1 3dB splitter

Power divider (also power splitter) is a passive device used mostly in the field of radio technology. It couples a defined amount of the electromagnetic power in a transmission line to a port enabling the signal to be used in another circuit. An essential feature of power splitter is that it only couples power flowing in one direction. Power entering the output port is coupled to the isolated port but not to the coupled port. It is designed to split power equally between two ports which are P2 and P3 in the following figure. [10]

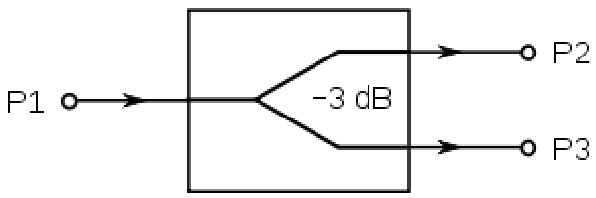

Figure 21 Structure of 3dB power splitter [10]

2.5.2 3dB hybrid

Hybrid couplers are the special case of a four-port directional coupler that is designed for a 3-dB (equal) power split. When the coupling is designed to be 3 dB it is called a hybrid coupler. The S-matrix for an ideal, symmetric hybrid coupler reduces to

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -i & -1 & 0 \\ -i & 0 & 0 & -1 \\ -1 & 0 & 0 & -i \\ 0 & -1 & -i & 0 \end{bmatrix} \tag{7}$$

The two output ports have a 90^0's phase difference (-i to −1) and so this is a 90^0 hybrid. [8]

Features of 90-degree Hybrid:

1. 90-degree Hybrid Directional Coupler splits power equally between its two output ports. [9]
2. Two output ports are 90 degrees out of phase with each other.
3. Construction is stripline in an aluminum housing.

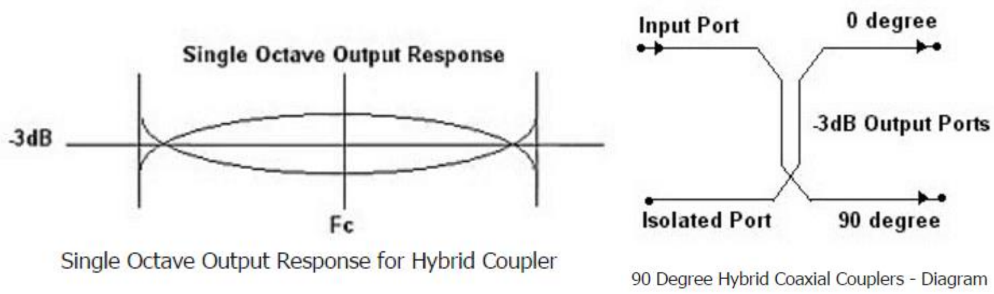

Figure 22 Structure of 3dB hybrid [9]

Chapter 3: Design and Implementation

In this part, the connection of devices is introduced. Each key step is discussed in detail which includes hardware implementation and GUI design. Besides, the simple simulation of low-pass filters is shown in this part.

The EM signal measurement system can work in the following process.

User set desired frequency throughout a GUI. Then, signals are generated by the ADF4351 chip which will pass the low-pass filters. One of the signals goes through the power splitter which will perform as reference signals, while the other signal is sent to 3dB hybrid to add an extra 900's phase shift. Then, one shifted signal passes DUTs. The feedback signal and reference signal will be transmitted to the AD8302 to detected amplitude and phase. After that, the voltage output will be collected, processed, and sent to GUI to display.

3.1 Hardware implementation

3.1.1 Frequency generation

The ADF4350 PLL synthesizer can generate frequency between 137.5MHz and 4.4GHz. In this system, the frequency needs to be from 800MHz to 2.5GHz which is well-covered by the range of frequency by ADF4350.

To achieve the frequency generation, I used a development suit which includes an ADF4350 and a STC8051 microcontroller. The structure of this suit is as follows.

Figure 23 ADF4350 development suit

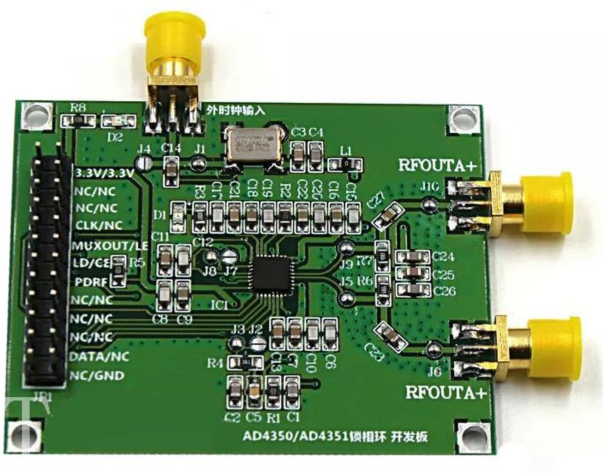

Figure 14 ADF4350 evaluation board

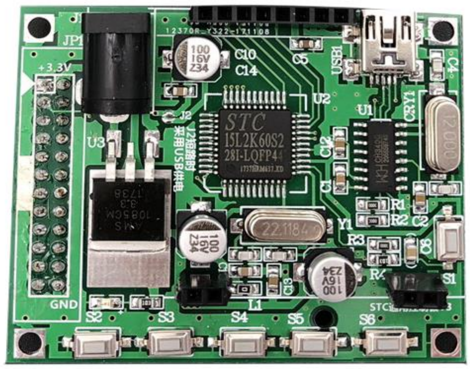

Figure 25 STC15F2K60S2

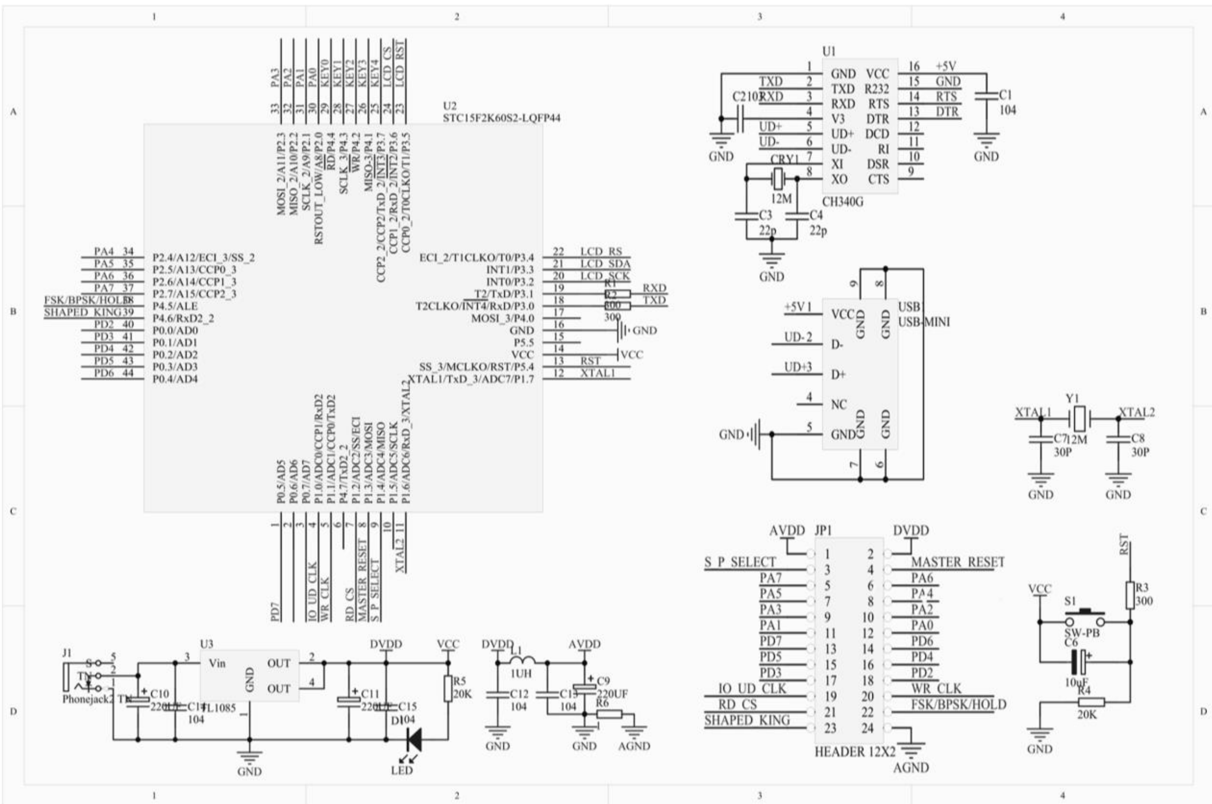

Figure 26 Common DDS control board [12]

To generate the required frequency, a method is used in the STC8051 chip programme. There are five range of frequency of AD4350, which are 137.5MHz to 275MHz, 275MHz to 550MHz, 550MHz to 1.1GHz, 1.1GHz to 2.2GHz, and 2.2GHz to 4.4GHz. By setting the value of MODE and dividers, I can calculate the value of INI and FRAC according to the following equations.

$$RF_{out} = \left[INT + \left(FRAC/MOD\right)\right] \times \left[f_{PFD}\right] / RF\ divider \tag{8}$$

$$f_{PFD} = REF_{IN} \times \left[\left(1+D\right) / \left(R \times \left(1+T\right)\right)\right] \tag{9}$$

