Ultrasonic Rangefinder: Design, Testing, and Implementation

1st Jason Mao
Electrical Engineering
California Polytechnic State University
San Luis Obispo, United States
jmao03@calpoly.edu

2nd Jaden Tran
Electrical Engineering
California Polytechnic State University
San Luis Obispo, United States
jtran249@calpoly.edu

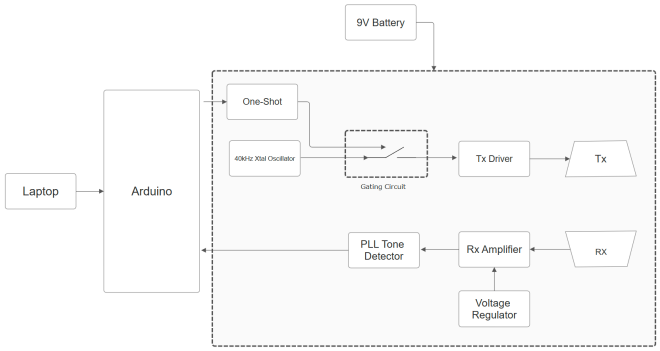


Fig. 1. Overall Topology of the Ultrasonic Rangefinder

***Abstract*—This paper introduces the key components of an ultrasonic rangefinder and describes the entire design and process from start to finish. The ultrasonic rangefinder uses time-of-flight measurements of transmitted echoes from nearby objects to determine the range. Concepts learned in lecture are proven through calculations and system tuning. By combining theoretical and practical knowledge, an efficient ultrasonic rangefinder can be built from scratch using individual components.**

***Index Terms*—Ultrasonic Sensors, Range Finding, Distance Measurement, Path Loss, Time-of-Flight, Sensor Calibration**

I. Introduction

In EE 449, Electronic Design Laboratory, the modules shown in Fig. 1 are constructed and analyzed weekly throughout the quarter. This paper will focus on covering all modules in detail and explaining the design process for each individual module. The goal of this lab is to create a functioning ultrasonic rangefinder utilizing a transducer pair with a center frequency of 40 kHz and other various components that fulfill necessary functions for complete operation. An Arduino microcontroller (Nano/Uno) will be used to trigger a 40 kHz burst from the transmitter. This burst will hit an object (cardboard wall/room wall) and reflect back into the receiver, where it will be detected and used to find the time of flight (ToF). ToF refers to the time it takes for an ultrasonic pulse to travel from the transmitter to the object and reflect back to the receiver. In this case, it is represented by the duration between when the trigger pulse activates and when the reflected signal is detected. With this, the distance between the transducer pair and the object can be found. The equipment required to do this includes a function generator, an oscilloscope, a power supply, and a multimeter. The necessary materials include the EE449 laboratory kit and multiple breadboards.

II. Ultrasonic Sensors

A. Ultrasonic Sensors: Theory

The ultrasonic sensor pair used in this lab is the 40T/R-12 by Jameco. This transducer pair (transmitter+receiver) has a center frequency of 40 kHz and requires a direct line of sight with the object in order to consistently transmit and receive signals. With a beam angle of 60°, it is important to keep them aligned with each other. The way the transducer pair works is that it converts electrical energy into ultrasound energy to transmit and then back into electrical energy to receive. When a 40kHz signal is applied to the transmitter, its piezoelectric crystal oscillates at 40kHz, generating an ultrasound wave that propagates through the air at the speed of sound. If the wave encounters an object, it reflects back as an echo. The receiver detects this echo, causing its piezoelectric crystal to vibrate and convert the ultrasound energy back into an electrical signal [1].

B. Ultrasonic Sensors: Design, Testing, Results

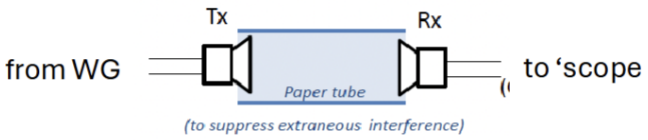


Fig. 2. Ultrasonic Sensor Testing/Verification Circuit

For the first module of the design, only the transducer pair will be tested. Tests will be performed to verify its center frequency and bandpass behavior as given by the datasheet. The test circuit used is shown in Fig. 2 and involves a function generator along with an oscilloscope. The transmitter (Tx) is taped on one end of a paper tube and the receiver (Rx) is taped on the other end. Doing this helps suppress noise from any outside interference. Using the function generator, a

square wave is applied to the Tx at varying frequencies close to 40 kHz. For each frequency, the gain is determined and then converted to decibels (optional). From this, the bandpass response of the Tx-Rx pair can be derived and modeled, as shown in Fig. 3. In this case, the output voltage was graphed against the frequency, which was converted from Hz to krad/s. Equation (1) shows how the gain is converted into decibels.

