Application Note
Jinhua Cao 912499916
TEAM ONE

I. Abstract

The baseband plays an important role in a radar system. The baseband part includes modulator, ADC, Low pass filter and Baseband-amplifier. The modulator generates the triangular wave, which will go through the VCO in the transmitter part. For the receiver, the signal from the mixer will be amplified by the amplifier and then go through a low pass filter, the signal is then sampled by the ADC and processed in the computer.

II. Design flow

Firstly, the baseband schematic is designed according to the application. In our system, it includes the regulator, the power amplifier and the filter. After the schematic obtained, the PCB layout is drawn, then after the fabrication, we need to solder and test the performance to see if the experimental result matches with the theory simulation.

In this design , the baseband schematic is first modelled in Kicad, then the layout is obtained and the PCB is fabricated from Bay Area Circuit. The schematic is shown in Fig. 1 The modulator is realized by using the Teensy 3.1 and DAC. The baseband receiver part is realized by the amplifier and low-pass-filter. Besides, to achieve the stable DC supply, the regulator is used to generate the 5V and 2.5V. The specified components used are listed in Table I. The top and bottom baseband PCB are shown in Fig. 2. The baseband is designed so that the RF board and baseband board could be stacked together to minimize the space as shown in Fig. 3. After the soldering, the final PCB is shown in Fig. 4. In Fig. 4, every component is soldered except for the 2.5V regulator since it does not work. It is very strange that the two regulators could not work at the same time. So in real test, we need two voltage supply. One is 5.3V which is supplied to the 5V regulator to generate the 5V and another is 2.5 V, which will be given directly to the components needed. In this case, no stable 2.5V is given, thus there may be noise from the power supply induced, therefore, it may have some influence on the performance of the RF board and the final radar imaging.

Table I: Lists of Baseband components

Components	Part number	Price	Number
Amplifier	TL974	$0	1
ADC	Sound Card	$0	1
Teensy	Teensy 3.2	$0	1
DAC	MCP4921	$0	1
Regulator_5V	LT1962	$3.52	1
Regulator_2.5V	LT1763	$4.78	1

What follows is the schematic diagram for the baseband as well as the PCB layout.