Then, the values can be written into ADF4350 registers.

```
void  WriteFreq(unsigned long Freq)  //137500KHZ
{float Freq_temp;
 unsigned long  INT,FRAC,MODE=3125;
 unsigned long R0_temp=0;  //Control INT  FRAC
 unsigned long R4_temp=0; // 2.2GHZ--4.4GHZ/1.1GHZ--2.2GHZ/550MHZ--1.1GHZ/275MHZ--550MHZ/137.5MHZ--275MHZ
 unsigned long  Out_Divider,RF_Divider;
if((34375<=Freq)&&(Freq<68750))  {Out_Divider=6;RF_Divider=64;}   //ADF4351
if((68750<=Freq)&&(Freq<137500)) {Out_Divider=5;RF_Divider=32;}   //ADF4351
if((137500<=Freq)&&(Freq<275000)){Out_Divider=4;RF_Divider=16;}
if((275000<=Freq)&&(Freq<550000)){Out_Divider=3;RF_Divider=8;}
if((550000<=Freq)&&(Freq<1100000)){Out_Divider=2;RF_Divider=4;}
if((1100000<=Freq)&&(Freq<2200000)){Out_Divider=1;RF_Divider=2;}
if((2200000<=Freq)&&(Freq<4400000)){Out_Divider=0;RF_Divider=1;}

  R4_temp=((R4&0XFF8FFFFF)|(Out_Divider<<20));
  WriteToADF435X(R4_temp);

 Freq_temp=Freq;
 Freq_temp=(Freq_temp*RF_Divider)/25000;
     INT  =(int)Freq_temp;
     FRAC=  (Freq_temp-INT)*MODE;
R0_temp=((INT<<15)|(FRAC<<3))&0x7ffff8;
 WriteToADF435X(R0_temp);

}
```

Figure 27 Codes to generate frequency

For the microcontroller needs to receive data from computer port, the program also use codes to communicate with a specific port.

```
serial_one_init();//initiate port

WriteFreq(Freq_init);
while(1)
{
    if(serial_one_read_count > 0)//if data received
    {
      Delay10ms();//delay
      //serial_one_send_length_string(serial_one_read_data,serial_one_read_count);
      serial_one_read_count = 0;
      WriteFreq(serial_one_read_data);
    }
}
```

Figure 28 Codes related to port communication

The following figures show the performance of ADF4350 and STC8051 development suit when producing different frequencies.

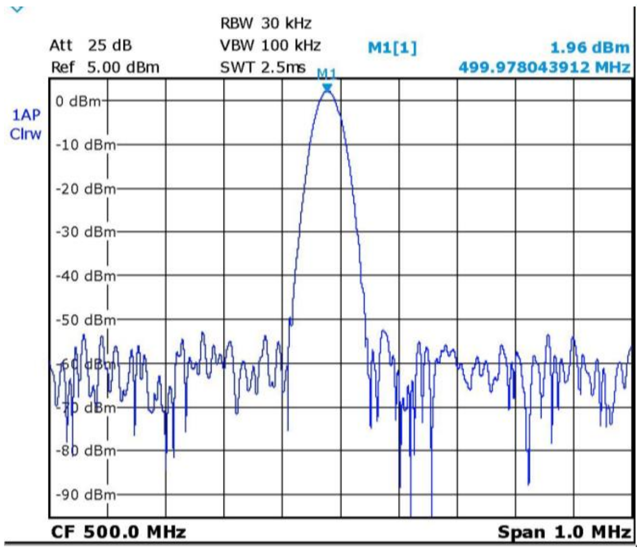

Figure 29 500MHz frequency

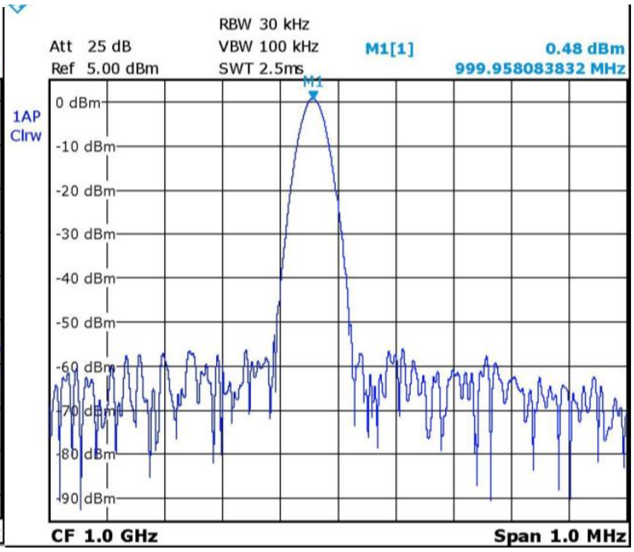

Figure 30 1GHz frequency

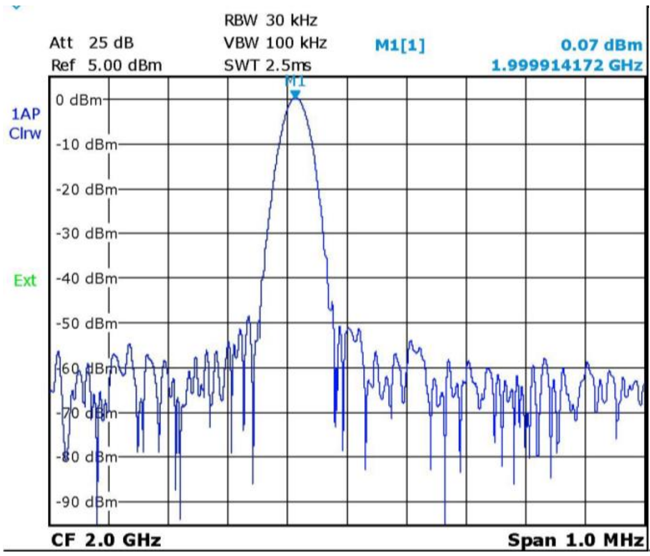

Figure 31 2GHz frequency

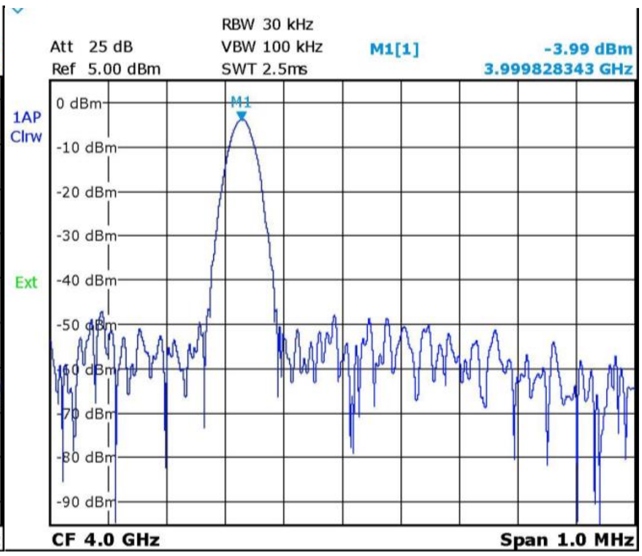

Figure 32 4GHz frequency

3.1.2 90^0's phase shift

A 90^0's phase shift can be achieved by using a 3dB 90^0 hybrid.

Figure 33 3dB 90^0 hybrid

The input port is connected to a signal generated by ADF4350. The phase difference between the two output ports is 90^0. However, the phase difference between the input port and output ports is unknown. Thus, calibrations need to be made to ensure the accuracy of the system.

3.1.3 Phase and amplitude detection

As I discussed before, there are two AD8302 chips are used. One is used to detect the reference signal, and the other is to detect the signal with 90^0's phase shift. The two AD8302 chips are connected to a microcontroller which is an Arduino board in this case.

The following figure shows the structure of AD8302 integrated board used in this project. The SMA ports are connected to the input signals. When the signals pass through the circuit, the phase difference will be output by P1 and the amplitude ratio will be output by P2, both of which will be represented as voltage value.

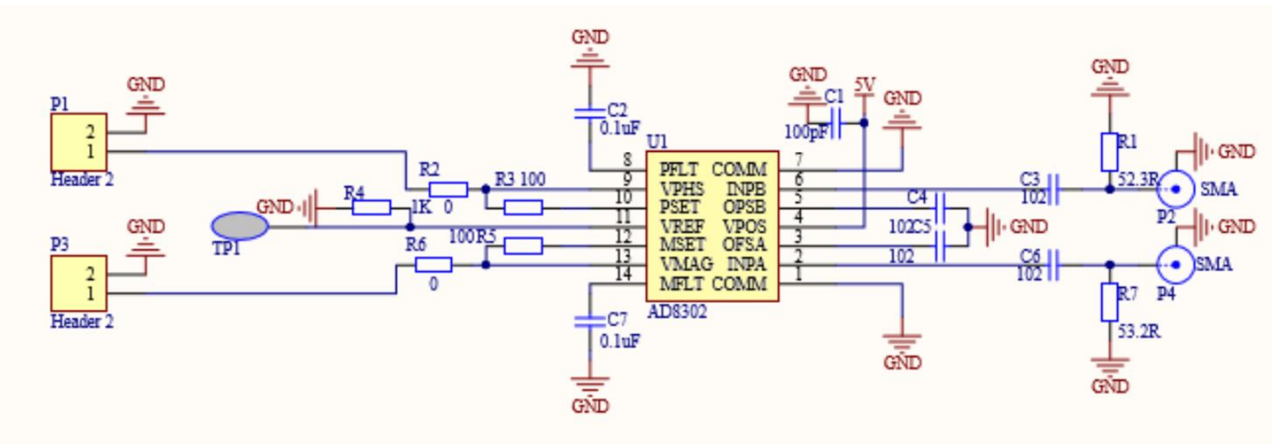

Figure 34 Inside circuit of AD8302 board [2]

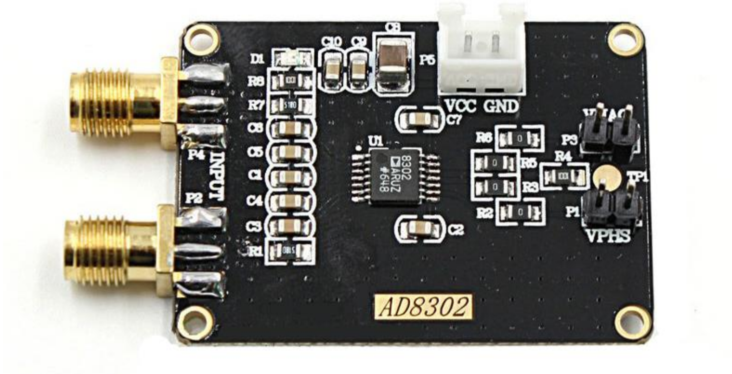

Figure 35 AD8302 board

The following shows the connections of the hardware and the way of processing data collected by Arduino board. The “phs1” and “mag1” are the values of two reference signals. The “phs1” is connected to pin A1 of Arduino, and “mag1” is connected to pin A2. In the figure X, the AD8302 at right is used to detect the reference signals. The reflected signal will be measured by the left AD8302.

```
// ref=analogRead(A0);
 phs1=analogRead(A1);
 mag1=analogRead(A2);
 phs2=analogRead(A3);
 mag2=analogRead(A4);
```