$$dB = 20 \log \left(\frac{V_{\text{out}}}{V_{\text{in}}} \right) \tag{1}$$

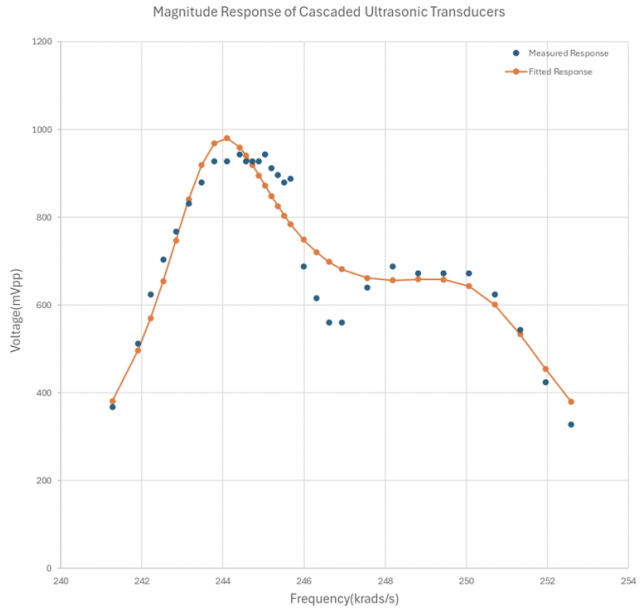


Fig. 3. Magnitude Response of the Tx-Rx Pair

Once this is done, the step response of the Tx-Rx pair is captured. This response defines the shortest-in-duration signal that can be used properly for range-finding applications. In Fig. 4, the measured step response oscillation frequency came out to be 37.23 kHz, and the shortest duration that can be sent came out to be 1.741 ms (allows for steady-state). This response was measured with an input signal of 10 Vpp at 50 Hz.

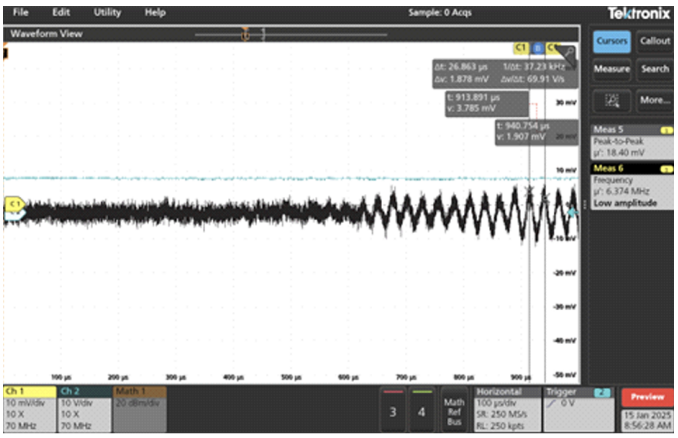


Fig. 4. Step Response of the Tx-Rx Pair

The Solver Excel app is used to derive the four parameters (ω_{o1}, Q_1, ω_{o2}, Q_2) from the magnitude response graph. Since the Tx and Rx exhibit a second-order high-Q bandpass behavior, the overall response of the system will be a fourth-order response. ω_{o1} and ω_{o2} represent the natural/resonant frequencies of the system, while Q_1 and Q_2 represent the quality factors associated with their respective resonances.

Ho	2.79553
wo1	243.71655krad/s
wo2	250.603282krad/s
Q1	80.33995613
Q2	49.00816234
fo1	38.79kHz
fo2	39.88kHz

Fig. 5. Fitted Data for Magnitude Response of the Tx-Rx Pair

III. Transmitter Driver

A. Transmitter Driver: Theory

Instead of the usual voltage amplification, the goal of the transmitter driver will be to produce a square wave with an amplified current. This will allow the transmitter to output a strong and stable signal that can properly echo off of another object. The chosen IC for this purpose is the TC4428 - 1.5A dual high-speed power MOSFET driver. This IC was chosen for its excellent driving capabilities that ensure clean, high-speed transitions on the transmitter side of the system [2]. Because the driver is of MOSFET type, it will generate a square wave at its output from its CMOS configuration.

B. Transmitter Driver: Design

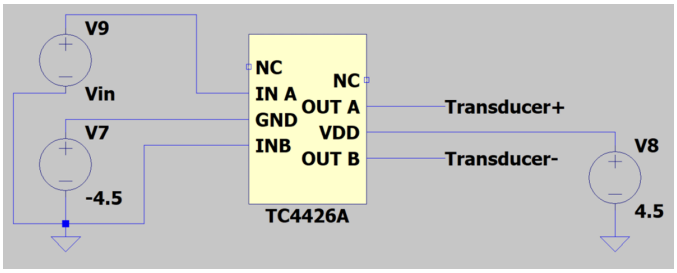


Fig. 6. Transmitter Amplifier Circuit

The transmitter driver will take a 40 kHz, 9 Vpp sine wave and will produce a current amplified square-wave version of it. This square wave will drive the transmitter and allow it to output a strong enough signal to get an echo for the receiver to detect. Additionally, when the transmitter does not receive a signal, it will output a constant high voltage to prevent unwanted oscillations and effects due to noise. The supply voltages for the initial design are -4.5V and 4.5V, but will be further modified when the 9V battery is implemented.