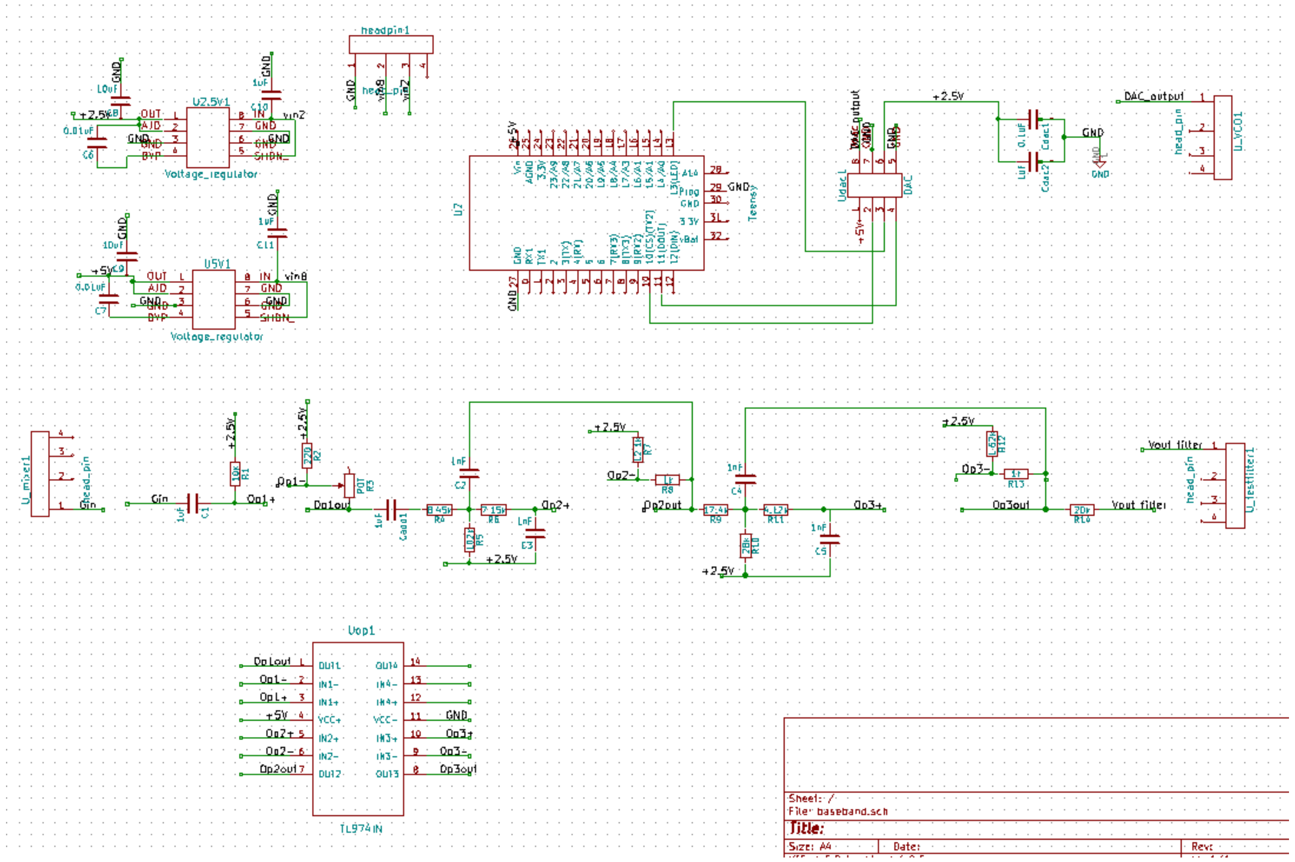
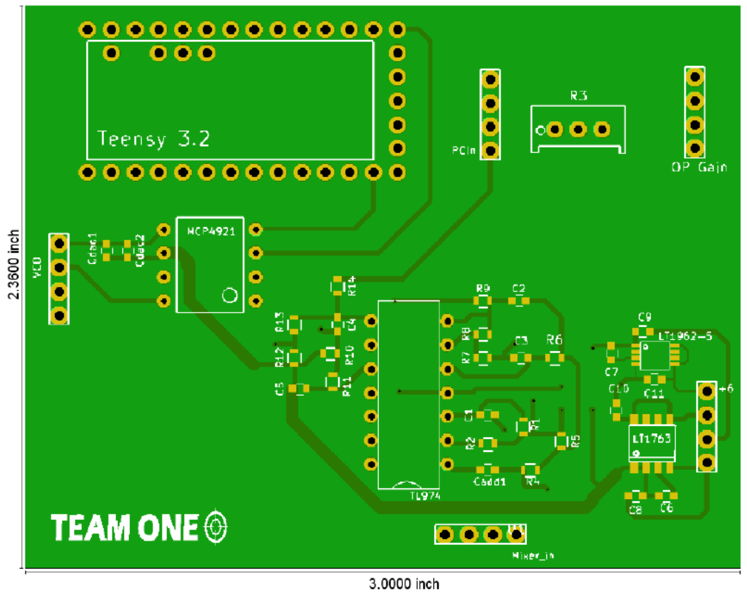
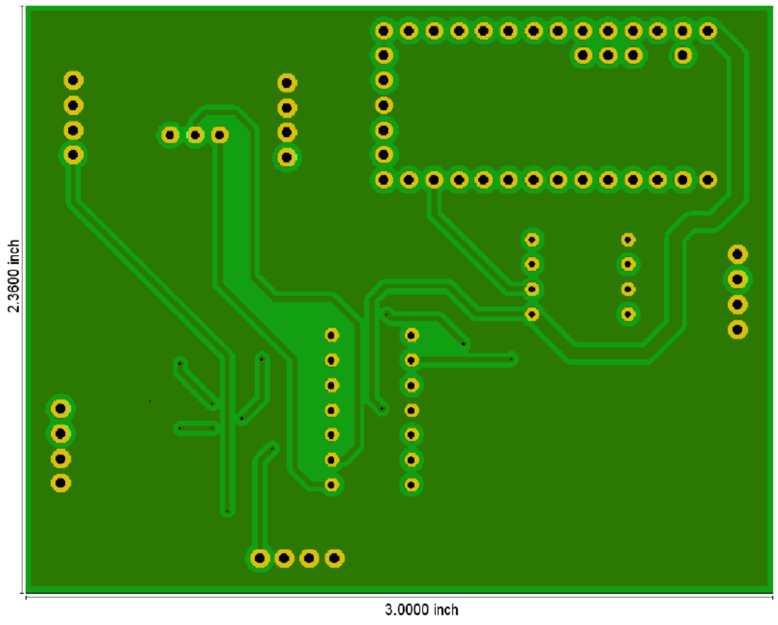


