Project Report

Ziyang Liu

Lab 1:

Overview:
Firstly, understand the mechanism about measure distance, it required the time-of-flight techniques.

Basically, its mechanism, first step send out an ultrasonic signal, and signal reflected back by the surface of any object. Then, a receiver will receive the reflected signal, and last micro-controller calculates the time difference between the sent and received signals.

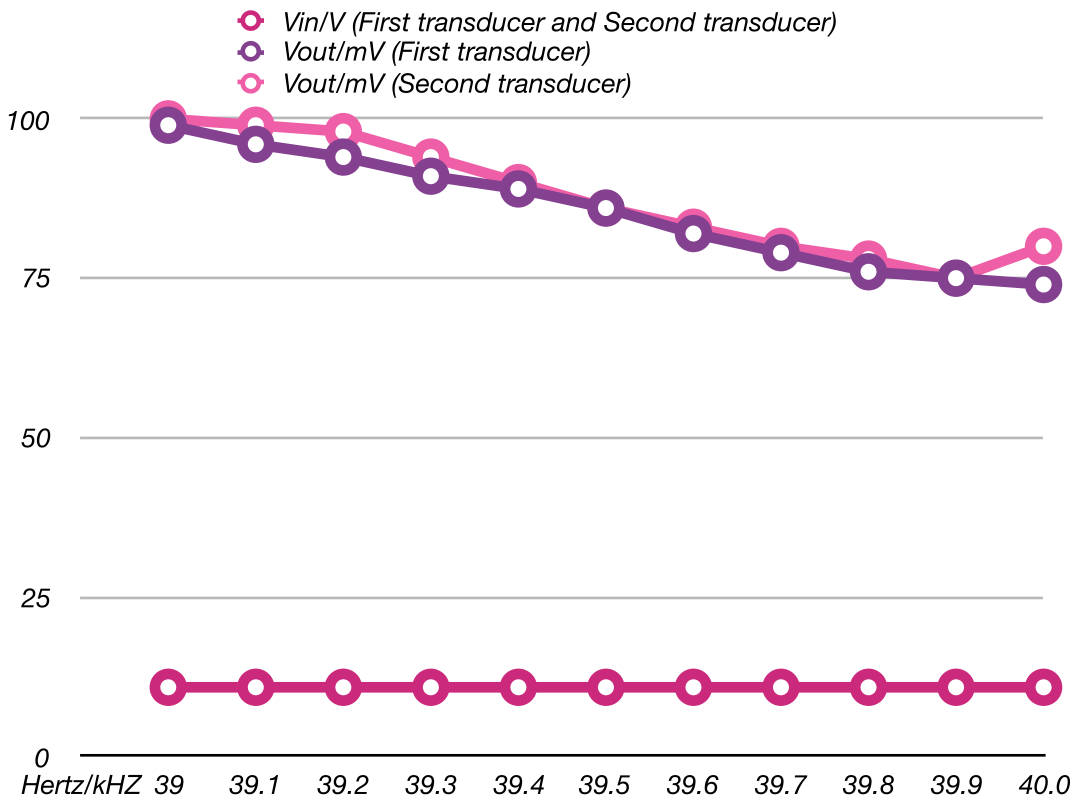
By using ultrasonic transducers to transmit and receive ultrasonic signals.

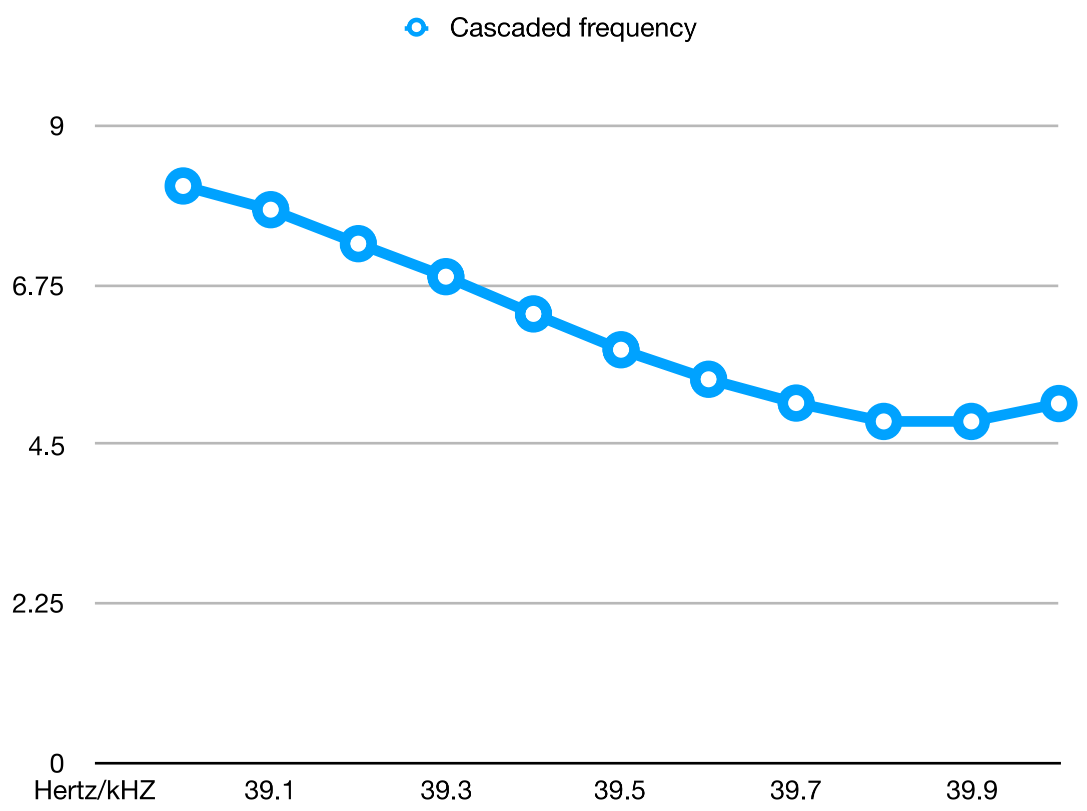
PC -> CPU(CC1350 Micro-controller) -> PWM -> Amplifier -> Ultrasonic Transmitter -> Ultrasound -> Obstacle -> Ultrasound Receiver -> Amplifier -> ADC -> PC