IV. Receiver Amplifier

A. Receiver Amplifier: Theory

The ultrasonic transducer used as a receiver produces a very small voltage output, and gets smaller as the distance increases. To be able to accurately measure the signal received by the transducer, an amplifier will be used to amplify the signal. The LMC662 was chosen over the other available LM358, since the gain-bandwidth product of the LMC662 is 1.4 MHz compared to the LM358's 1 MHz GBW [3]. A higher GBW provides more room for gain and frequency performance, which can be helpful later in the design. The receiver amplifier is connected in typical non-inverting configuration, resulting in a gain that can be calculated using Equation (2).

$$A = 1 + \left(\frac{R_2}{R_1}\right) \tag{2}$$

B. Receiver Amplifier: Design

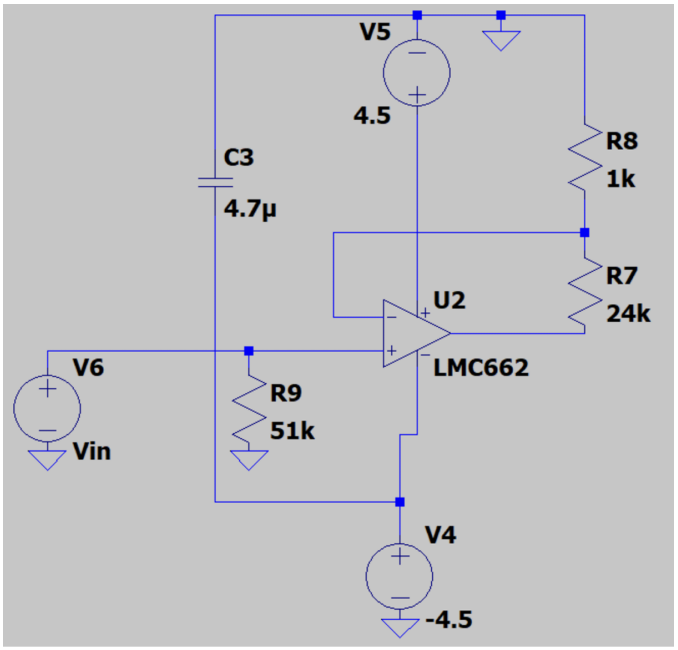


Fig. 7. Receiver Amplifier Circuit

C. Receiver Amplifier: Testing and Results

Figure 7 shows the circuit for the initial receiver amplifier. Figure 8 shows the results of the amplifier and whether or not it can reliably detect a signal at varying distances.

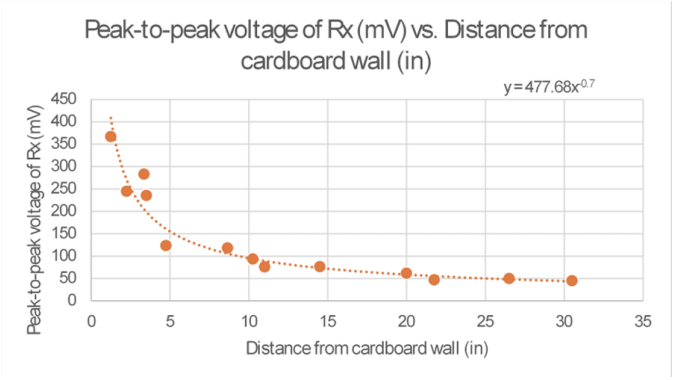


Fig. 8. Path-loss vs. Distance Relation Graph

V. High Gain Receiver Amplifier

A. High Gain Receiver Amplifier: Design

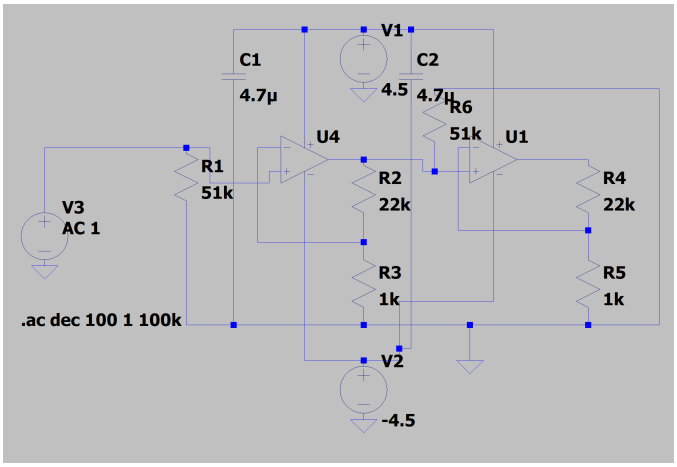


Fig. 9. High-Gain Receiver Amplifier Circuit

To achieve the minimum V_{RMS} for the phase-locked loop to lock on at the receiver, a higher gain circuit than presented in the last section is needed. Rather than increasing the gain on the single op-amp circuit, another op-amp was cascaded, allowing for more gain with a better frequency response. The gain of the two op amps are equal, and the cascaded gain is given by Equation (4).

$$A_1 = A_2 = 1 + \left(\frac{23k}{1k}\right) = 23\frac{V}{V} \tag{3}$$

$$A = A_1 * A_2 = 529\frac{V}{V} \tag{4}$$

Simulating the circuit in LTSpice yields a gain of 51.3 dB at 40 kHz, ensuring minimum V_{RMS} for the PLL to lock on at 3.5 m.