Fig. 1 The corresponding schematic design for the baseband board .



(a) Top view



(b) Bottom View

Fig. 2. The layout of the baseband. (top and bottom view)

The RF board and the Baseband boards are assembled in connection with one another so that the system is smaller and more compact as shown bellow

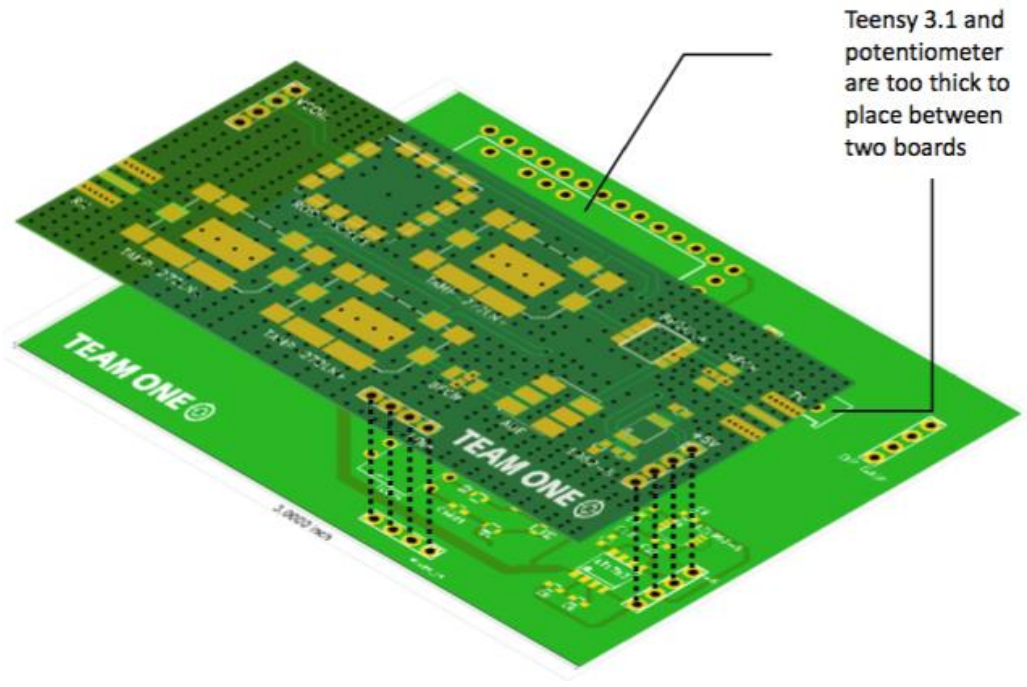

Teensy 3.1 and potentiometer are too thick to place between two boards
TEAM ONE
TEAM ONE

Fig. 3. The radar system layout.