1. Characterizing ultrasonic transducer:

By using ultrasonic transducer to converts between electrical energy and kinetic energy, basically sound energy. It will receive an input signal and only output at a certain frequency, which is "fundamental frequency". In order to achieve this, feed the transducers with a frequency close enough to its fundamental frequency to produce a significant output.

For the first task, I have plot a diagram to demonstrate the data, in order to find their fundamental frequency, and calculate the results to get cascaded frequency:





Note: Cascaded frequency = (Vout/Vin) X (Vout/Vin)

From these data, it is clearly seen that the frequency started to increase around 40kHertz. In conclusion, fundamental frequency is close to 40kHertz.

Lastly, I increased and decreased the distance between the object and transducers, determine the change in the frequency response, and I found that the nature of air is closed to closed to resistor, it effects the signal transformation.

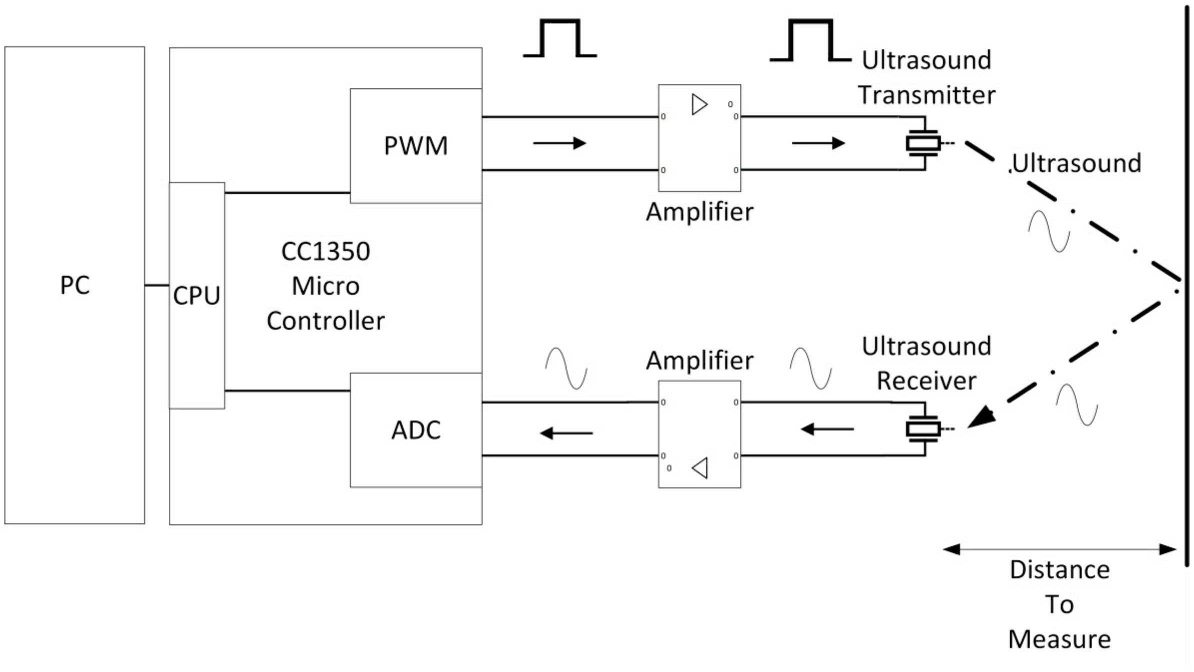
Lab 2:

In this lab, I built two amplifiers to amplify the signal from the two transducers.

