***Ultrasound Distance Detection on Moving Objects – A Research Project.***

***By Kieran Small – 10699216***

***Supervisor – Professor Adrian Ambrose***

Summary/Abstract

Acknowledgements

Contents

Introduction

Design Brief

Testing Methods

Project Management

Conclusions

References

Bibliography

Appendices: Project, Proposal, Work, Plan, Costings Etc Etc

# Introduction:

This project aims to accurately measure objects whilst they are moving using the medium of ultrasound to do so. The main point of this is not the application itself, i.e. distance measurement, since there are many devices which can do this, using infra-red and lidar, and although ultrasound can have its advantages compared to these, such as how it is not impacted by light and so can be used just as successfully during the day or night, ultrasound typically works over shorter distances and can be disturbed by adverse weather conditions if used outside (rain drops can cause random scattering of the signal along with changing its speed of propagation an uncalculatable amount when moving through the water) and can impeded by large changes in temperature (the speed of sound ranges between 331.5m/s at 0 degrees and 362m/s at 35 degrees which although not incredibly significant would change the same readings, in for example winter and summer). Moreover an Ultrasound distance measurer can be bought for about £5 for amateur projects which interfaces with an Arduino controller and works up to 200cm and with an accuracy with 3mm (so says the spec HC-SR04: <https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf>)\*\*. Rather the aim of this project is to use this application as a means by which the limits and something of the medium of ultrasound itself can be explored.

Ultrasound is a particularly interesting and useful medium for communication and in sensors as it is non-invasive, the technology surrounding it is fairly well understood and so ultrasound transducers are relatively inexpensive (BACK UP). It is also fairly easy to test and set up and ‘look at’ in the lab and a lot of the knowledge and practices can be transferred to higher frequencies without too great an effort which makes it a perfect medium to research and look at for this final year project. Also, since ultrasound is used extensively in medicine to give non-invasive imaging of the internal body, most scholarly articles are focused around that function and it is difficult to find article purely discussing the use of ultrasound in measuring distance and this is what this report will aim to do.

*DESIGN OF DISTANCE MEASUREMENT PROJECT*

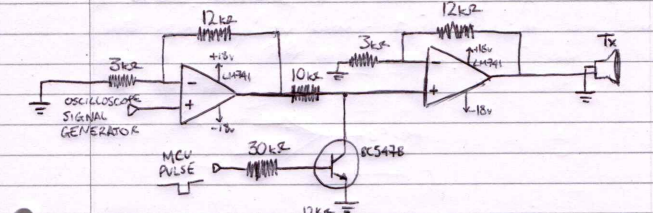
*INITIAL SET UP*

The basic principle of using ultrasound is a fairly simple one and is governed by the equation relating speed time and distance, namely that *Speed* = (1). Since sound waves travel at a constant speed through the same medium (i.e. water or air), with fluctuations in speed arising only from the temperature, and is known to be about 343m/s at room temperature though air; if the time taken for a sound wave to travel to an object and reflect off of it and return to where it was sent is measured, then how far away that object is can be calculated by re-arranging formula (1) so that *Distance = (Speed \* Time)/2* (2)*.* The calculation is divided by two since the time measured is how long it takes to reach the object *and* get back again. All that is required, then is to measure how long it takes for an ultrasound signal to be sent by one transducer and received by another which are adjacent to each other and are pointing in the same direction.

The initial idea was to send a sinewave signal as a pulse created by a microcontroller; upon sending this a timer would start. The received signal would then be rectified so that the rising edge of the pulse could signal the timer to stop and be measured. When the code was first written for this, however, it would crash due to a ‘critical error’. After further examination and discussion with the lab technicians it was determined that the microcontroller processor couldn’t run quickly enough to produce a sinewave at 40kHz without using a technique involving ‘DMC’ to access the microcontroller clock directly. It was thus decided that, since this was quite a complicated task and this project was not focused on coding, a bench oscilloscope would be used to create a carrier wave signal using the ‘generate’ function on them, to be sent and received by the ultrasound transducers. Three different frequency transducers had initially been acquired, 25kHz, 40kHz and 50kHz, so that the effect of different frequencies could be compared in finding the most accurate frequency for distance measurement and so using the oscilloscope signal generator also meant that the frequency could be quickly and easily changed without having to go back and alter the code.