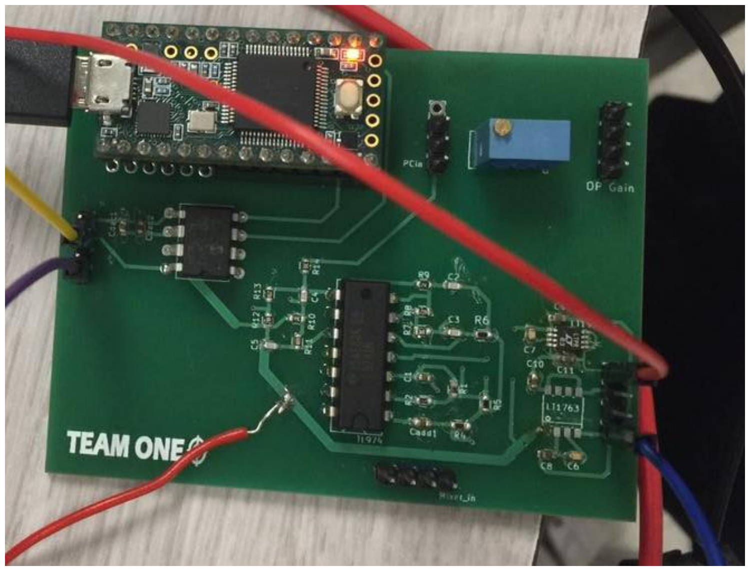


Fig. 4. The baseband PCB after soldering.

Test result of the Baseband

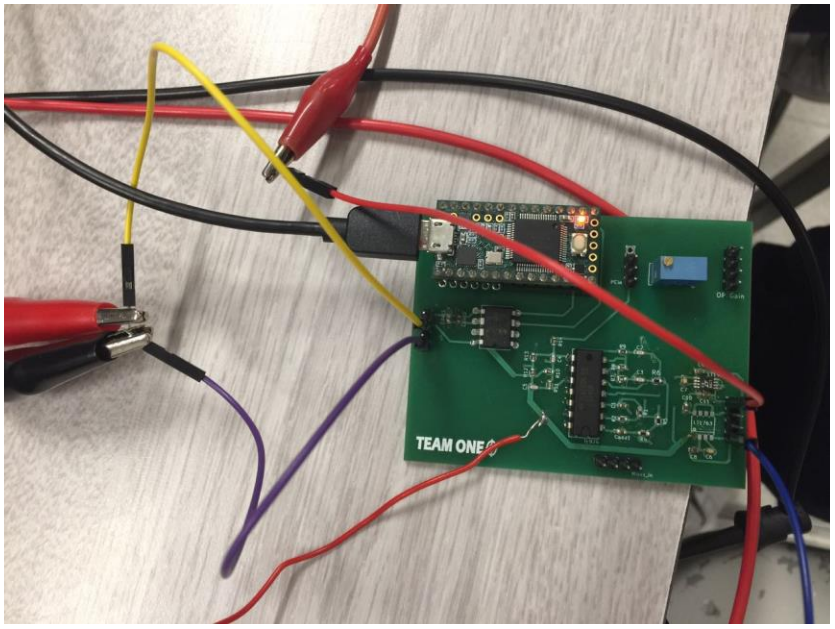


Fig. 5. The PCB in testament.

Part One: The testing setup for the transmitter part of the baseband

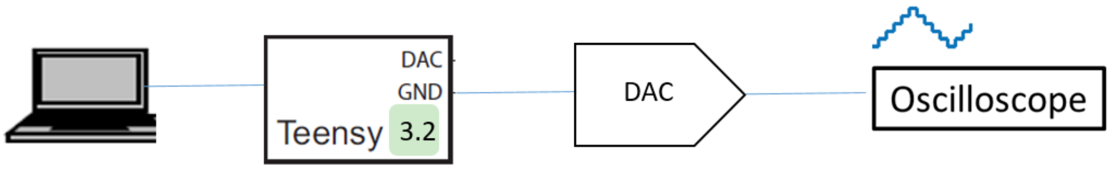


Fig.6 Test bench of the transmitter of the baseband.