Therefore, for a 10Vpp transmitter input, after transmission, receiver output voltage only 300mVpp. For another one, to amplify a 3.3Vpp signal to 10Vpp, so the Gai will be approximately 3. TL071 operational amplifiers used for the model.

For the transmitter amplifier, build a non-inverting amplifier with gain being 3, set positive supply to +15V and negative supply to -15V, signal generator used to simulate the input signal from controller, and set frequency to 40kHZ(fundamental frequency), waveform to square, duty cycle to 52%, amplitude to 3.3Vpp and DC offset to 1.65V. However, should make choice of Rf and Rg, from the equation (Gain = 1 + Rf/Rg), given that gain equals to 3, therefore, Rf should be two times of Rg, lastly, I choose Rf for 20kOhm, and Rg for 10kOhm, and constructed circuits for transmitter on the break board.

For the receiver amplifier, it is slightly different, it required output form receiving amplifier need to be within 0.1-3.3V, and if obtained a 330mVpp signal, and chooses amplification rate to be 10, then output should be 3.3Vpp amplitude, and circuit was exactly the same as the two have built. Lastly, connect both transducers and amplifiers, and connect signal generator and generate a 3.3 App square wave, and measure the output from receiver amplifier. From observation, it is clearly that when distance is closer, the amplitude of output voltage will beyond the 3.3Vpp, therefore I detected that minimum distance is around 15 centimeters.



Finally, there is no noise found from the power supply.

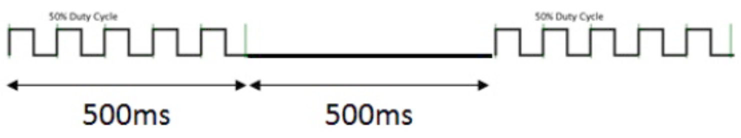
Lab 3:

In this lab, I tested the Analogue to digital converter(ADC) on the CC1350 micro-controller board. By using code composer studio, import projects from blackboard.

Basically, analogue-to-digital converter is used for converting an analogue voltage into a digital number at a specified sampling rate. However, ADC sampled a voltage, the number will be stored for the CPU core to access.

For example 1, it is a guidance about how to reading data from the ADC and displaying this data on the Code Composer Studio. Second step, is to follow the instructions to set up and initialize the values. At last, obtained a graph to display the data size, and “Buffer Ready” printed on the Putty.

For example 2, it is a slightly different from first example, it is continuously reading data from ADC and displaying it. Two buffers was used in this example, one for receiving ADC data, and an-



other one read and process data. For modifying the code, it was “max = 0”, it should be modified to:

Code presentation:

```
// max = 0 :—>
max = microVoltBuffer[i] > max )?microVoltBuffer[i]:max;
```

Lab 4:

In this lab, I tested the PWM generator on the CC1350 micro-controller board, and import project into code composer studio.

For the first task, when we generating the PWM signal, the reason why by using the PWM_setDuty and Task_sleep functions, because there is a gentle silence in second period on the figure shown below, in this period, receiver will required to collect the data of reflected ultrasonic wave from the obstacle, it is delayed by the round trip time of the signal.

By drawing the block diagram for connection between components, connect CC1350 to the input pin(DIO 24) of transmitter amplifier, output pin(DIO 24)of CC1350 connect to receiver amplifier, and measure the output signal, and note down that output from receiver should not exceed 3.3 V (Because of the distance is too close).

Blocking diagram shown above

Lab 5:

In this lab, it is to detect the ultrasonic signal with the micro-controller's ADC and calculate the distance on CCS, and import range_detect project into PC.

By generating a pulsed ultrasonic signal, it is required to be able to detect the rising edge of the received signal. We will use a digital filter to detect the signal, and auto-correlate itself.

For implementation part, the task is to construct a function named is_it_here(), because there is real and imaginary parts if the correlation, it takes in a buffer and a position.

Code presentation:

```
Bool is_it_here(volatile uint32_t * buffer, int i)
{
        Complex32 num = autocorrelate(buffer, i);
        int magnitude = (num.real * num.real)  + (num.imag + num.imag);

        if (magnitude >= SENSITIVITY){
                return TRUE;
}
        else{
                return FALSE;
}
}
```

Note : started on line 145 of 'adcSingleChannel.c', there is always command line appeared in the Putty.

From now, it is accessible to the position of signal, therefore we need to search through buffer. And to write a for loop to access each position. Lastly, delete the test code and implement a loop.

Code presentation:

```
// Loop through the buffer to find the position
// TODO: complete

For (i = 0; i< ADCBUFFERSIZE - FILTER_COEFFS; i++){
        if (is_it_here(microVoltBuffer, i))
                {
                        break;
                }
}
```

Finally, I ran my application to measure the distance to an object, it accurately measure the distance between the obstacle and transducers.