Figure 36 Port connection with AD8302 and Arduino

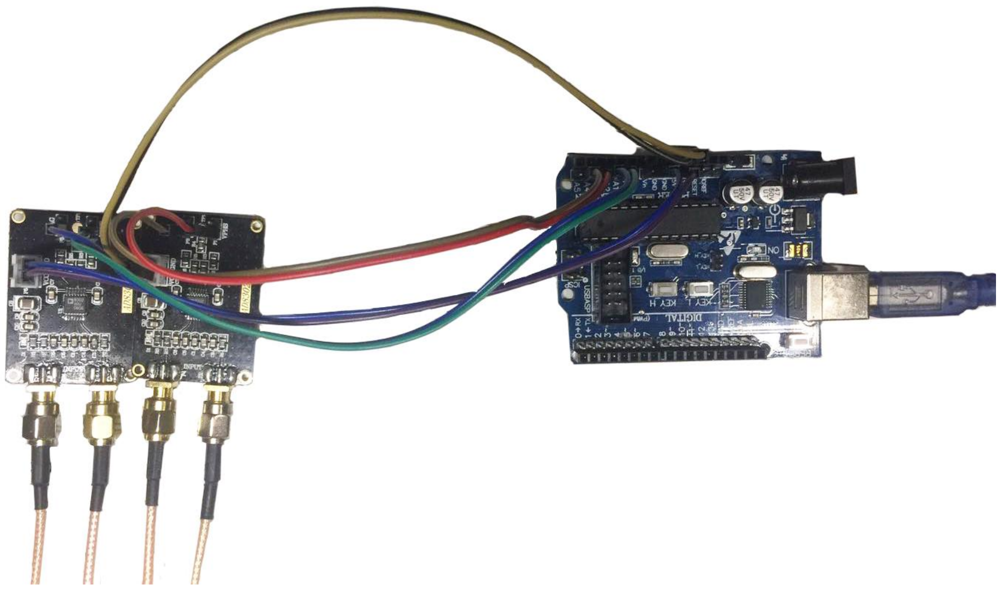

Figure 37 Physical connection of AD8302 and Arduino

The initial output of AD8302 is voltage, which is received by Arduio. Because Arduino has a 10-bit analogy to digital converter and the voltage between 0 and 5V is devided into integer values between 0 and 1023. To conver the value, a function is used as follows.

```
//float vref = ref*(5.0 / 1023.0);
float vphs1 = phs1*(5.0 / 1023.0);
float vmag1 = mag1*(5.0 / 1023.0);
float vphs2 = phs2*(5.0 / 1023.0);
float vmag2 = mag2*(5.0 / 1023.0);
```

Figure 38 Conversion of voltage

According to the characterization of AD8302 chips, which is shown in Figure 8 and Figure 9, the voltages can be transferred into phase difference values and magnitude ratio values.

```
float magniRatio1 = -30.0 + vmag1*(100.0 / 3.0);
float phsDiff1 = 180.0 - 100.0*vphs1;
float magniRatio2 = -30.0 + vmag2*(100.0 / 3.0);
float phsDiff2 = 180.0 - 100.0*vphs2;
```

Figure 39 Calculation of phase difference and amplitude difference

3.1.4 Devices used

Table 1: Hardware devices and their brief description

Name	Details	Parameter	Amount
STC12LE5A60S2	Control the ADF4350 and LCD1602		1
AD8302	Phase measurements units		2
ADF4350 PLL	Generate the frequency		1
Low pass filter	Transfer the sine wave into square wave so that AD8302 can detect it.	500-4400MHz	2
Arduino UNO R3	Read data from AD8302		1
3dB hybrid	Shift the phase by 90 degrees	800-2500MHz	1
5dB power splitter	Splitting one signal to two same signals	800-2500MHz	1
Circulator	Measure the DUTs	800-2700MHz	1
1W SMA Load 50ohm	cables		2

3.2 GUI design

A GUI is required to make users easily to handle the measurement and get the results directly. The GUI will be a C++ program under the environment of VS 2017, which will contain mainly two functions. The first one is to generate the frequency by entering the value to the software. The second one is to receive the data sent from a certain port and display the phase differences and the amplitude differences on the screen. The following figure shows the implemented GUI.

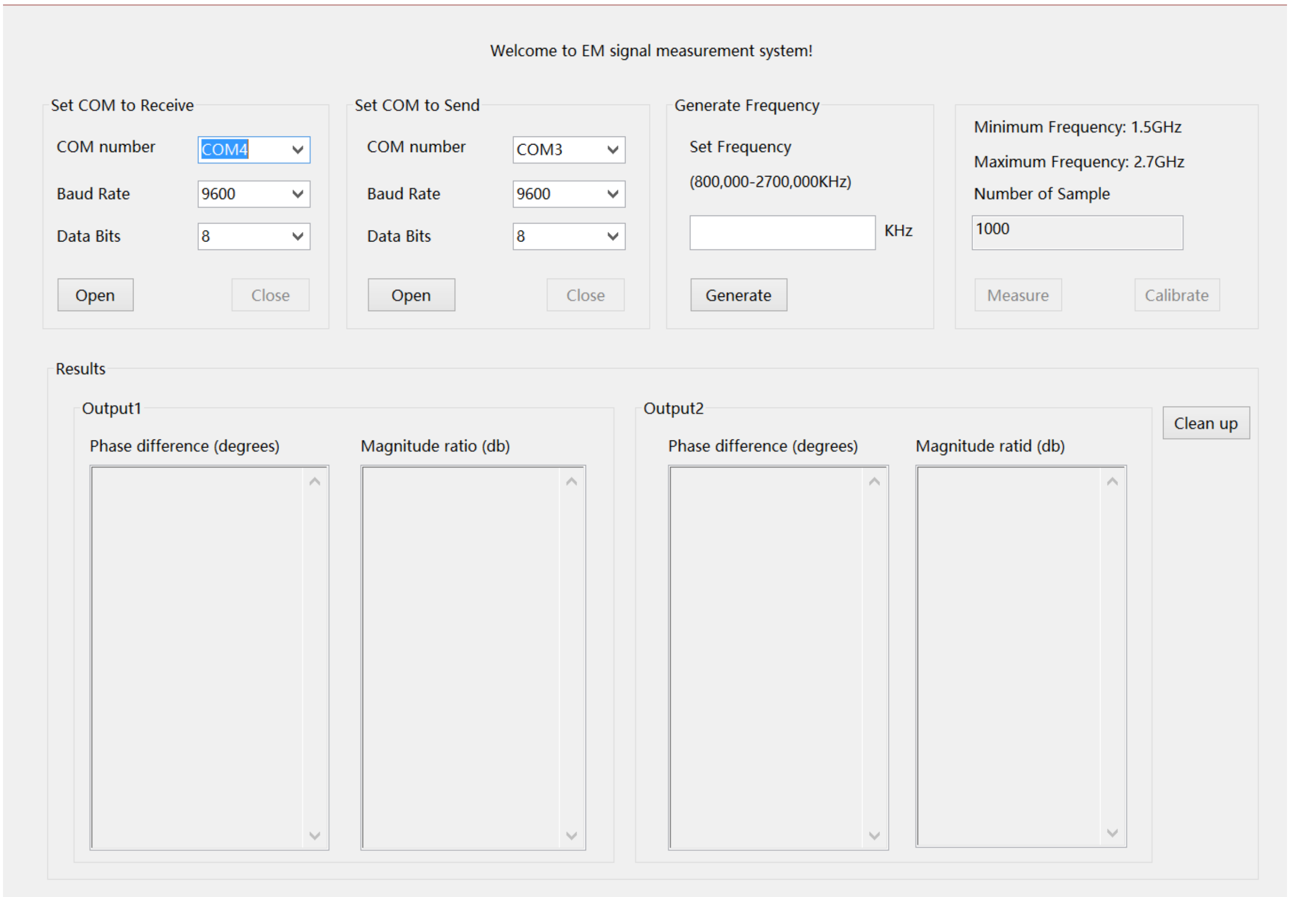

Figure 40 GUI

3.2.1 MFC program design

Microsoft Foundation Classes (MFC) is a class library and a programme framework in VS C++. By using MFC, the GUI program can be easily written.

The function of each element is as follows.

1. Set COM to receive: button “Open”

When the button is entered, the COM will be opened with the parameters set by users, and the system will begin to receive and display real-time data sent to computer COM, which is COM4 by default in this program. Once the COM is opened, the button Close, button Measure will be enabled. Meanwhile, a Boolean flag “m_comOK1” is set as true, which is used to judge if the port is opened or not.

```
void CEMmeasurementsystemV2Dlg::OnBnClickedOpenbutton1()
{
    // TODO: Add your control notification handler code here
    int nSel1;
    CString selComnumber1;

    if (m_ctrlMscomm1.get_PortOpen())
        m_ctrlMscomm1.put_PortOpen(FALSE);

    nSel1 = m_ctrlComNumber1.GetCurSel();
    selComnumber1.Format(_T("%d"), nSel1 + 1);
    m_ctrlMscomm1.put_CommPort(nSel1 + 1);//Select COM
    m_ctrlMscomm1.put_InputMode(1);
    m_ctrlMscomm1.put_InBufferSize(1024);
    m_ctrlMscomm1.put_OutBufferSize(1024);

    CString str;
    int selBaudRate1 = GetDlgItemInt(IDC_BAUDRATE1);
    str.Format(_T("%d"), selBaudRate1);
    CString selDataBits1;
    GetDlgItemText(IDC_DATABITS1, selDataBits1);

    str = str + _T(",n,") + selDataBits1 + _T(",1");
    m_ctrlMscomm1.put_Settings(str);//Set parameters

    if (!m_ctrlMscomm1.get_PortOpen())
    {
        m_ctrlMscomm1.put_PortOpen(TRUE);//Open COM
        m_ctrlMscomm1.put_RThreshold(2);
                                //m_ctrlMscomm1.put_Settings(_T("9600,n,8,1"));
        m_ctrlMscomm1.put_InputMode(1);
        m_ctrlMscomm1.put_RThreshold(1);
        m_ctrlMscomm1.put_InputLen(0);
        m_ctrlMscomm1.get_Input();
        MessageBox(_T("COM")+selComnumber1+_T(" is opened successfully!"));
        GetDlgItem(IDC_OPENBUTTON1)->EnableWindow(FALSE);
        GetDlgItem(IDC_CLOSEBUTTON1)->EnableWindow(TRUE);
        GetDlgItem(IDC_MEASURE)->EnableWindow(TRUE);

        m_comOK1 = true;
    }
    else
    {
        MessageBox(_T("COM") + selComnumber1 + _T(" cannot be opened!"));
        m_comOK1 = false;
    }


}
```

Figure 41 Codes of open button 1

2. Set COM to send: button “Open”

When the button is entered, the COM will be opened with the parameters set by users, and the users can enter the value of frequency to generate frequency between 800MHz and 2700MHz. A flag “m_comOK2” will be set as true which is like “m_comOK1”.