The result for the DAC output is shown below, when the frequency is lower, we can get the triangular wave in Fig.6. When the frequency increases, the DAC output becomes the stair wave in Fig. 7. Besides, by changing the Arduino code, (The tesing code for the teensy is in the appendix.)the amplitude of the output could also be changed adaptively.

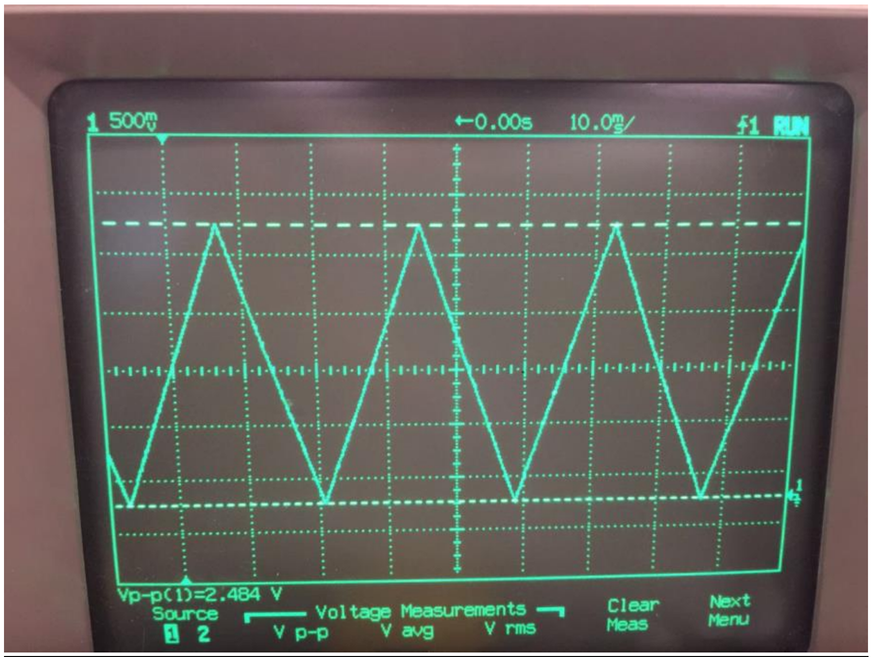

1 500mV
←0.00s 10.0ms/
f1 RUN
Vp-p(1)=2.484 V
Source
1 2
Voltage Measurements
V p-p
V avg
V rms
Clear Meas
Next Menu

Fig. 6.

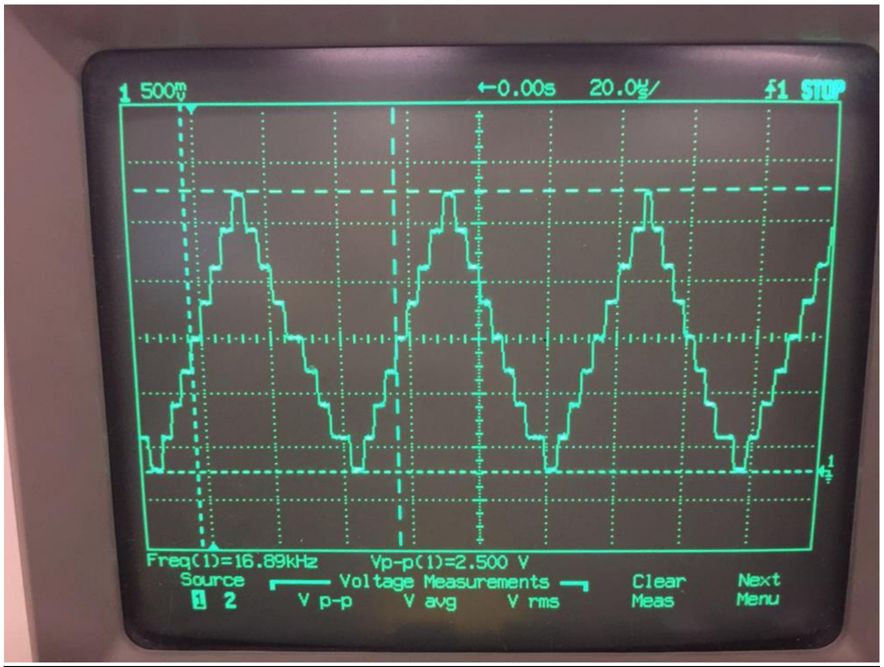

1 500mV
←0.00s 20.0µs/
f1 STOP
Freq(1)=16.89kHz
Vp-p(1)=2.500 V
Source
1 2
Voltage Measurements
V p-p
V avg
V rms
Clear Meas
Next Menu

Fig. 7.

