COLUMBIA UNIVERSITY

FINAL PROJECT REPORT
E3390 ELECTRONIC CIRUITS DESIGN LAB

Ultrasonic Positioning System

Sabir Shrestha
Samuel Kusi
Deep Shrestha

Submitted in partial fulfillment of the requirements for the Bachelor of Science Degree
5/10/2010

Table of Contents

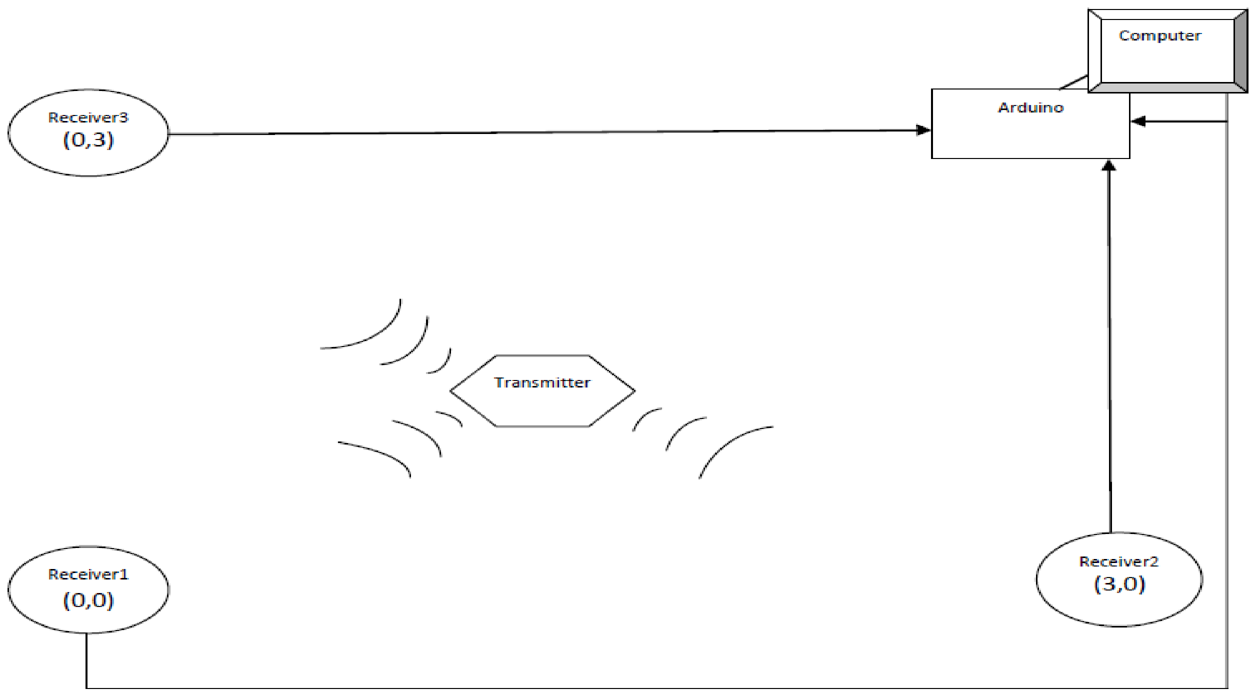
Abstract

This Ultrasonic Positioning System technology is very similar to the GPS, except that this is on a small scale and the role of transmitters and receivers have been reversed. In GPS system, multiple satellites transmit real time signal to a single receiver and the position of the receiver (object) is calculated based on time delays and the speed of light. Whereas in this system we have a single transmitter sending pulses to three receivers and position of the transmitter (object) is calculated based on the speed of sound.