*Transmitter*

In order to measure the time it took for the signal to travel to and from the transmitter/receiver unit, and also to create a finite signal pulse so a start time could easily be measured, it was chosen that the carrier wave would go through a bjt which would act as a switch controlled by the MCU. The MCU would emit a logic high (3.3V) which would strangle the carrier wave and then drop for a short period (10ms) to a logic 0 (0v) which would then let the carrier wave through for that 10ms before going high again. As soon as the MCU sends a logic 0 the timer starts and continues until the received pulse lets it know to stop it. It was also decided that the carrier signal would be amplified by a gain of 25 so that the signal could be sent a further distance and would still be picked up by the receiver and this would done by cascading two LM741 op-amps with a gain of 5 together. The initial set up for the transmitter is shown below:

*Receiver*

Text, letter

Description automatically generatedWhen the receiving transducer acquires the signal pulse it is amplified, again by 25 by cascading two LM741 op-amps with a gain of 25, in order to have a larger signal to make the analysis of it easier and in order to pass a threshold to decide when a pulse has been detected. A resistor with a 20kΩ value is added between the received signal and ground to give it a suitable impedance so that the signal is received properly. The amplified signal is then rectified using an envelope detector, going through a diode to get rid of the negative half of the signal and then using a 100nF capacitor and 30kΩ resistor.

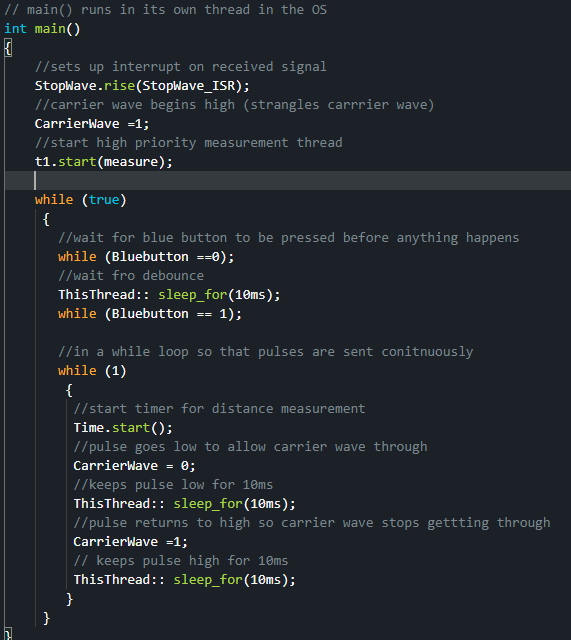
Since the speed that the rectifier drops back to zero isn’t vital as the pulse rate is very slow compared to other applications of the envelope detector where it is rectifying a signal containing ASCII information at a fast baud speed, then the values are set large to make sure a sharp rising edge and smooth line are given. The rectified pulse is then sent through a LM741 acting as a comparator. The threshold voltage was set to 1.1V, as the source for the threshold voltage was the 3.3V output pin on the MCU and so this was divided by three using a potential divider to get the threshold voltage. As soon as the rectified signal climbs higher than 1.1v on its rising edge, the output voltage of the comparator hits the top its voltage rail and otherwise sits at negative side of the voltage rail. All the op-amps on both the receiving and transmitting side of the circuit are powered off of the same power supply which is supplying -18V - +18V as this is the maximum voltage stated on the datasheet that the op-amps can be supplied with. This means that the output of the op-amp which is being used as a comparator is either -18v when no signal has been detected or at +18v when the received signal has been detected. Since the pins on the MCU can only tolerate a voltage range between 0-3.3V then the output of the comparator is first sent through a diode to put the signal at 0v instead of -18V and then goes through a potential divider to bring +18V down to an MCU safe 3V before it is attached to a GPIO input pin on the MCU to give the pulse that stops the timer. The initial set up for the receiver as described above is shown below:

A picture containing text, appliance, stove, kitchen appliance

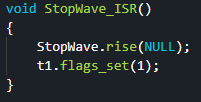
Description automatically generated

*Code*

As briefly explained above, the code outputs a square wave pulse 10ms high voltage and then 10ms zero voltage on the DAC output pin. The timer starts as soon as zero voltage begins to output. It will wait for the blue button to be pressed before beginning outputting the pulses but it was decided for testing purposes that the code would then send this pulse repeatedly so it could be easily seen and captured when looking on the oscilloscope. This code is written in ‘main’, and at the start of main a high priority thread is also initialised to collect the distance measurement data in. Main is shown below:



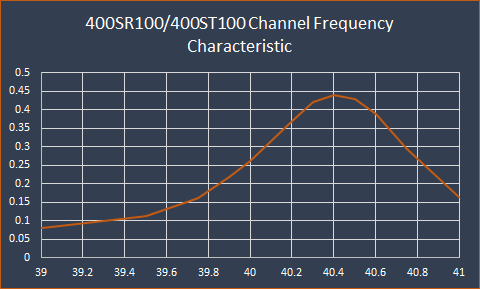
The high priority thread waits for a flag set by an interrupt service routine that fires whenever a high pulse is shown on the GPIO pin that is connected to the rectified received signal.