Part Two: The testing setup for the receiver part of the baseband

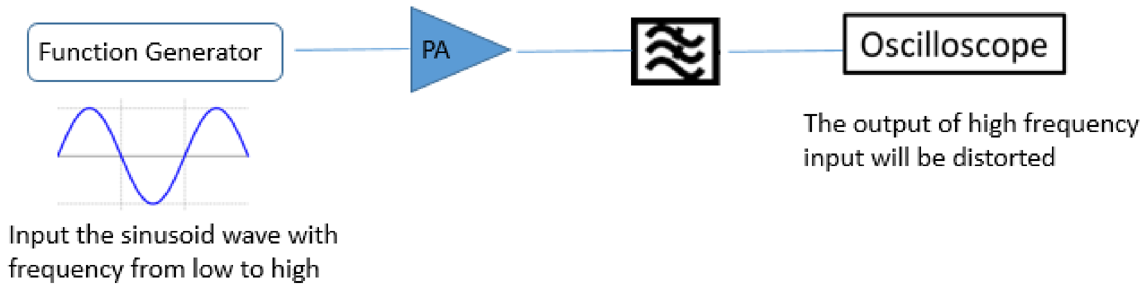
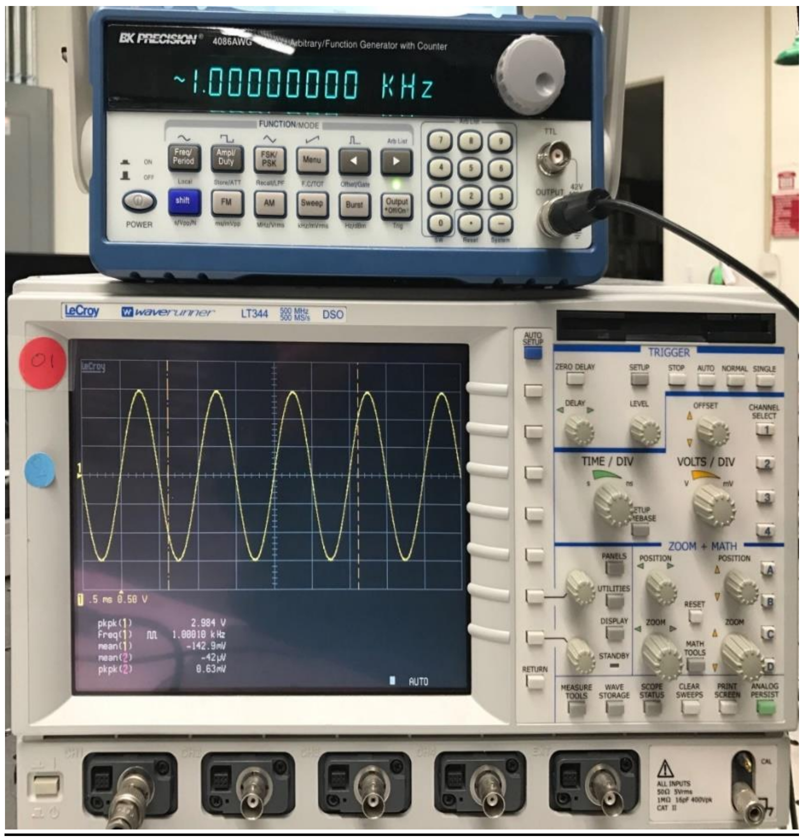

Function Generator
PA
Oscilloscope
The output of high frequency input will be distorted
Input the sinusoid wave with frequency from low to high

Fig.8 Test bench of the receiver of the baseband.