Introduction

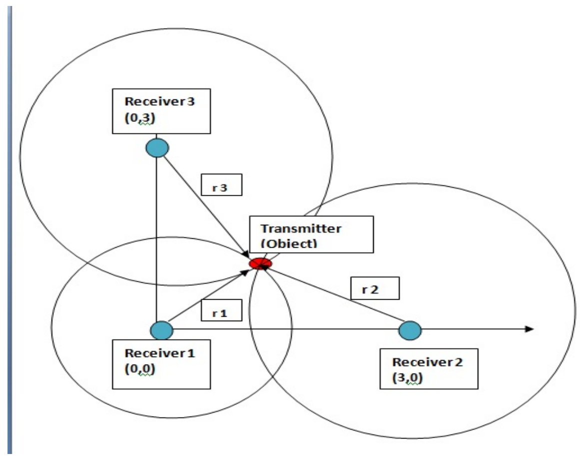
The project focuses on building a system capable of tracking position of an object through the use of ultrasonic waves. While GPS is an excellent tool for such application outdoors, it is very inefficient in an indoor setting. The use of ultrasonic waves is a better alternative in a limited space environment. For this project we are building a system that will utilize three ultrasonic receivers and trilateration algorithm implemented through Arduino to track the position of an object transmitting ultrasonic waves. The receivers will be placed strategically in three locations of a room. Each receiver will receive ultrasonic waves from the transmitter in different time period depending on the distance of the transmitter from each receiver. When the receivers receive a signal, it will be sent to an arduino, where the trilateration algorithm will be implemented to calculate its position with respect to one of the receivers (origin). The result will be then displayed on the screen as x and y co-ordinates.

Block Diagram



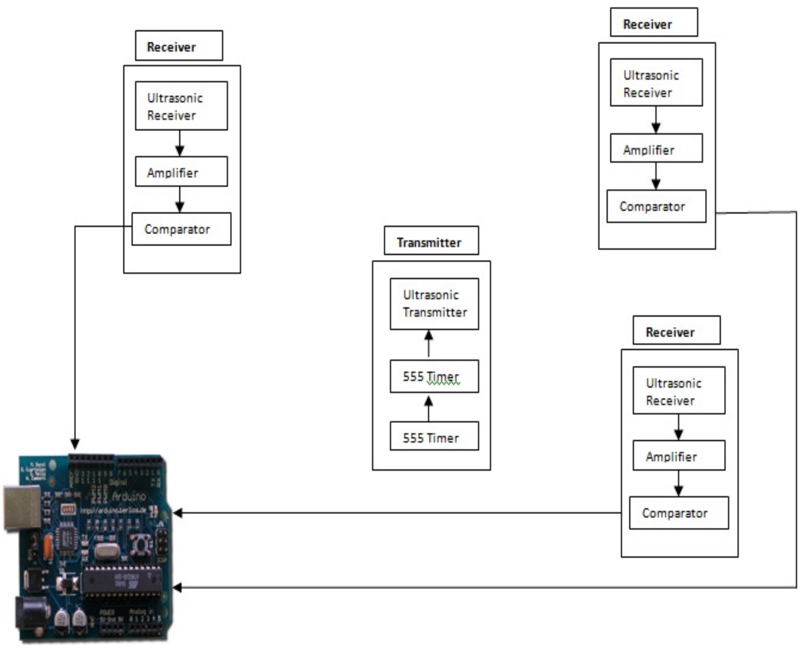
Theory

Trilateration: This is a mathematical technique that is employed by Global Positioning System (GPS) to determine the user position, speed and elevation. Trilateration uses the known locations of two or more reference points and the measured distance between the subject and each reference point.



Data from a single receiver narrows the position of the object down to a large area corresponding to the circles above. Adding data from a second receiver narrows position down to the region where two spheres overlap. Adding data from the third receiver, provides a relatively accurate position.