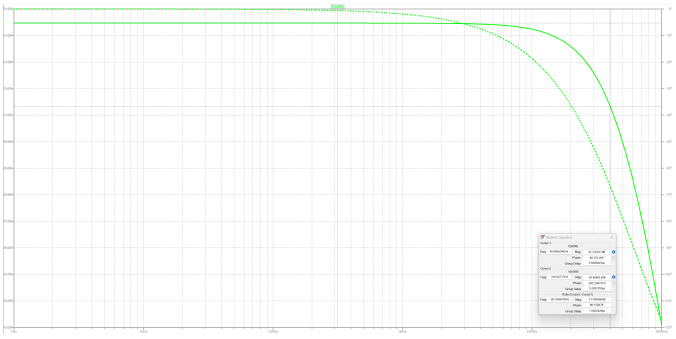


Fig. 10. Cascaded Amplifier Frequency Response

B. High Gain Receiver Amplifier: Testing and Results

- New distance results from cascaded amplifiers

VI. One-Shot and Gating Circuit

A. One-Shot and Gating Circuit: Theory

Our data from measuring the step response of the ultrasonic transducers showed that the time to steady-state of the transducer's response is 1.741 ms. We configured a 555 Timer as a one-shot triggered by the falling edge of an input pulse [4]. The output of the one-shot will allow a 40 kHz signal into the transmitter for a set period of time. The minimum pulse duration of the one-shot required is 1 ms, so the one shot was configured to output a pulse at that duration.

$$T_{\text{pulse}} = 1.1RC \tag{5}$$

$$T_{\text{trig}} < T_{\text{pulse}} \tag{6}$$

$$R_{\text{trig}}C_{\text{trig}} = \frac{1}{6}RC \tag{7}$$

Choosing R = 10 kΩ, we get C = 94nF, and we choose R_{trig} = 51 kΩ and get C_{trig} = 3.6 nF.

The 40 kHz signal being sent into the transmitter driver is controlled by the one-shot output into the gating circuit. For the gating circuit, we use the LM393 comparator with the one-shot as the reference into the non-inverting terminal and a $1_{\frac{}{2}}V_s$ input to the inverting terminal. This allows the output to go high when the one-shot output is high [5]. At the output, a pull-down resistor is connected to the 40 kHz wave. This makes it so that when the output is high, the pull-down resistor causes the high output to oscillate at 40 kHz signal, achieving the gate functionality where a 40 kHz signal is sent to the transmitter only when a pulse triggers the one-shot, allow for control of the transmitter via a microcontroller.

B. One-Shot and Gating Circuit: Design, Testing, Results

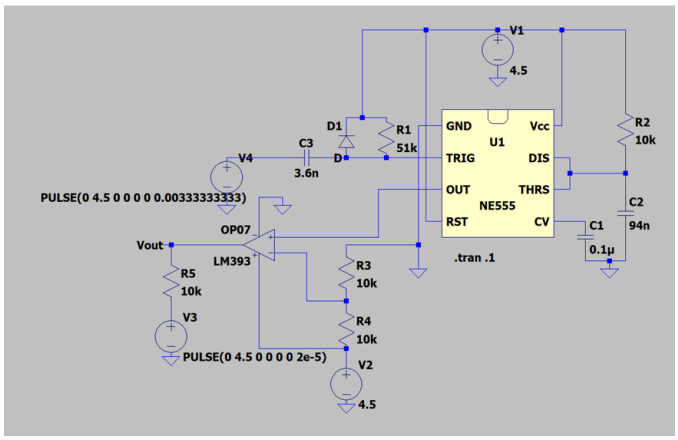


Fig. 11. One-Shot/Gating Circuits

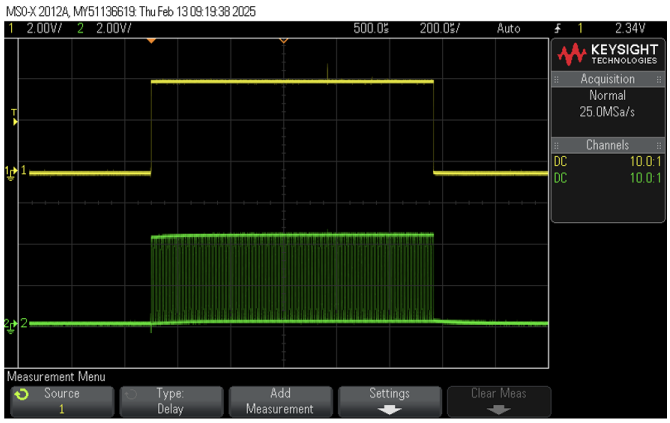


Fig. 12. 40 kHz Burst (Green) with One-Shot Pulse (Yellow)