The output of the filter is shown as follows in Fig.9: when the input frequency is low, the output is the sine wave, the cutoff frequency of the filter is about 15Khz, where the output power is half of that of the low frequency.


BK PRECISION 4086AWG Arbitrary/Function Generator with Counter
~1.00000000 KHz
LeCroy waverunner LT344 DSO

```
1 .5 ms 0.50 V

pkpk(1)          2.984 V
Freq(1)  ⎍⎍   1.00010 kHz
mean(1)         -142.9mV
mean(2)           -42μV
pkpk(2)           0.63mV
```

Fig. 9. Filter output in low frequency (1KHz).

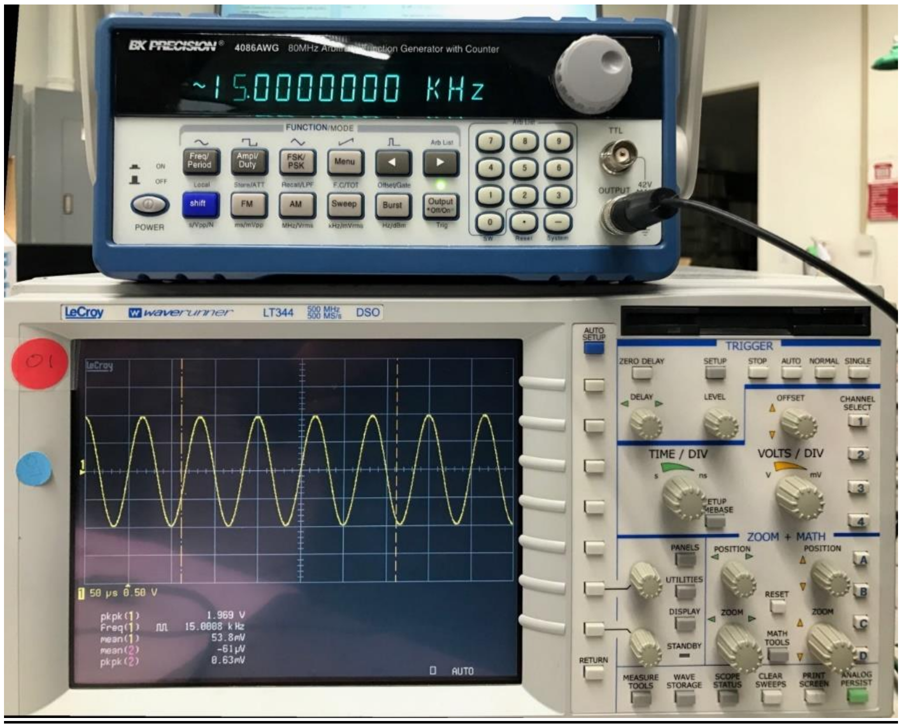


Fig. 10. Filter output in high frequency(15KHz).

1. The regulator test: During the test, the regulator for the 2.5V does not work when the 5V regulator works. The 2.5V regulator and 5V regulator could not work at the same time. To solve this problem, additional power supply voltage is used. The reason for this problem is still unknown. And this will be regarded as the future work. It is possibly that during the PCB layout, there is something wrong with the two regulator.

III. Conclusion and suggestion

1. The baseband sub-circuit works well except for the 2.5V regulator. The reason is still not clear. The suggestion for the following students is to use the voltage reference instead of using two regulators.
2. During the PCB layout:

   (1). The wire trace for the DC voltage could be drawn wider.

   (2). The position of the components could be adjusted to make the traces shorter so that the noise is smaller.

   (3). Surface mount components are recommended because they are easier to be de-soldered.

   (4). Adequate testing point should be added for critical signal lines.

Appendix

```
/*
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 =0; a <=4092; a=a+512)
  {
    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 = 4092; a >= 4; a = a-512)
  {
    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);
  }
}
```