Structure of System

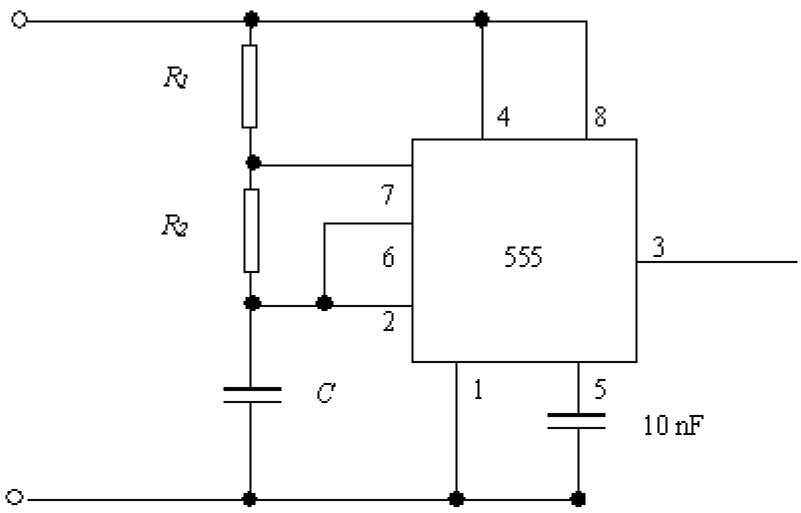


Transmitter

The transmitter circuit consists of an ultrasonic transmitter and two 555 timers. It has an angle coverage of 60 degrees and such six of these transmitters were connected in a circular pattern to yield angle coverage of 360 degrees. This was to increase the efficiency of the system to ensure all the receivers are able to receive signals when the transmitter is within our coordinate frame. The first 555 timer is used to trigger the supply voltage of the second 555 timer to pulse every 5 milliseconds. The second 555 timer then produces a cycle of 10 pulses at 40 kHz every 5 milliseconds. Below are illustrations of how the 555 timer works and the wave forms produced by each of the timers used.

555 Timer

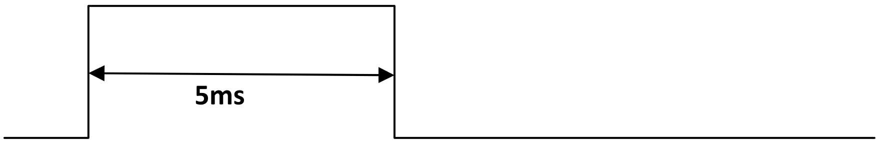
The 555 timer is configured as an astable multivibrator as shown in Figure 3. The multivibrator operates by the capacitor, C, charging through R1 + R2 and discharging through R2 when power is applied to the circuit. As the charge on the capacitor reaches about 2/3 of the supply voltage (10V) ,the upper comparator is triggered. The reset pin becomes active which resets the flip-flop that controls the state of the output pin. As a result, the output of the signal goes back to 0V. The capacitor will then begin to discharge through resistor R2. When the voltage across the capacitor reaches 1/3 of the supply voltage, the lower comparator is triggered. This will cause the control flip flop to become active. As a result, the output will go high. The capacitor will then begin to charge. This cycle of continuous charging and discharging results in a continuous stream of rectangular pulses which are illustrated in the waveforms below.



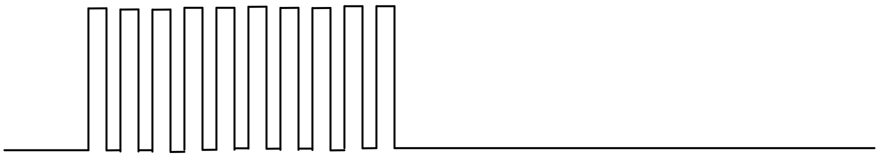
Pin Description

Pin1: Ground; Pin2: Trigger; Pin3: Output; Pin4: Reset; Pin5: Control Voltage;
Pin6: Threshold; Pin7: Discharge; Pin8: Positive Supply Voltage

Figure 3. Astable Multivibrator



Wave Form 1: Pulse from first 555 Timer



Wave Form 2: Pulse from second 555 Timer

The frequency and duty cycle of operation of the multivibrator depend on R1, R2 and C. The following formulae were used in selecting values for R1, R2 and C to obtain a desired frequency of 40 kHz:

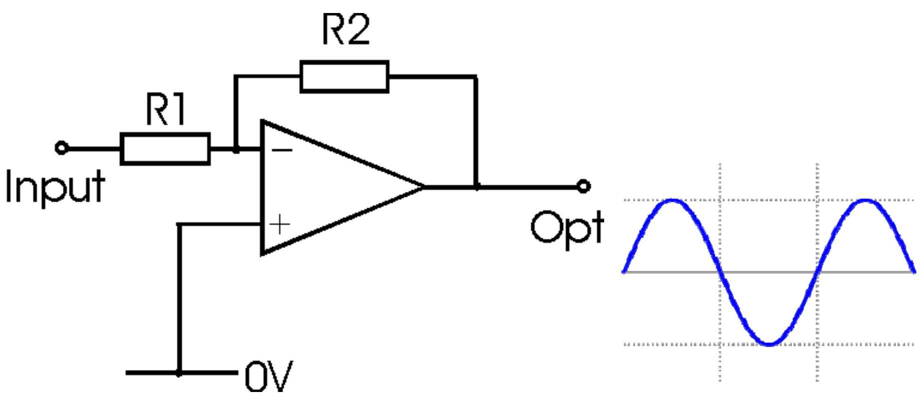
$f = \dfrac{1.44}{(R_1 + 2R_2)C}$; $D = \dfrac{R_2}{(R_2 + 2R_1)}$; $R_1 = 140$ Ohms; $R_2 = 140$ Ohms ; $C = 100nF$

Receiver

The receiver circuit comprises a 40 kHz ultrasonic receiver, an amplifier and a comparator. The ultrasonic receiver has a very good range for our application and such only one stage of amplification was needed. An amplifier of 1000x was used to amplify the received signals to counteract the attenuation of the signals as transmitter moves further from receiver within the coordinate frame. Output of the ultrasonic receiver was a sine wave and therefore needed to be converted to nice clean pulses. A comparator was used to convert the sine wave into square waves.

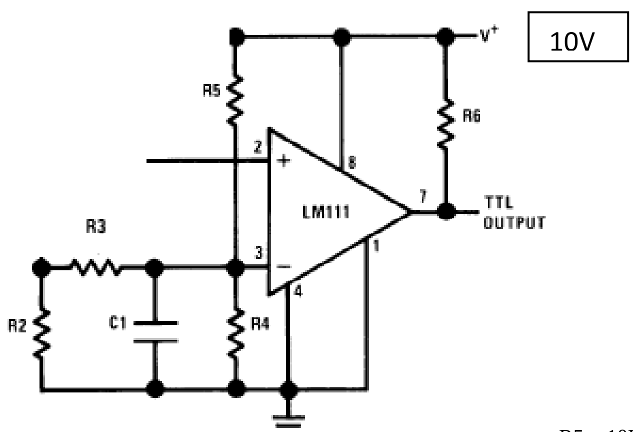
Amplifier

An inverting amplifier circuit with a gain of 100 was built. Signal from the receiver was applied as the input voltage of the amplifier. The amplifier represented by triangle amplifies the input voltage it receives and inverts its polarity producing an output voltage. The negative feedback configuration enables the output of the amplifier to keep the inputs near the same voltage so that saturation does not occur. The resistor R2 and R1 determines the gain of the op amp by the equation, $V = -(R2/R1)\,Vg$



Comparator

Since the output from the amplifier is a sine wave we need to convert it back to a square wave before it is fed in to the Arduino. The comparator accepts two analog inputs, compares them and produces an output either high or low. We used an opamp in constructing a comparator. The opamp acted like a switch that allows two options at the output. The threshold voltage of our comparator design was 5V. This means that if the input voltage is below is 5V the output will be high and if it is grater the output will be low.



R5 = 10K, R4 = 220 Ohm, R3 = 1K, R2 = 100K, R6 = 10K, C1 = 20nF

Arduino

The Arduino Mega was used to analyze the signals received by the receivers. Unlike Arduino Duemilanove which provides only two interrupt pins, Arduino Mega provides six interrupt pins. This suited our project better as we had to analyze three receiver signals to implement trilateration algorithm. The interrupt pins in the arduino allow detection of a signal under various conditions such as when a signal shows a rising edge or a falling edge or when a signal changes. For our purposes, we used the rising edge detection. Every time the arduino detects a rising edge, it calls a user defined function/method. In our case, we had the three interrupt pins of the arduino attached to three different functions. When either of these functions was called, we noted the time in microseconds. The time is noted is the time elapsed after the execution of the program. After noting all three times, we calculated the time difference and implemented the trilateration algorithm to calculate the relative position of the transmitter. This position was displayed on the screen in terms of x and y co-ordinates.