3. Generate Frequency: button “Generate”

When the button is entered, the specific value of frequency will be sent to the computer COM which is COM3 by default in this program.

```
void CEMmeasurementsystemV2Dlg::OnBnClickedGeneratebutton()
{
    // TODO: Add your control notification handler code here
    if (m_comOK2 == true)
    {
        UpdateData(TRUE);
        CString strSend = COleVariant(m_strSendData);
        int sendData = _ttoi(strSend);
        if (sendData >= 800000 && sendData <= 2700000)
        {
            m_ctrlMscomm2.put_Output(COleVariant(m_strSendData)); //Send data
            MessageBox(_T("Frequency is successfully set as ")+strSend+_T(" KHz"));
        }
        else
        {
            MessageBox(_T("Invalid frequency!"));
        }
    }
    else
    {
        MessageBox(_T("Please open a COM!"));
    }
}
```

Figure 42 Codes of generate button

4. Button “Measure” and button “Calibrate”

The function of these two buttons is to display the measurement results and the calibration results. In the programme, a flag “m_ifCli” is set to judge which result should be displayed.

```
void CEMmeasurementsystemV2Dlg::OnBnClickedMeasure()
{
    // TODO: Add your control notification handler code here
    m_ifCli = 2;
    m_ctrlMscomm1.put_PortOpen(FALSE);
    m_ctrlMscomm1.put_Settings(_T("9600,n,8,1"));
    m_ctrlMscomm1.put_PortOpen(TRUE);//Open COM
    m_ctrlMscomm1.put_RThreshold(2);
    //m_ctrlMscomm1.put_Settings(_T("9600,n,8,1"));
    m_ctrlMscomm1.put_InputMode(1);
    m_ctrlMscomm1.put_RThreshold(1);
    m_ctrlMscomm1.put_InputLen(0);
    m_ctrlMscomm1.get_Input();
    GetDlgItem(IDC_MEASURE)->EnableWindow(FALSE);
    GetDlgItem(IDC_CLIBRATE)->EnableWindow(TRUE);
}
```

Figure 43 Codes of measure button

When the button "Measure" is entered, the value will be set as 2, then by a "if…else…" conditional statement, the system will decide the measurement result should be displayed. This is the same with button "Calibration".

```
void CEMmeasurementsystemV2Dlg::OnBnClickedCalibrate()
{
    // TODO: Add your control notification handler code here
    m_ifCli = 3;
    m_ctrlMscomm1.put_PortOpen(FALSE);
    m_ctrlMscomm1.put_Settings(_T("9600,n,8,1"));
    m_ctrlMscomm1.put_PortOpen(TRUE);//Open COM
    m_ctrlMscomm1.put_RThreshold(2);
    //m_ctrlMscomm1.put_Settings(_T("9600,n,8,1"));
    m_ctrlMscomm1.put_InputMode(1);
    m_ctrlMscomm1.put_RThreshold(1);
    m_ctrlMscomm1.put_InputLen(0);
    m_ctrlMscomm1.get_Input();
    GetDlgItem(IDC_MEASURE)->EnableWindow(TRUE);
    GetDlgItem(IDC_CLIBRATE)->EnableWindow(FALSE);


}
```

Figure 44 Codes of calibrate button

3.2.2 Computer port communication design

To make the program communicate with computer ports. Two ActiveX controls are used to deal with data dent and data received. The main function of ActiveX controls in this program is to receive data and display it by a "for…" loop.

```
void CEMmeasurementsystemV2Dlg::OnCommMscomm1()
{
    // TODO: Add your message handler code here
    VARIANT variant_inp;
    COleSafeArray safearray_inp;
    LONG len, k;
    BYTE rxdata[2048]; //Set BYTE array An 8-bit integer that is not signed.
    CString strtemp;
    unsigned char flag;

    if (m_ctrlMscomm1.get_CommEvent() == 2 && m_ifCli == 2)
    {
        Sleep(1000);
        variant_inp = m_ctrlMscomm1.get_Input();     //Read buffer
        safearray_inp = variant_inp;               //Transfer variant to ColeSafeArray
        len = safearray_inp.GetOneDimSize();       //Get data length
        for (k = 0; k<len; k++)
            safearray_inp.GetElement(&k, rxdata + k);
        for (k = 0; k<len; k++)
        {
            BYTE bt = *(char*)(rxdata + k);
            strtemp.Format(_T("%c"), bt);
            char ch = (char)bt;
            if (ch == '!')
            {
```

Figure 45 Codes of MsComm

3.3 Simple simulation of low-pass filters

The ADF4350 can generate sinusoidal signals between 137.5MHz to 4.4GHz by an integrated voltage-controlled oscillator VCO. There is a 1/2/4/8/16/32/64 divider circuit to help generate frequency in different range. However, because of the frequency divider, the waveform of ADF4351 in high frequency is square wave which is the combination of the sinusoidal wave of the desired frequency and harmonic waves. To reduce the harmonic waves and generate required sine wave, low-pass filters need to be used in the system.

The expression of square wave within a period:

$$f\left(t\right)=\begin{cases}-\frac{E}{2}, & -\frac{T}{2}\leq t<0\\ \frac{E}{2}, & 0\leq t\leq\frac{T}{2}\end{cases} \tag{10}$$

Fourier coefficients:

$$a_n=\frac{T}{2}\int_{-\frac{T}{2}}^{0}\left(-\frac{E}{2}\right)\cos n\omega_0 tdt+\frac{T}{2}\int_{0}^{\frac{T}{2}}\left(\frac{E}{2}\right)\cos n\omega_0 tdt=\frac{E}{n\omega_0 T}\left[\left(-\sin n\omega_0 t\right)|_{-\frac{T}{2}}^{0}+\left(\sin n\omega_0 t\right)|_{0}^{\frac{T}{2}}\right]=0 \tag{11}$$

$$b_n=\frac{T}{2}\int_{-\frac{T}{2}}^{0}\left(-\frac{E}{2}\right)\sin n\omega_0 tdt+\frac{T}{2}\int_{0}^{\frac{T}{2}}\left(\frac{E}{2}\right)\sin n\omega_0 tdt=\frac{E}{n\omega_0 T}\left[\left(\cos n\omega_0 t\right)|_{-\frac{T}{2}}^{0}+\left(-\cos n\omega_0 t\right)|_{0}^{\frac{T}{2}}\right]= \frac{E}{2\pi n}\left[2-2\cos\left(n\pi\right)\right] \tag{12}$$

Thus,

$$b_n=\begin{cases}\frac{2E}{n\pi}, & n \text{ is odd}\\ \text{X}, & n \text{ is even}\end{cases} \tag{13}$$

The Fourier expansion:

$$f\left(t\right)=\frac{2E}{\pi}\left(\sin\omega_0 t+\frac{1}{3}\sin 3\omega_0 t+\frac{1}{5}\sin 5\omega_0 t+...+\frac{1}{n}\sin n\omega_0 t\right) \tag{14}$$

According to equations above, square wave is the combination of the sinusoidal wave of the desired frequency and harmonic waves. The frequency of the harmonic waves is higher than the desired frequency. Thus, low pass filters can be used to estimate the harmonic wave. Due to the large range of frequency which is from 800MHz to 2700MHz, several filters are used to cover it. In this part, I mainly simulate the filters with cut-off frequency of 1.1GHz and 2.2GHz. In order to simulate filters, I used the Advanced Design System (ADS).

The following figure shows the structure of the circuit. I built a schematic and added a S-parameter, a DA_LCLowpass Filter and two 50 Ohm loads.

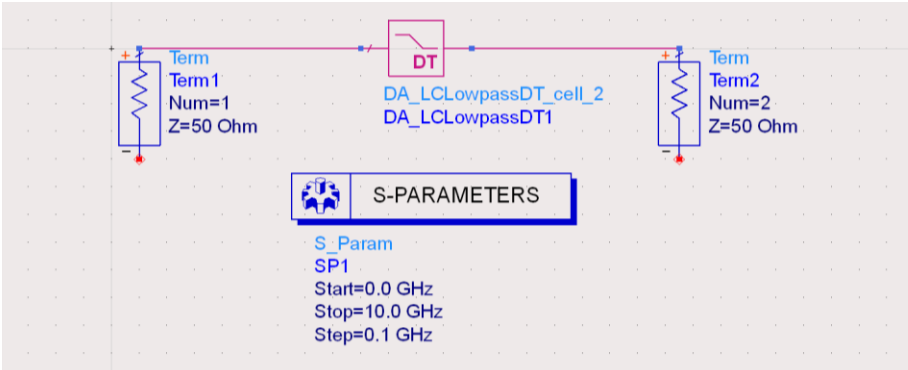

Figure 46 Filter circuit

Then, I set the parameters of the filter by using “Filter DesignGuide” where I created a Chebyshev filter with 1100MHz “Fp” and 1430MHz “Fs”. The order turned out to be 5.