VII. PLL-Based Tone Decoder

A. PLL-Based Tone Decoder: Theory

To allow for more precise measurement of the time between transmitted pulse and received, we used a phase-lock loop. The PLL locks on to the 40 kHz signal received and amplified, switching its output from high to low. The time between the microcontroller sending the pulse to the one shot to the PLL locking onto a signal can be used to determine the distance between the transmitter and the object it is detecting. The PLL used for this is the LM567, typically used as a tone decoder in touch-tone telephones [6]. To set the PLL to lock onto a 40 kHz signal, we set the free-running frequency of the voltage-controlled oscillator in the LM567 at 40 kHz using Equation (8).

$$f_o = 1.1R_1C_1 \tag{8}$$

Choosing v, we get C_1 = 2.8 nF, using 2.7 nF as that is what was available. To set capture range of the PLL, we choose a bandwidth of 8% of the free running frequency. The LM567

datasheet gives an table corresponding to bandwidths. For a bandwidth of 8%, we get Equation (9), yielding C_2 = 0.1 uF. The datasheet specifies C_3 = 2*C_2, yielding 0.2 uF.

$$f_o C_2 = 4.1kHz * uF \tag{9}$$

B. PLL-Based Tone Decoder: Design and Testing

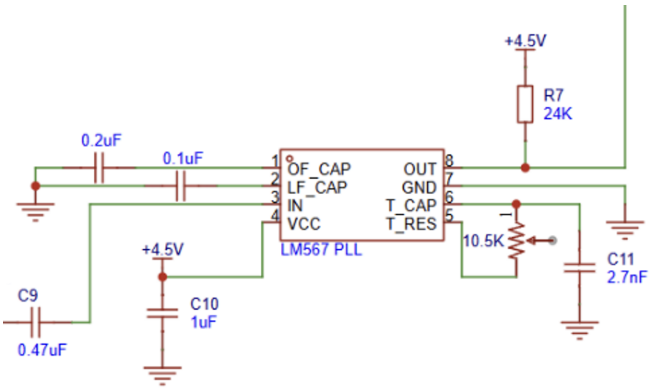


Fig. 13. PLL-Based Tone Decoder Circuit

Figure 13 shows the schematic for the PLL tone decoder circuit to detect our 40 kHz received signal. A potentiometer is used in place of R_1 to adjust the free-running frequency of the VCO to 40 kHz. Adjusting the free-runnign frequency yields R_1 = 10.5 kΩ. A 0.47 uF coupling capacitor at the input connects to the output of the high-gain RX amplifier. The LM567 has weak pull-up, so a 24 kΩ resistor is used to pull the output up to V_s.

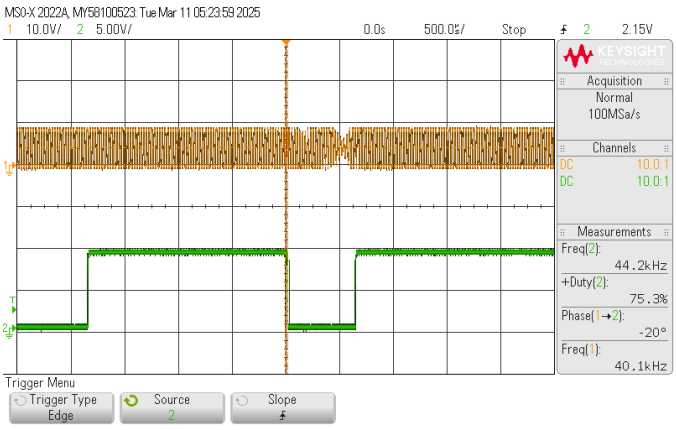


Fig. 14. PLL Circuit Output

Figure 14 shows the high-to-low and low-to-high transition of the PLL output when it locks onto a 40 kHz frequency then loses the lock.

VIII. Pierce Crystal Oscillator

A. Pierce Crystal Oscillator: Theory

The Pierce crystal oscillator will act as the 40 kHz signal source for the one-shot circuit to pulse and send out the waveform needed to drive the 40 kHz transmitter. The components needed to construct it include an ECS-400-12.5-13X - 40 kHz crystal, a CD4069 - hex logic inverter, and several passive components. The circuit itself operates as a positive feedback loop, and the Barkhausen criteria must be satisfied in order for the circuit to function as intended. To pass this criteria, the circuit frequency must allow the output of the circuit to have a gain of one or more and zero phase shift compared to the input [10]. Ideally, this frequency is at the resonant frequency of the crystal, but in this case, it is 39,999 Hz.

B. Pierce Crystal Oscillator: Design, Testing, Results

The design for the oscillator uses an ECS-400 - 40 kHz crystal, a CD4069 - hex logic inverter, and several resistors/-capacitors to create a stable oscillator at the crystal's resonant frequency of 40 kHz [7]. The inverter is biased by the feedback resistor R5, which sets the necessary gain and the CMOS operation to linear, instead of saturating the output to HIGH or LOW. R6 provides additional feedback to sustain oscillations and helps compensate for any losses in the crystal circuit, including parasitic losses and external interference. The two capacitors sandwiching the crystal are needed to form the phase-shift network that ensures the oscillation of the crystal stays at 40 kHz. These components allow the first buffer to act as an amplifier with a gain greater than 1 for the 40 kHz signal. The second buffer is just a buffer that is used to connect the oscillator to the gating circuit. Adding this prevents the gating circuit from loading down the oscillator, which can lead to the deterioration of the oscillator's stability and frequency [8]. The output of the second buffer will then be pulsed with the one-shot circuit and passed to the Tx driver amplifier for transmission.