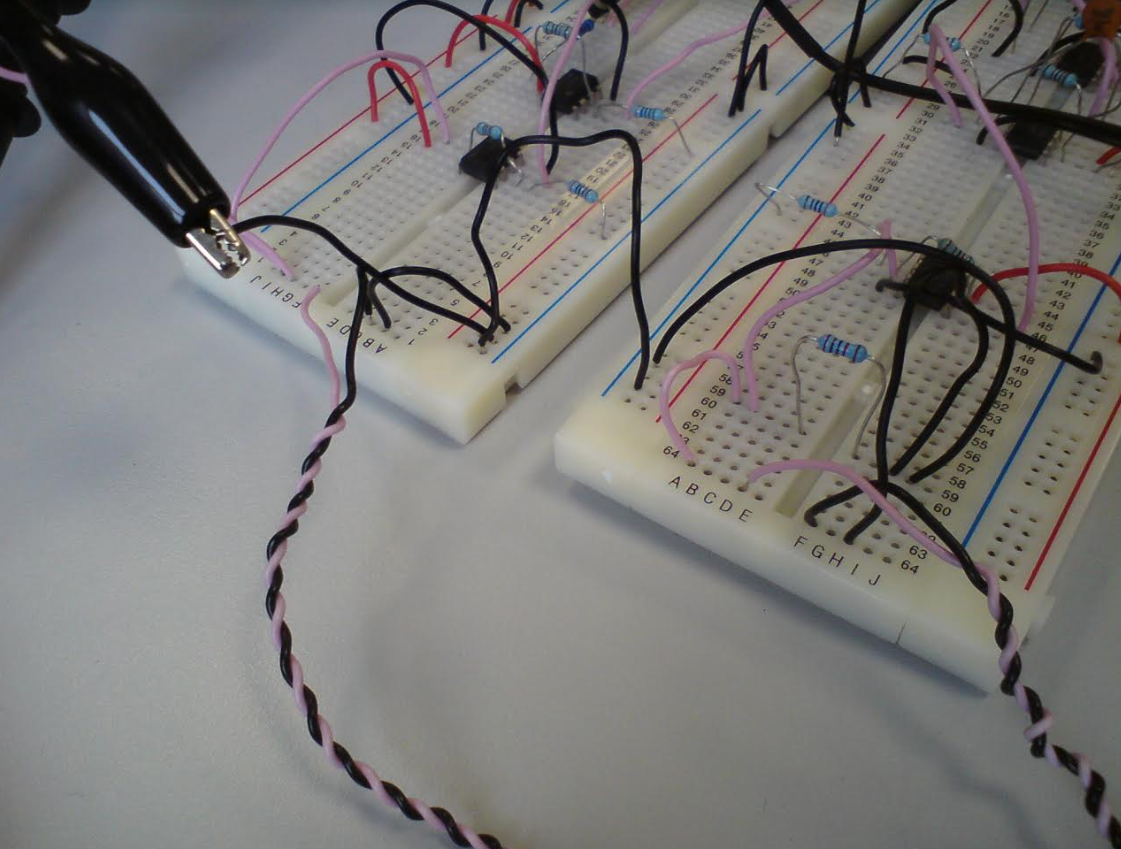


When this flag has been set the measure thread will run, taking the time reading and storing it in a variable. From this measurement it then calculates the distance the carrier wave has travelled and outputs both this and the time it took to the serial monitor before resetting and stopping the timer for the next measurement and sleeping until next distance measurement is received:



This was the original set-up of the distance measurement component, and so testing of this set-up was begun to see how well/accurately it worked. The 40kHz transducers were the first pair used, and after checking the characteristic channel frequency for these particular transducers by setting them up facing each other at a distance of 10cms apart and scrubbing through different frequencies between 39and 41kHz, it the carrier wave was set to 40.4kHz:



The set up for the initial testing was as follows. Using a cardboard as a makeshift mounting surface for the two transducers, hand twisted wires connected the transducers to two testing breadboards, one containing the transmitter circuit and the other containing the receiver circuit as can be seen in figures x and y. An A4 sheet of cardboard was used as the surface that the ultrasound signal would reflect off to give a distance measurement and was held upright at an angle roughly perpendicular to the table-top surface by a set of clamps. The distance between the end point of the transducer and the cardboard reflector was measured by a tape measure and the distance between the transducers and the cardboard was changed by moving the cardboard closer or further away from the transducers.