Bill of Materials

Part	Manufacturer	Quantity	Price ($)
Arduino Mega		1	59.95
Transmitter	Kobitone	6	29.34
Receiver	Kobitone	3	14.67
Resistor		10	5
Capacitor		4	2
555 timer	Philips Semiconductor	2	0.5
Op Amp	Philips Semiconductor	1	1
		Total	112.46

Health Safety and Environmental Issues

Ultrasound technology is widely used in many aspects of our industry today. The biomedical industry employs this technology the most. Focused high-energy ultrasound pulses can be used to break calculi such as kidney stones and gallstones into fragments small enough to be passed from the body without undue difficulty. Even though this technology is heavily used in the health industry, there are some risks that cannot be ignored. Occupational exposure to ultrasound in excess of 120 dB may lead to hearing loss. Exposure in excess of 155 dB may produce heating effects that are harmful to the human body, and it has been calculated that exposures above 180 dB may lead to death. The power emitted by our system in decibel is much less that 120 dB making our system free from any health any environmental risks.

Problems Encountered and How it was Resolved

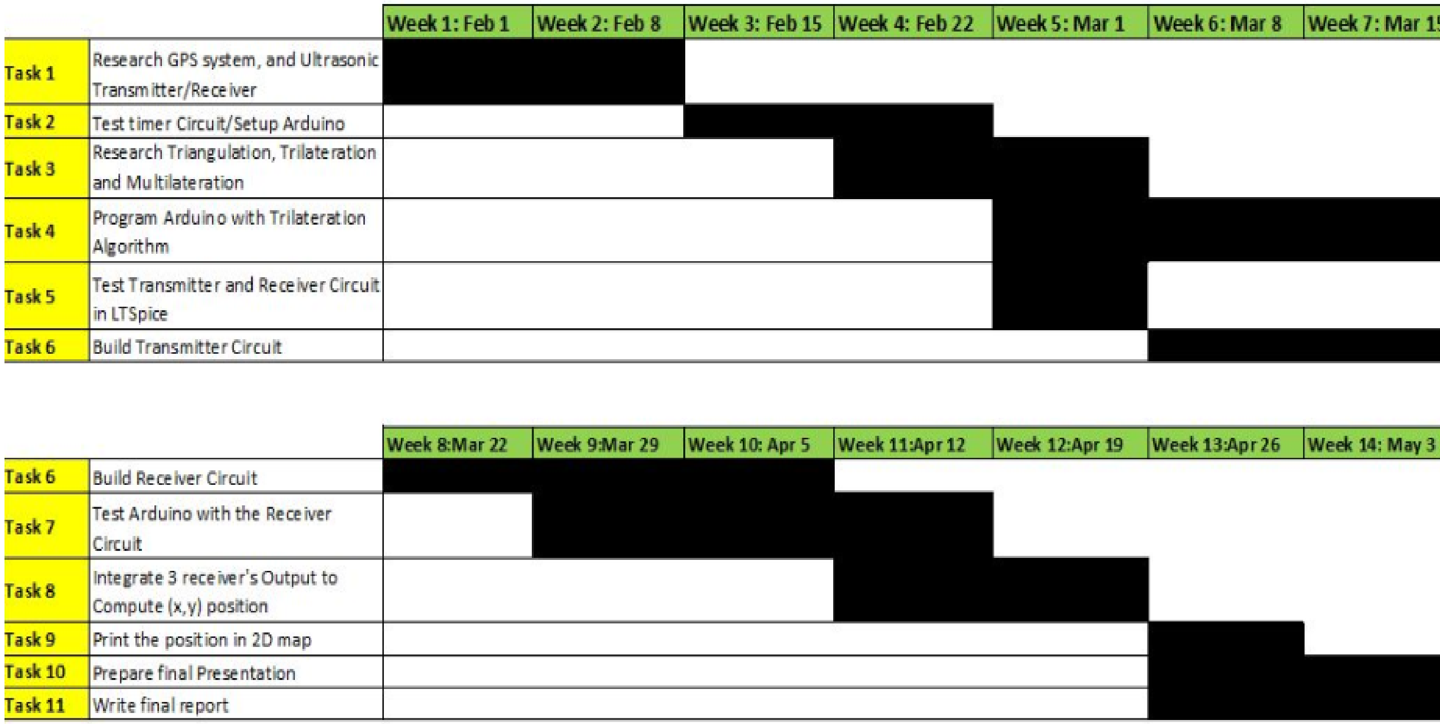
- Recording the time between transmission and receipt of the signal
- Pulsing the transmitter at the right frequency to detect the delay.