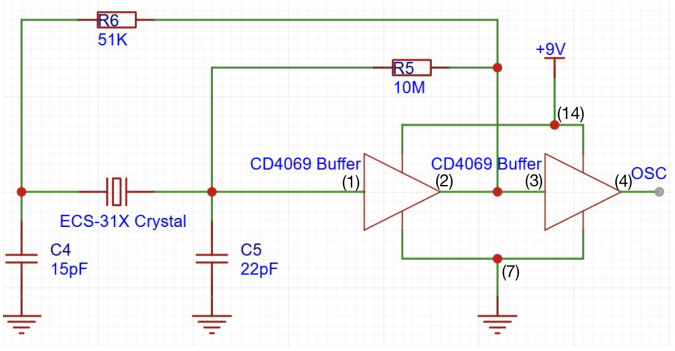


Fig. 15. Pierce Crystal Oscillator Circuit

To test the Pierce crystal oscillator for the open-loop part, a 100 mVpp sine-wave is inputted to the oscillator circuit from the end of R6 connected to the output of the first buffer. A pair of oscilloscope probes will then be connected to the output of the first buffer. The second buffer is taken out from the

circuit. The operating frequency of the input is then varied by 1 Hz in the vicinity of 40 kHz, as the crystal is incredibly sensitive. The best result was from a set input frequency of 39,999 Hz. This gave a gain of $\frac{730mV}{206mV}$ = 3.544, meeting the required gain of one or more, and a phase offset of just 1 degree. This satisfied the Barkhausen criteria for open-loop testing, and meant that the circuit built could move onto the closed-loop test.

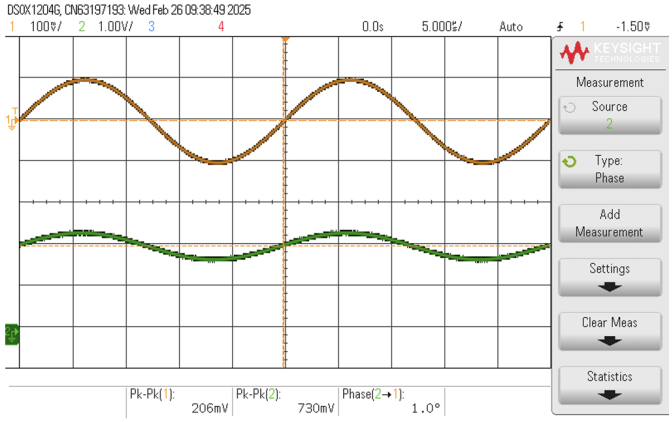


Fig. 16. Open-Loop Testing of Oscillator Circuit

For the closed-loop testing, a small DC voltage is inputted to the oscillator circuit. The expected output is a 40 kHz square waveform that will replace one connection from the function generator in our system; the microcontroller will replace the other connection. This 40 kHz square waveform will be pulsed by the one-shot circuit, which will force the gating circuit to let it through to the Tx driver amplifier. This will result in a transmitted 40 kHz burst waveform whenever the gating pulse coming from the one-shot circuit is high.

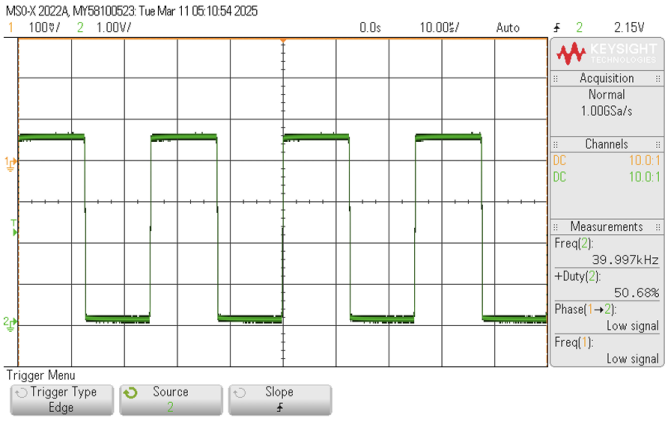


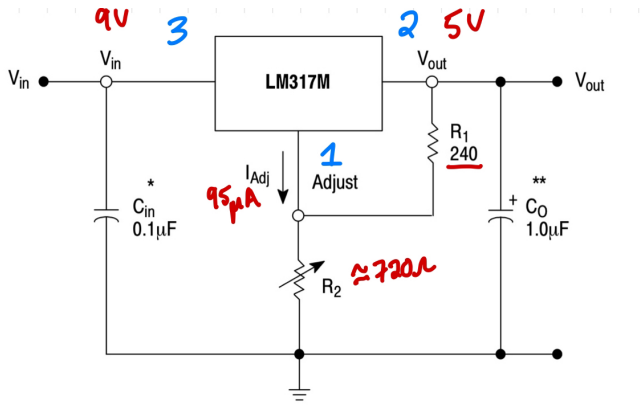
Fig. 17. Closed-Loop Testing of Oscillator Circuit

IX. (9V Battery Operation)

A. 9V Battery Operation: Theory

With the system on a 9V battery operation, it will become fully portable and will not require an expensive power supply. A voltage follower can be used to buffer the output voltage of the battery and ensure that it is delivering a stable 9V. DC offsets in the system will need to be resolved, as they can lead to unnecessary signal saturation and improper logic operation throughout the system. These offsets can be fixed by employing the use of a LM317M - voltage regulator.