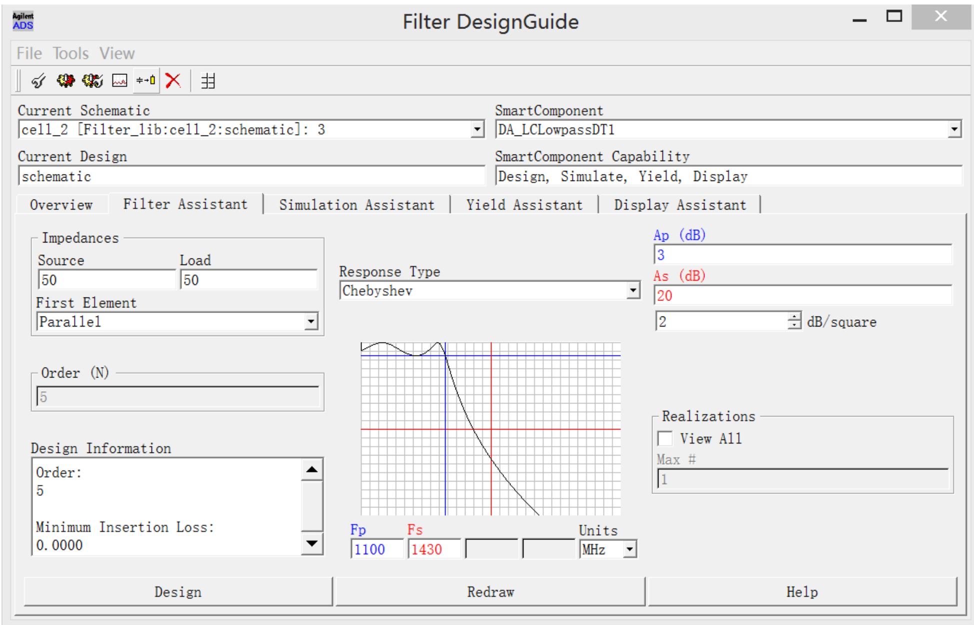

Figure 47 Filter design guide for 1.1GHz

Then I transfer the devices into capacitors and inductance.

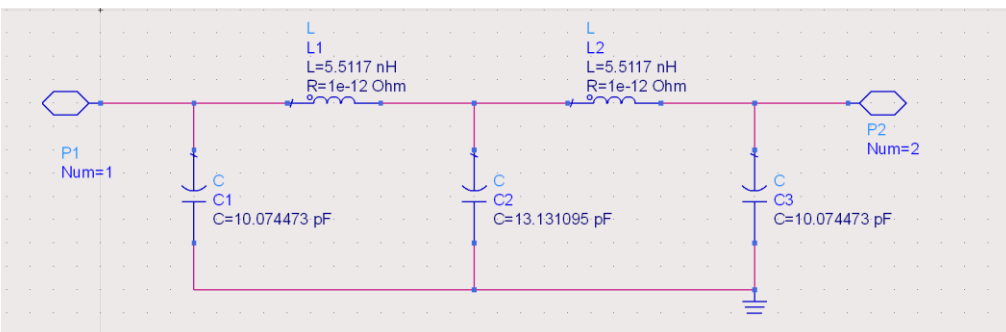

Figure 48 Circuit of filter with 1.1GHz cut-off frequency

The following figure is the simulation result of low-pass filter with 1.1G cut-off frequency.

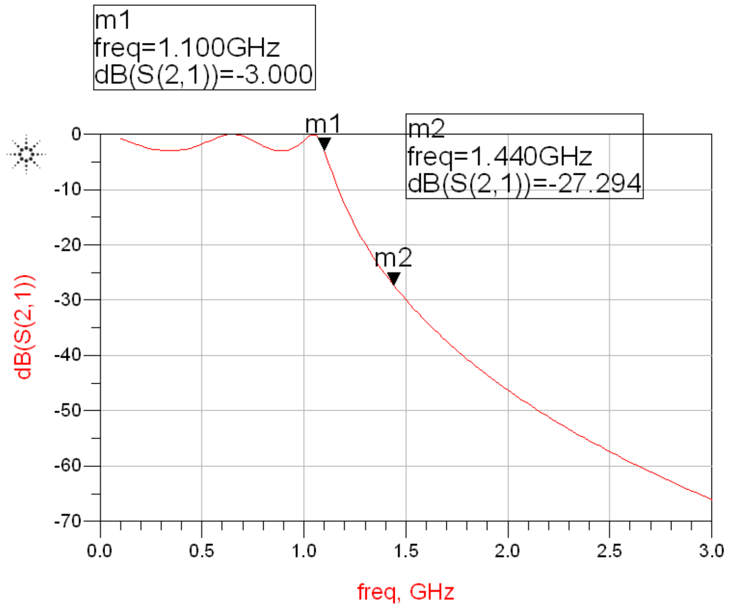

Figure 49 Simulation result of filter with 1.1GHz cut-off frequency

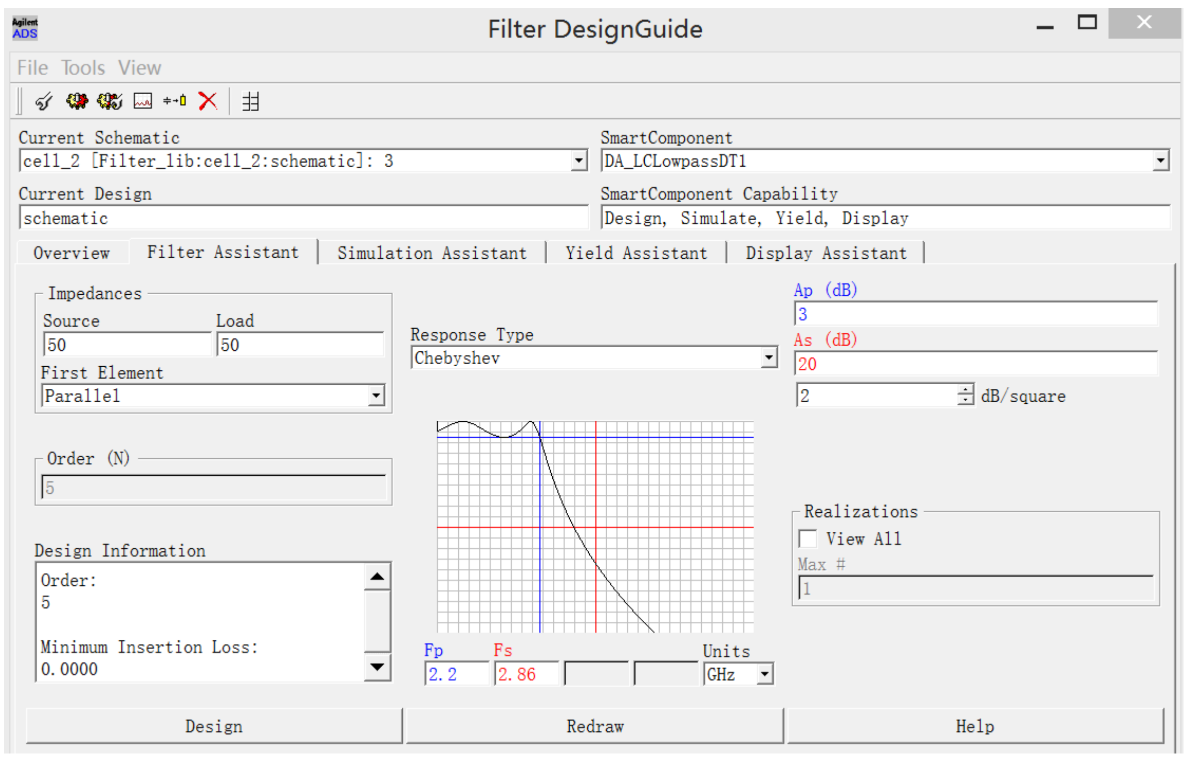

Figure 50 Filter design guide for 2.2GHz

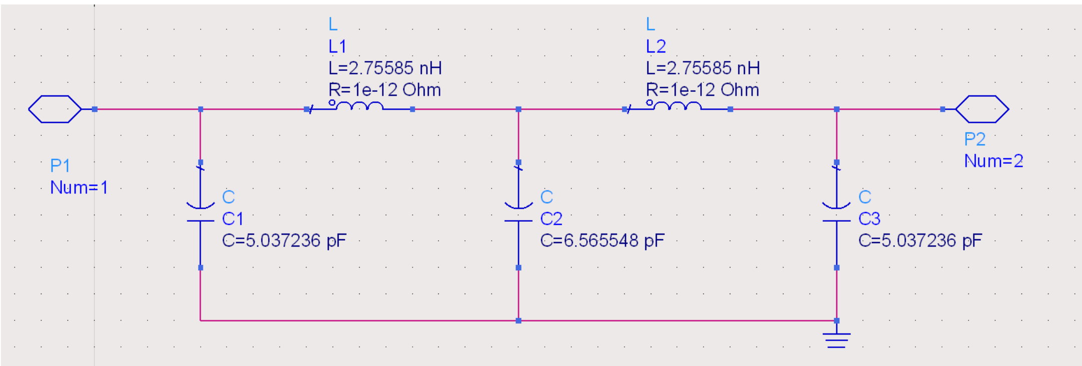

Figure 51 Circuit of filter with 2.2GHz cut-off frequency

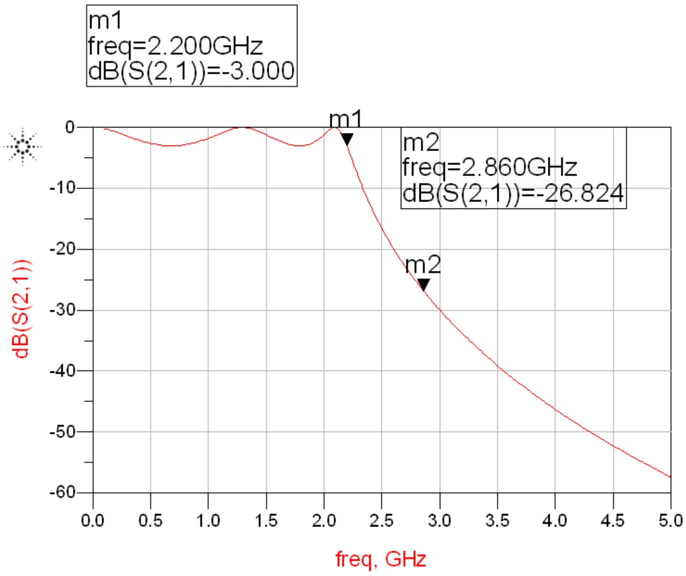

Figure 52 Simulation result of filter with 2.2GHz cut-off frequency

Chapter 4: Results and Discussion

4.1 Frequency generation

The ADF4350 PLL synthesizer is capable of producing frequency up to 4.4 GHz. In this project, the frequency can be only generated from 800MHz to 2.7GHz because of the bandwidth limits of hardware devices.