Solution: With a second 555 timer to trigger the other 555 timer being fed into the transmitter we were able to get a reasonable delay for arduino to accurately record time. With the ability of interrupt pins to detect rising edges of pulse and knowing the time between pulses are transmitted, the arduino code is modified for this operation.

- Noise interference

Solution: With the use of high pass filters we were able to get rid noise at frequencies lower than 40kz

Final Gantt Chart



Conclusion

The Ultrasonic Positioning system has numerous applications in the various industries. It can be used for machinery positive feedback control, robotics guidance and tracking, high security object guidance and tracking and also as an Indoor GPS. In addition this technology can be employed to controlling the movement of camera which is being used to record the activities of a moving object.

The project shows that the principle of positioning with ultrasonic sound waves is possible. Since the ultrasonic transmitters and receivers used have such a narrow angle in which they transmit and receive the system needs some help from the user with aiming. Preferably transmitters with wider angle should be used. Since the system was proved not to be linear at all distances (of reasons unknown) the distance measurements were good in some areas but got an error when the distance got too large. But when several measurements are done on the same place, coordinates calculated are very stable which shows that the interrupt functions works well and that they're not interfered with other functions.

Criticisms of This Course

The one semester time frame allocated to complete these projects has been a daunting task. From our experience, there was not enough time for testing. However, the one credit practice engineering course which was introduced last semester can serve as a good starting point for these projects. In the near future, this one credit course should have a lab component that will allow student to actually start building senior design projects that will lead into the next semester. I believe this head start will improve efficiency and creativity of projects as one has more time for ideas and testing.

Appendix (Arduino Code)

```
#define speedofsound 1125

double x,y;

int ultraSoundSignal = 7; // Ultrasound signal pin

unsigned long t1,t2,t3;

unsigned long counti;

double tim1, tim2, tim3;

int i;

void setup() {
  Serial.begin(9600);

 pinMode(ultraSoundSignal, OUTPUT);  // Sets the baud rate to 9600

 t1 = t2 = t3 = 0;
 i = 0;
}

void loop() {
  if(t1 == 0 && t2 ==0 && t3 == 0){
     digitalWrite(ultraSoundSignal, LOW); // Send low pulse
     delay(100); // Wait for 2 microseconds
     counti=micros();
     digitalWrite(ultraSoundSignal, HIGH); // Send high pulse
     delayMicroseconds(50); // Wait for 50 microseconds
     digitalWrite(ultraSoundSignal, LOW);
     attachInterrupt(0,time1,RISING);
   attachInterrupt(1,time2,RISING);
   attachInterrupt(4,time3,RISING);
  }

  else   if (t1 != 0 && t2 != 0 && t3 != 0 && i == 0){

    tim1 = ((double)(t1 - counti))/1000000;

    tim2 = ((double)(t2 - counti))/1000000;

    tim3 = ((double)(t3 - counti))/1000000;

    y = (((tim1*tim1)-(tim2*tim2))*(speedofsound*speedofsound)+9)/6;

    x = (((tim1*tim1)-(tim3*tim3))*(speedofsound*speedofsound)+9)/6;

    Serial.print(x);

    Serial.print(" ");

    Serial.println(y);

    i++;
    noInterrupts();
  }
}
```

```
void time1(){
  t1 = micros();
  detachInterrupt(0);
  Serial.print(t1);
  Serial.print(" ");
  Serial.println("t1");
}

void time2(){
  t2 = micros();
  detachInterrupt(1);
  Serial.print(t2);
  Serial.print(" ");
  Serial.println("t2");
}

void time3(){
  t3 = micros();
  detachInterrupt(4);
  Serial.print(t3);
  Serial.print(" ");
  Serial.println("t3");
}
```