B. 9V Battery Operation: Design, Testing, Results



$$V_{out} = 1.25\ V\left(1 + \frac{R_2}{R_1}\right) + I_{Adj}\,R_2$$

Since I_{Adj} is controlled to less than 100 μA, the error associated with this term is negligible in most applications.

Fig. 18. Datasheet Equation for LM317M Linear Regulator

To set up the LM317M, the datasheet equation must be used to calculate the necessary resistor values to create the circuit [9]. Once this is set up, it can be easily tested by inputting 9V at pin 3 (V_{in}) and using a multimeter to measure the voltage at pin 2 (V_{out}). The output voltage will act as a 5V power supply for any components that need to be DC offsetted in the system.

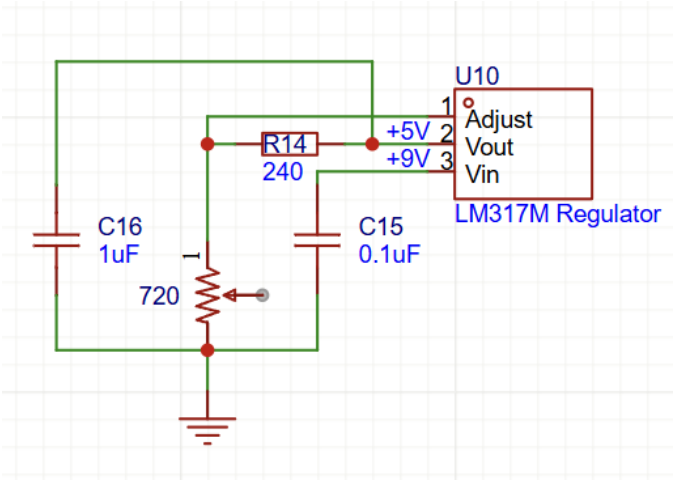


Fig. 19. Linear Regulator Circuit for 4.5V offset

X. Microcontroller

A. Microcontroller: Code Design, Testing, Results

The microcontroller selected is the Arduino Uno, due to its easy set-up and user-friendly IDE. The Uno will provide the trigger input for the one-shot circuit and will also process the output from the PLL as an echo.

The Arduino code should meet the following requirements:

- Trigger the one-shot circuit periodically
- Measure the time elapsed from the trigger event and the lock event while displaying the result on the serial monitor
- Perform several measurements and calculate their average and standard deviation
- Adjust for any zero range delay offset
- Display distance estimated from time-of-flight measurements

```
// Pin assignments
const int triggerPin = 9; // connect to
    trigger (pin 2) of 555-timer
const int echoPin = 6; // connect to output
    (pin 8) of LM567-PLL

// Constants
const float speedOfSound = 343.0; // in m/s
const float zeroOffset = 0.1; // Zero-range
    delay offset adjustment (ADJUST)
const int numSamples = 35; // Number of
    measurements per cycle
const int triggerPulse = 1000; // Trigger
    pulse width in microseconds (1 ms min)

void setup() {
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);

  // Start a 300 Hz square wave on pin 9 (50%
      duty cycle)
  tone(triggerPin, 300);
}

void loop() {
  float distances[numSamples];
  float sum = 0;

  // Collect multiple distance measurements
  for (int i = 0; i < numSamples; i++) {

    // Trigger the ultrasonic pulse and send
        to one-shot
//   digitalWrite(triggerPin, LOW);
//   delayMicroseconds(2);
//   digitalWrite(triggerPin, HIGH);
//   delayMicroseconds(triggerPulse);
//   digitalWrite(triggerPin, LOW);

    // Measure pulse duration
    float duration = pulseIn(echoPin, LOW,
        29150); // Timeout at ~5m range
    float distance = (duration * speedOfSound
        / 2000000) - zeroOffset;
    distances[i] = (distance > 0) ? distance :
        0; // Ignore negative values
    sum += distances[i];
    delay(10); // Small delay between samples
  }

  // Compute average and standard deviation
  float avg = sum / numSamples;
  float variance = 0;
  for (int i = 0; i < numSamples; i++) {
    variance += pow(distances[i] - avg, 2);
  }
  float stdDev = sqrt(variance / numSamples);


  // Display results
  Serial.print("Avg Distance: ");
  Serial.print(avg * 100); // convert m to cm
  Serial.print(" cm, Std Dev: ");
  Serial.print(stdDev * 100); //convert m to cm
  Serial.println(" cm");

  delay(250); // Wait before next measurement
      cycle
}
```