4.2 Detection

The results of the project contain several parts.

To detect the accuracy of 3dB hybrid and 3dB power divider, I connect the output signals of 3dB power divider into one AD8302, which is “output1” in the following figure. Meanwhile, the outputs of 3dB power divider are connected to the other AD8302, the result of which is displayed as “output2”. Ideally, the values should be 0, 0, 90, and 0. However, due to reflections and losses, the result cannot be perfect. Thus, calibrations are necessary to improve the performance of the system.

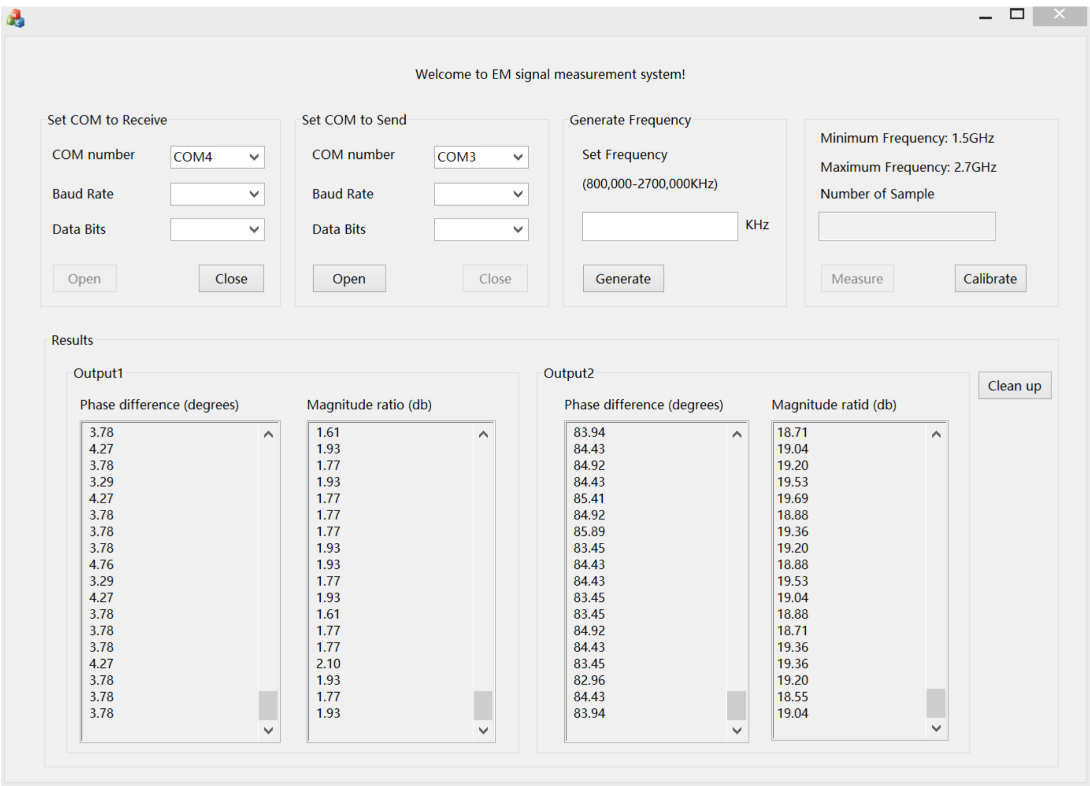

Figure 53 Result 1

After calibration, the result is correct in the allowance of errors.

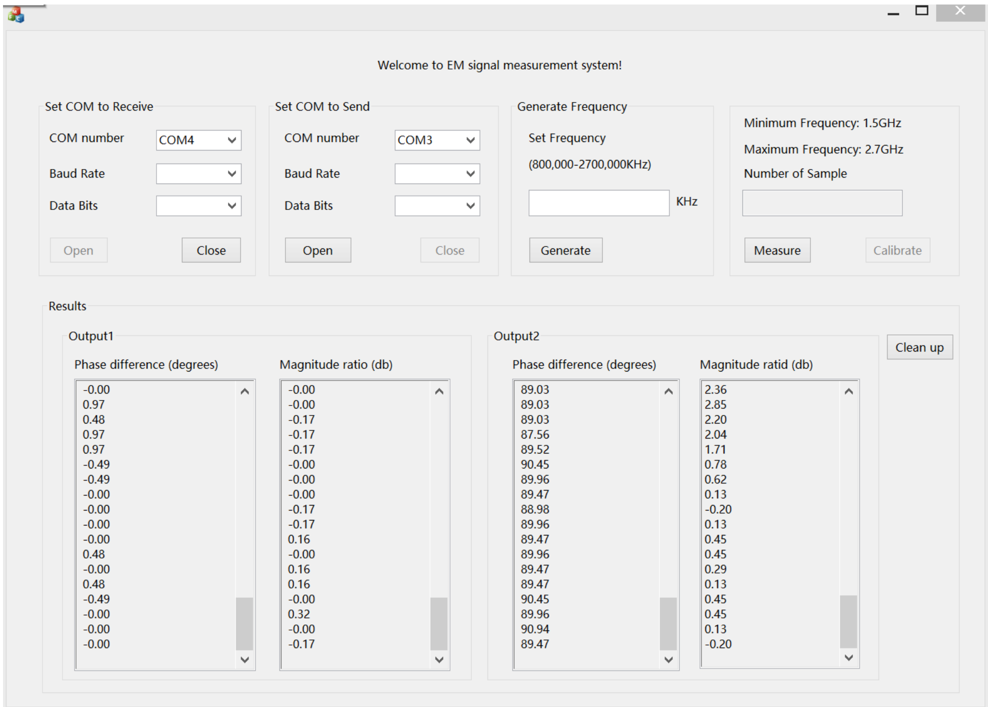

Figure 54 Result 2

The result of EM measurement system with no circulator.

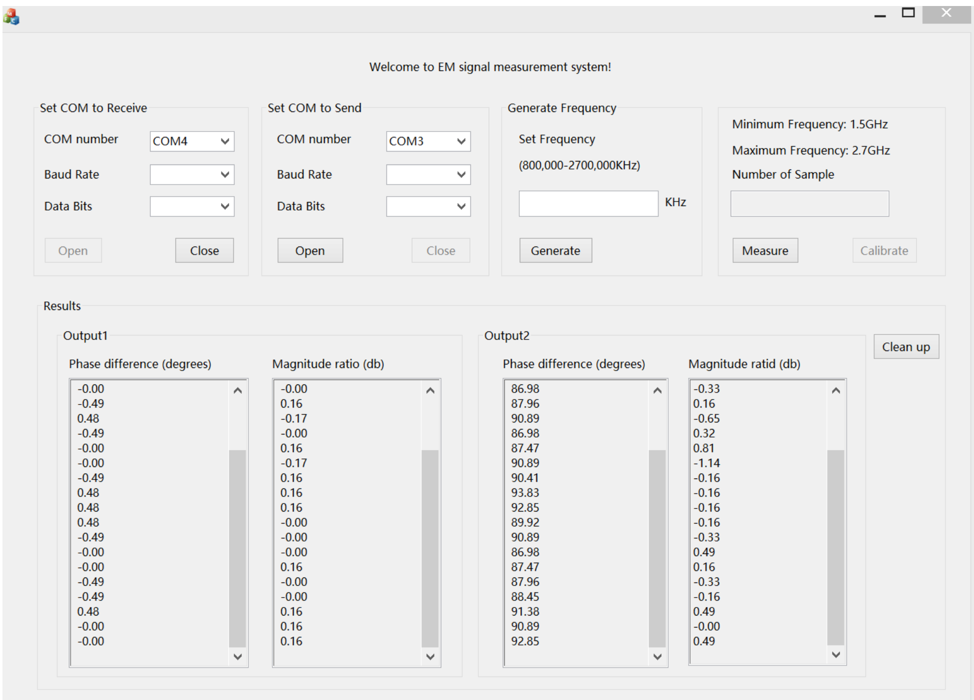

Figure 55 Result 3

The calibration result of completed EM signal measurement system.

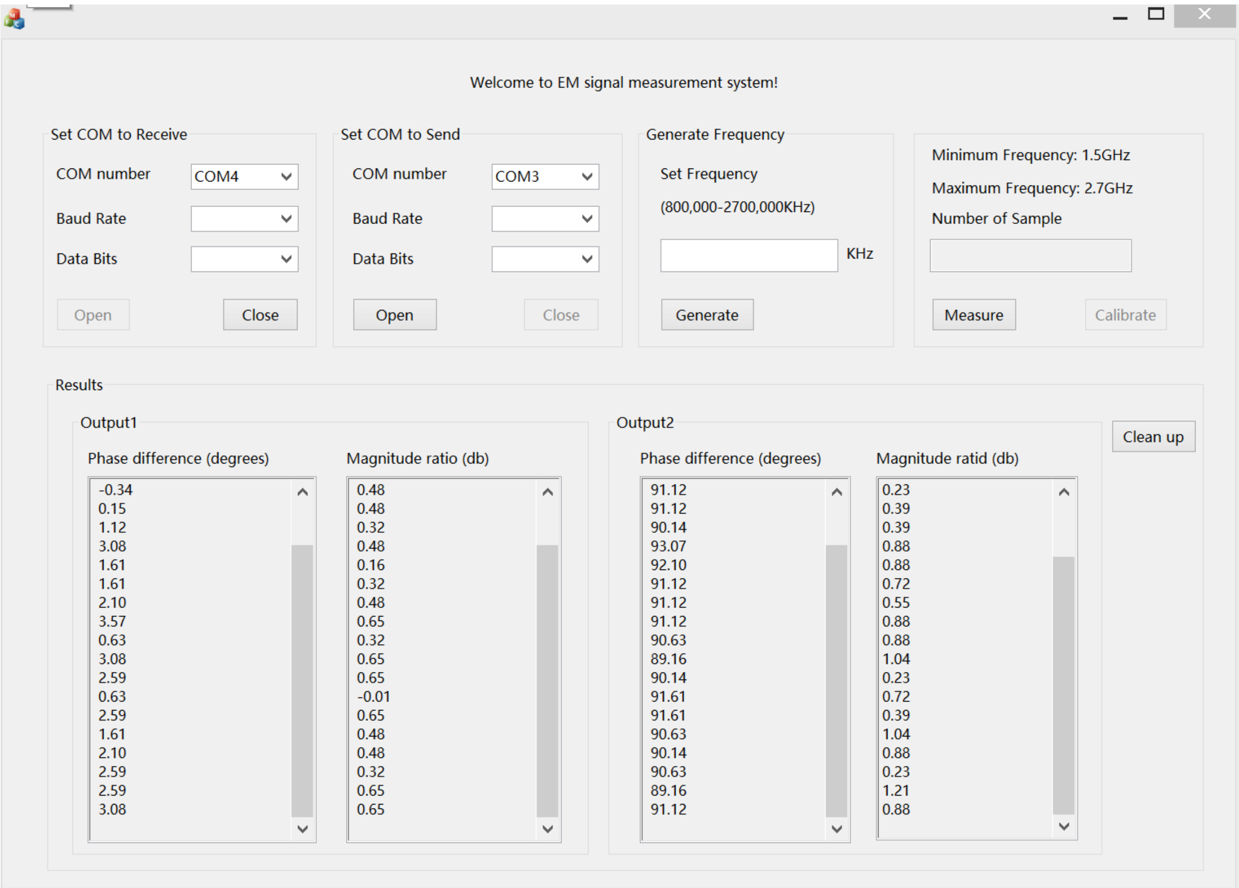

Figure 56 Result 4

Chapter 5: Conclusion and Further Work

In this report, I discussed the project in detail including the background of the EM system, its design and implementation, the results and future work. The background mainly introduced the necessary knowledge about improved measurement method and the principle of key devices which are ADF4350, STC8051 microcontroller, AD8302 detector, 3dB power splitter and 3dB hybrid. In the implementation part, the connection of devices is introduced. Each key step is discussed which includes hardware implementation and GUI design. By this part, the whole system is introduced in detail. For the Results and Discussion part, the measurement results and calibration results are displayed.

I have finished the task of validation and improvement on existing measurement methods (based on AD8302). I read papers about AD8302, ADF4350, 3dB Hybrid and the existing EM measurement system. I used two AD8302 chips, a 3dB splitters and a 3dB 900 hybrid to build an initial prototype. I have detected the phase difference and amplitude of signals from 800MHz to 2500MHz. Besides, to make sure that the 3dB splitters and the hybrid work well, I detected the two devices and the output was correct within the allowable range of error.

I also designed a circuit to provide sinusoidal signals across a band of frequencies. At first, I tried to use an Arduino board to control the ADF4350 to generate the frequency. However, the Arduino board is not fast enough. To improve the efficiency and make the system more portable, I found a development suit with a STC8051 microcontroller (STC15F2K60S2) and an ADF4350 which can generate frequency between 137.5MHz to 4.4GHz. Then, I read papers about ADF4350 and STC8051 chip. I programmed the suit under the Keil4 environment to generate frequency up to 2.7GHz.

To make the measurement easy to conduct, I also developed a software interface to control drive the frequency generation process and to detect signals. I designed the GUI using C++. According to the bandwidth limits of hardware, this software allows users to set the frequency between 800MHz to 2700MHz, which sends data to a certain USB port connected to the ADF4350 development suit mentioned above. This software can also receive the data sent from a certain USB port by a microcontroller and display the phase differences and the amplitude differences on the screen.

To validation the final design, I connected the hardware and the software to make the whole system work well. To make the measurement more accurate, I also tested the devices separately and make calibrations.

For future work, there are several aspects which can be improved further. First, as this system uses two computer ports to connect with the software, the system needs two microcontrollers. To make the system more portable and efficient, there can be only one microcontroller to rend and receive data from one certain computer port. Second, the phase shift can be improved by using a programmable module, which can change the phase of signal to a range of values, making the system flexible.

References

[1] Analog Devices (2002). *AD8302*. [ebook] Available at: http://www.analog.com/cn/products/rf-microwave/rf-power-detectors/non-rms-responding-detector/ad8302.html#product-reference [Accessed 22 May 2017].

[2] Qihang Zhang (2017). *Design and implementation of an EM measurement system.*

[3] Qianyu Li (2017). *Design and implementation of a dielectric measurement system.*

[4] B. García-Baños1, José M. Catalá-Civera2, Felipe Peñaranda-Foix2, Antoni J. Canós2 and Oscar Sahuquillo Navarro3 (2012). *Microwave Sensor System For Continuous Monitoring of Adhesives Curing Process*. Available at: http://dx.doi.org/10.1088/0957-0233/23/3/035101

[5] Sam Wetterlin (2007). *Use of Phase Shift to Resolve Sign Ambiguity and Improve Accuracy in the AD8302 Phase Detector.*

[6] U.S.A: Analog Devices, Inc. (2017), *Wideband Synthesizer with Integrated VCO Data Sheet ADF4350*. [ebook] pp.1, 5, 11-18. Available at: http://www.analog.com/cn/products/rf-microwave/pll-synth/fractional-n-plls/adf4350.html [Accessed 15 Jul. 2017].

[7] ADI CAST (China Application Support Team) (2009). *ADI ADF4350 Driver Test Code.*

[8] https://www.microwaves101.com/encyclopedias/hybrid-couplers

[9] https://en.wikipedia.org/wiki/Power_dividers_and_directional_couplers

[10] http://www.atmmicrowave.com/coaxial/coupler/hybrid-90-directional/

[11] Analog Devices (2017). *ADF4350*. [ebook] Available at: http://www.analog.com/en/products/clock-and-timing/phase-locked-loop/fractional-n-pll/adf4350.html

[12] Lu Wenhua (2014), *Common used DSS control board*

[13] STC micro (2013), *STC15F2K60S2 datasheet*

Acknowledgement

In the process of completing my project, I have received a lot help from my supervisor and my friends. Here I would like to thank them sincerely for their concern, patience and encouragement.

Appendix

北 京 邮 电 大 学 本科毕业设计（论文）任务书
Project Specification Form

学院 School	International Scho	专业 Programme	Telecom (H6N2)	班级Class	2014215107
学生姓名 Name	WANG Yinuo	学号 BUPT student no	2014213051	学号 QM student no	140920440
设计（论文）题目 Project Title	Development of EM signal measurement system				
论文题目（中文）	电磁场信号测量系统的实现				
题目分类 Scope	Implementation	Antennas and Optical	Software and Hardware		

主要任务及目标 Main tasks and target	By
Task 1: Validation and improvements on existing measurement methods (based on AD8302 chip)	February 2018
Task 2: Design a circuit to provide sinusoidal signals across a band of frequencies	March 2018
Task 3: Develop a software interface to control the frequency generation and to detect signals	April 2018
Task 4: Validate the final design	April 2018

主要成果 Measurable outcomes

1) Validated software and hardware designs of signal generation and signal detection

2) Functional system (software and hardware) of the measurement system

3) Comparisons between the measured results with simple simulations to validate the final outcome

主要内容 Project description

Full characterisation of EM circuits or designs can be achieved by the ability to accurately measure the amplitude and phase of the detected signals. The aim of this project is to design and implement an EM measurement system to provide a means to measure the phase and amplitude of signals operating at frequencies up to 2.7GHz (and beyond). The successful candidate will be analysing and improving previous designs and on completing a SW+HW system to achieve the required targets. The design should be validated by using an appropriate simulation tool or through comparisons with measurements from a network analyser. By the end of the project, the candidate should produce a complete functional measurement system on a PCB board.

概述 Project outline

Development of an EM measurement system can help determining the characterization of EM circuits by accurately measuring the difference of phases and the difference of amplitudes of two signals up to 2.7GHz. My work is to improve the previous hardware system and build the software, implementing a complete functional EM measurement system on the PCB board.

There are mainly 4 tasks in my project. Task 1 is validation and improvement on existing measurement methods (based on AD8302); task 2 is to design a circuit to provide sinusoidal signals across a band of frequencies; task 3 is to develop a software interface to control drive the frequency generation process and to detect signals; task 4 is to validation the final design.

The existing given system can approximately detect voltages and phases across two ports by providing a certain voltage output, without determining the direction of the phase difference. I will use a new method, which will require another AD8302 chip, to identify whether the phase difference is a negative number or a positive one. Besides, to generate the signal of frequency up to 2.7GHz, I will find a suitable frequency generation method (chip) and implement the method to provide wideband sinusoidal signal.

As for the software, I will write a GUI to make users easily to handle the measurement and get the results directly. The GUI will be a C++ program under the environment of VS 2017, which will contain mainly two functions. The first one is to generate the frequency by entering the number to the software. The second one is to receive the data sent from a certain port and display the phase differences and the amplitude differences on the screen. Furthermore, because a new Arduino board will be used, the program in it needs to be written to receive data from port and generate the signal required.

In the end, the system should be an implemented one, and the software and hardware need to work harmoniously with each other to get the correct result of phase difference and amplitude difference of the two signals. Different EM circuits and devices will be needed to detecting by the system to test its accuracy.