The pulseIn function will measure the time elasped from right when the PLL goes high to low (40 kHz signal detected and locked on) until it goes back to high again. Delays are used to ignore any noise coming from the immediate transmission of the 40 kHz pulse waveform. This will make the resulting output become less noisy. The first testing of the code proved to have very messy results, as the serial monitor was outputting a distance whenever something was in direct contact with the transducer pair, but it was not accurate. To fix this issue, the number of measurements per cycle taken was increased to 35 from 10. This made the results have a smaller margin of error, but there were still obvious issues that could be fixed.

```
16:07:20.796 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:21.169 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:21.587 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:21.960 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:22.379 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:22.755 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:23.173 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:23.543 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:23.962 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
```

Fig. 20. Serial Monitor with nothing in front of Tx/Rx Pair

When nothing is in front of the transducer pair, the serial monitor displays nothing, as there is no sufficient data to use in the code's calculations. This is good, as it shows the PLL is not locking onto the ceiling.

```
16:07:25.542 -> Avg Distance: 5.79 cm, Std Dev: 17.36 cm
16:07:26.005 -> Avg Distance: 65.82 cm, Std Dev: 131.99 cm
16:07:26.378 -> Avg Distance: 10.40 cm, Std Dev: 22.51 cm
16:07:26.841 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:27.260 -> Avg Distance: 27.47 cm, Std Dev: 82.42 cm
16:07:27.724 -> Avg Distance: 5.26 cm, Std Dev: 15.79 cm
16:07:28.141 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:28.517 -> Avg Distance: 0.00 cm, Std Dev: 0.00 cm
16:07:28.938 -> Avg Distance: 5.05 cm, Std Dev: 15.14 cm
16:07:29.311 -> Avg Distance: 2.70 cm, Std Dev: 4.84 cm
```

Fig. 21. Serial Monitor while waving hand in front of Tx/Rx Pair

Waving a hand in front of the transducer pair gives distance values in the serial monitor. Some of these values made sense, like the 5.79 cm and the 5.05 cm, as the hand was close on top of the transducers. Other values were way too far off, and it is suspected that they come from the PLL not settling to a proper lock-on point. This essentially makes it a glorified motion sensor.

XI. Future Improvements

A. Possible Improvements to the System

Our current code allows the ultrasonic sensor to successfully detect objects and somewhat track their distances. Further improvements can be made to the microcontroller code, especially in the data filtering and averaging area. By fine-tuning the code to better collect reliable data, measurements taken will make more sense and the Arduino will work off of less noise. The receiver and PLL sections of the system can also be improved, as the PLL was able to detect the echo and lock onto it, but noise and external interferences caused it to give the Arduino garbage data. It is suspected that the receiver is coupled to something else on the circuit, but due to time constraints, this was not further investigated. The circuit could be made on a larger breadboard to allow for more room for components. Doing this can help resolve the coupling issues seen along with any crosstalk between the transmitter and receiver.

B. Final Design Schematic

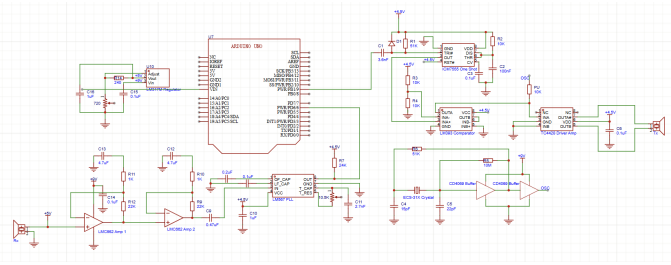


Fig. 22. Overall System Schematic under 9V battery operation

References

[1] DRex, "A Complete Guide to Ultrasonic Sensors: How They Work, Their Types, and Applications - DRex Electronics," DRex Electronics, Jan. 21, 2025. https://www.icdrex.com/a-complete-guide-to-ultrasonic-sensors-how-they-work-their-types-and-applications/ (accessed Mar. 03, 2025).

[2] Microchip, "1.5A Dual High-Speed Power MOSFET Drivers", TC4428A Datsheet, Mar. 2014.

[3] Texas Instruments, "LMC66x CMOS Dual Operational Amplifiers", LMC660, LMC662 Datasheet. Mar. 1998 [Revised Feb. 2024].

[4] Renesas, "ICM7555, ICM7556 General Purpose Timers", ICM7555 Datasheet, Mar. 2020.

[5] Texas Instruments, "LM393B, LM2903B, LM193, LM293, LM393 and LM2903 Dual Comparators," Oct. 1979. [Revised Jan. 2025]

[6] Texas Instruments, "LM567x Tone Decoder", LM567CN Datasheet, May 1999 [Revised Jan. 2022].

[7] ECS International, "ECS-31X Tuning Fork Crystal", ECS-31X Series Datasheet, Rev 2017.

[8] Texas Instruments, "CD4069UB CMOS Hex Inverter", CD4069 Datasheet, Nov. 1998 [Revised Jan. 2019].

[9] ONSemi, "Voltage Regulator – Adjustable Output, Positive", LM317MTG Datasheet, Dec. 2024.

[10] P. Nayeri, "Lecture 14: Sinusoidal LC and crystal oscillators with inverting amps, Pierce crystal oscillator," 2025.