中期目标 What I expect to have working at the mid-term oral

1. Improve the existing measurement methods and implement a hardware circuit that can distinguish the phase difference by using two AD8302 chips.
2. Find a suitable frequency generation method (chip), and prepare the hardwares to implement the method to provide wideband sinusoidal signal.
3. Design a primary GUI that can display data read from computer port.

Fill in the sub-tasks and select the cells to show the extent of each task

	Nov		Dec		Jan		Feb		Mar		Apr		May	
Task 1: Validation and improvements on existing measurement methods (based on AD8302 chip)														
Literature review		■	■	■										
Finding suitable components to build an initial prototype		■	■											
Implement the hardware circuit		■	■	■		■	■							
Test the accuracy of the measurement						■	■							
Task 2: Design a circuit to provide sinusoidal signals across a band of frequencies														
To find a siutable frequency generation method (chip)			■	■		■	■	■						
To implement the method to provide wideband sinusoidal signal							■	■	■					
To test the method and validate it							■	■	■					
Task 3: Develop a software interface to control the frequency generation and to detect signals														
Design a suitable GUI		■	■	■		■	■	■	■	■				
Write the code to read data from computer port		■	■	■		■								
Write the code to control the frenquency of sinusoisal signal							■	■	■	■				
Implement calibration code for the measurement setup									■	■				
Task 4: Validate the final design														
Integrate the software with hardware						■	■	■	■	■	■			
Test the system						■	■	■	■	■	■			
Wirte and submit final report									■	■	■	■		

北 京 邮 电 大 学

BBC6521 Project 毕业设计 2017/18

Early-term Progress Report

初期进度报告

<table>
<tr><td>学院
School</td><td>International School</td><td>专业
Programme</td><td>Telecommunications Engineering with Management (H6N2)</td><td>班级
Class</td><td>2014215107</td></tr>
<tr><td>学生姓名
Student Name</td><td>Yinuo Wang</td><td>BUPT 学号
BUPT Student No.</td><td>2014213051</td><td>QM 学号
QM Student No.</td><td>140920440</td></tr>
<tr><td>QM 电子邮件
QM Email</td><td colspan="2">yinuo.wang@se14.qmul.ac.uk</td><td>BUPT 电子邮件
BUPT Email</td><td colspan="2">wangyinuo@bupt.edu.cn</td></tr>
<tr><td>设计（论文）题目
Project Title</td><td colspan="5">Development of EM signal measurement system</td></tr>
<tr><td colspan="6">已完成工作：
Finished Work：
Read papers about AD8302, ADF4350 and EM measurement system.
Find suitable components to build an initial prototype.
Connet the components with 3dB hybrid and test the phase differences.
Design a primary GUI and find proper methods to achieve its functions.

Problem:
The phase difference of the current circuit is not accurate.</td></tr>
<tr><td colspan="3">是否符合进度？On schedule as per GANTT chart?</td><td colspan="3">[YES/NO]</td></tr>
</table>

1

下一步：
Next steps:

[Focus on the next **immediate** steps, i.e. until the next report – DELETE THIS LINE]

Improve the accuracy of the measurement

Achieve the functions of GUI which should receive and send data via computer port.

Find a suitable frenquency generation method (a chip).

2

北 京 邮 电 大 学

BBC6521 Project 毕业设计 2017/18

Mid-term Progress Report

中期进展情况报告

学院 School	International School	专业 Programme	Telecommunications Engineering with Management (H6N2)	班级 Class	2014215107
学生姓名 Student Name	Yinuo Wang	BUPT 学号 BUPT Student No.	2014213051	QM 学号 QM Student No.	240920440
QM 电子邮件 QM Email	yinuo.wang@se14.qmul.ac.uk		BUPT 电子邮件 BUPT Email	wangyinuo@bupt.edu.cn	
设计（论文）题目 Project Title	Development of EM signal measurement system				

毕业设计（论文）进展情况，字数一般不少于 1000 字
The progress on the project. Total number of words is no less than 1000

目标任务: Targets set at project initiation:

(must be the same as "What I expect to have working at the mid-term oral" in the Spec)

1. Improve the existing measurement methods and implement a hardware circuit that can distinguish the phase difference by using two AD8302 chips.
2. Find a suitable frequency generation method (chip), and prepare the hardware to implement the method to provide wideband sinusoidal signal.
3. Design a primary GUI that can display data read from computer port.

是否完成目标 Targets met?	[YES/NO] NO

目前已完成任务 Finished Work:

Validation and improvement on existing measurement methods (based on AD8302)

I have read papers about AD8302, ADF4350, 3dB Hybrid and EM measurement system. The existing given system can approximately detect voltages and phases across two ports by providing a certain voltage output, without determining the direction of the phase difference. I will use a new method, which will require another AD8302 chip, to identify whether the phase difference is a negative number or a positive one.

1

According to the literature review, I have found suitable components to build an initial prototype including STC12LE5A60S2, AD8302, ADF4350 PLL, LCD1602, Arduino UNO R3, 3dB hybrid, 800-2500MHz 5dB power splitter, 800-2500MHz, and 50ohm SMA Load. Then I built a prototype. In order to improve the accuracy of the measurement, I use 3dB Hybrid and two AD8302 chips. By using a 3dB hybrid, I detected the phase difference which is 90 degrees at around 1800MHz. To make 90 degrees phase shift, I detect the phase difference of two output signal and the result is -180. After I connected the 3dB Hybrid to one signal, the result becomes -90 degrees.

Develop a software interface to control drive the frequency generation process and to detect signals

As for the software, I need to write a GUI to make users easily to handle the measurement and get the results directly. The GUI will be a C++ program under the environment of VS 2017, which will contain mainly two functions. The first one is to generate the frequency by entering the number to the software. The second one is to receive the data sent from a certain port and display the phase differences and the amplitude differences on the screen.

By now, I have designed a primary GUI and find proper methods to achieve its functions. I have written a GUI by C++. Now, I can use this software to set the frequency between 800MHz to 2700MHz. Now, I am working on the software part to receive the port data. I can roughly receive the data and show it in one edit box now.

Design a circuit to provide sinusoidal signals across a band of frequencies

To generate the signal of frequency up to 2.7GHz, I need to find a suitable frequency generation method (chip) and implement the method to provide wideband sinusoidal signal.

Firstly, I have found some small completed products. Basically, they use a microchip and a ADF4350, but the products cannot be programmed. So I write an Arduino program to generate frequency up to 2.7GHz. This program allows users to set the frequency by themselves through specific COM. And now I can use the C++ application to directly control the Arduino board rather than use the Arduino software.

2

尚需完成的任务 Work to do: Further improve the accuracy of the measurement and broaden the band wide. Find a better way to generate the frequency and implement it. Implement the GUI and achieve the function which is receiving the data sent from a certain port and displaying the phase differences and the amplitude differences on the screen. Validate the final design. In the end, the system should be an implemented one, and the software and hardware need to work harmoniously with each other to get the correct result of phase difference and amplitude difference of the two signals. Different EM circuits and devices will be needed to detecting by the system to test its accuracy.	
能否按期完成设计（论文） Can finish the project on time or not:	[YES/NO] YES
存在问题 Problems: The phase difference of the current circuit is not accurate. A more efficient way of generating frequency need to be found.	
拟采取的办法 Solutions: To make the phase shift more accurate and cover more band wide, I will further test each part in detail and adjust the hardware circuit. For the frequency generation, I have used an Arduino board to control the ADF435x evaluate board. In order to be more efficient, I will try to program the STC12LE5A60S2 micro controller and find other smaller development suit at the same time.	
最终论文结构 Structure of the final report: Abstract	

3

Chapter 1: Introduction

1.1 Project Context and Motivation

1.2 Project Objectives

1.3 Project Tasks

This part will mainly introduce the overall project and talk about its objectives and tasks.

Chapter 2: Background

2.1 Existing EM measurement system

In this part, I will introduce the given EM measurement system based on the previous materials.

2.2 ADF4350

2.3 AD8302

2.4 Low-pass filter

2.5 3dB Splitter and 3dB Hybrid

2.6 Improved method for EM measurement System

In 2.6, I will mainly introduce the principle of the improved method which is the key to this project.

In this chapter, I will give a brief description to each part and mainly discuss its functions in this project.

Chapter 3: Design and Implementation

3.1 Frequency Generation

3.2 Implementation of hardware circuit

3.3 GUI Design

In this chapter, I will show the structure of each part and introduce the way of implementation. I will also give an overall structure of the system.

Chapter 4: Results and Discussion

4.1 Frequency generation

4.2 Wave detection

4

<table>
<tr><td colspan="2">In chapter 4, I will show the result of each part.

Chapter 5: Conclusion and Further Work
5.1 Conclusion
5.2 Further Work

References
Acknowledgement
Appendix
Risk Assessment
Environmental Impact Assessment</td></tr>
<tr><td>日期 Date:</td><td>3/2/2018</td></tr>
</table>

5

Risk Assessment

Several risks may occur in the process mentioned in the beginning of chapter 3.

The ADF4350 is an ESD (electrostatic discharge) sensitive device. Damages may occur on device due to high energy ESD. To avoid performance degradation and loss of function, proper precautions should be taken. Moreover, a lot heat can be produced when ADF4350 is working. High temperature can do harm to the chip. The following table shows some risks.

Table 2: Risk assessment

Description of risks	Description of impacts	Likely hood rating	Severity of the risk	Preventive methods
ESD	Performance degradation and loss of function	4	4	Proper precautions should be taken to avoid this risk.
High temperature	Heat may damage the chip.	3	4	Pay attention to the measurement and shorten the time used.
Software crash	The GUI stops working.	2	2	Close the COM which receives data from Arduino.
Wrong connection of wire	The wires between AD8302 and Arduino are connected wrong. The Arduino stops working.	2	3	Check the connection of the circuit before measurement.

Environmental Impact Assessment

As all the units of the EM signal measurement system are highly integrated circuits and chips, the connection materials are wires and transmission lines, and ADF4350 synthesizer can produce electromagnetic waves, the system may leak and cause radiation pollution. Besides, the chips create heats when they operate, especially the ADF4350, causing pollution as well. The wires are made of plastic and copper which are not degradable. If they are carelessly abandoned, they will do harm to the environment.