Ultrasonic Rangefinder

Brett Griffin - brgriffi@ualberta.ca
Matthew Johnston - mjj3@ualberta.ca
Group #4

Accurately detect the distance between the two ultrasonic transducers, using the Altera DE2 board.

Table of Contents

Abstract

Ultrasound is sound which propagates above the human audible range[1]. Ultrasonic transducers are electromechanical devices that convert electrical energy into mechanical and vice versa. Transmitting a pulse to one transducer, and measuring the time of flight (TOF) after receiving the signal on the second transducer, the distance between the two transducers can be measured. The transducers can be placed in two configurations. The first configuration results in both transducers pointed in the direction to be measured (side by side). This configuration allows for objects to be measured, granted the echo returns to the second transducer. The second configuration allows for each transducer to be pointed at each other. This allows for an accurate measurement of the distance between both units. A non-inverting op amp, high pass filter and an analog to digital converter(ADC) processes the returning signal . The design relies heavily on the DE2 development board from Altera, as well as on both hardware and software components where communication between each component is critical. A low cost, low power consumption design has been obtained, placing emphasis on the importance of part/component decisions, and possible extensions of the design for various ultrasonic applications. A heavy focus was placed on a robust test plan both in the hardware and software components of the design. The measurements produced by this system satisfy our functional requirements.

[1] Greater than 20Khz

Background

It will always be necessary to produce low cost innovative solutions to industry. In Alberta, energy is the provinces main economic driver. There is a requirement to provide low cost, innovative technologies to supplement this industry. Ultrasound is used in a wide varieties of industry in Alberta, including: oil & gas, pipelines, medicine, non-destructive testing, automotive, etc. Our goal is to design a system that utilizes the properties of ultrasound, to measure distances between two ultrasonic transducers. Our goal was to develop a system for a specific application, that is also robust, and flexible enough in terms of the hardware design that its specific functionality can be modified only with software modification. We designed the system with these goals in mind because we feel that system design involving ultrasonics is extremely relevant to industry in Alberta.

The wide application range and importance of ultrasonics is identified in [4]. Lynnwood lists the following lists of ultrasonic applications: “ultrasonic measurement of flow, temperature, density, porosity, pressure, viscosity and other transport properties, level, position, phase, thickness, composition, anisotropy and texture, grain size, stress and strain, elastic properties, bubble, particle and leak detection, non-destructive testing, acoustic emission, imaging and holography.”

When designing ultrasonic systems it is important that “the sensor [...] be robust against noise from vehicle engines and other sound sources” [1]. No matter what the application, noise eradication should be integrated heavily into the design and this will not be overlooked in our rangefinder application. For our application specifically, noise will need to be filtered to avoid incorrectly starting/stopping our timers, which are triggered by the transmitted and received ultrasonic waves, for the purposes of distance measuring. Tanzawa [1] also states that in noisy environments, “conventional sonar using the time of flight of the narrow impulse often measures incorrectly the range to an object because of [...] noises mentioned”. Although successfully designing functional circuitry for our application is imperative, focus will be places on noise filtering in software and hardware.

Safety Calculations

Ultrasound is inaudible, high-frequency noise(>20kHz) that can be dangerous to humans in some cases. Safety considerations must be taken when using ultrasound. Health Canada reports safe levels of ultrasonic exposure regardless of exposure time. For 40 kHz frequencies [5] reports that the safe threshold for humans is a sound pressure level(SPL) less than 110 dB. Although this is the safe threshold determined by the government of Canada, [5] also states that SPLs lower than 120 dB have not been demonstrated to

cause hearing loss. The transducers we will be using (400EP250) have test ratings of 113 dBmin at a distance of 30 cm with a voltage supply of 10 V_{rms}. Based on the operating voltage (V_{rms}) and the distance from the source (d), we can calculate the SPL from the following equation:

$$SPL = dB_{ref} + 20log_{10}(V_{rms}/V_{ref}) + 20log_{10}(d_{ref}/d) \quad [6]$$

Calculating the safe distance, d, with SPL = 110 dB, dB_{ref} = 113 dB, V_{rms} = 3.3 V_{rms}, V_{ref} = 10 V_{rms}, d_{ref} = 30 cm, it was found that the safe distance from ear to ultrasound source is ~28 cm without ear protection during operation. Note that V_{rms} is calculated from a 3.3 V signal. For the safety of classmates this is an acceptable distance. For our testing, we may consider using hearing protection during transducer operation, although as a safety precaution we will never expose our ears to the operational ultrasonic source at a distance less than 28 cm.

Functional Requirements

The system has one distinct purpose: accurately measure the distance between the two transducers. There are other smaller requirements. The system should minimize error as much as is possible. Being that there are two configurations for operation, there will be two requirements based on the configuration. For the configuration with the transducers positioned side-by-side, measuring the range of a nearby object it should have a maximum range of ~3-5 feet from the object. When the system is configured so that the transducers are placed opposite of each other, pointing at one another, the maximum range will increase to ~10 feet. The system should minimize error with measurements being ±1-5 inches from the actual measured distance of the transducers. It is also a requirement in our minds that the actual transducer circuitry/housing should be as small as possible for maximum system portability. To also allow for added portability, the system should be energy efficient, as it will be running off of a battery pack.

To a certain extent the design has met these requirements. The hardware-software communication/integration functions properly, and distances are typically precise to within 1-5 inches, with a range of up to ~7 ft. depending on the distance measured. There are, however, instances where the range outputted is extremely different than the actual distance. We believe this is likely due to improper signal verification to stop the timer. Although we do implement a high-pass filter in hardware to try and filter out erroneous low-frequency noise, we were unable to successfully integrate a hardware bandpass filter.

Hardware filters were not our first choice in terms of filtering the signals, but we were unable to successfully implement a filter in software from the DSP builder software. Because we do not successfully filter out high-frequency noise, any ambient high frequency noise from the environment could potentially stop the timer before the signal was actually received; this is likely the reason that the outputted ranges are not 100% accurate all the time. Typically, however, the outputted ranges are correct, and are usually within about 1-5 inches of error. Error in the outputted ranges increased as the actual range increased. This could potentially be attributed to the transducers coupling to air. A transducer with a grill interface, as opposed to an aluminium interface could potentially increase the coupling and hence increase the power transfer.

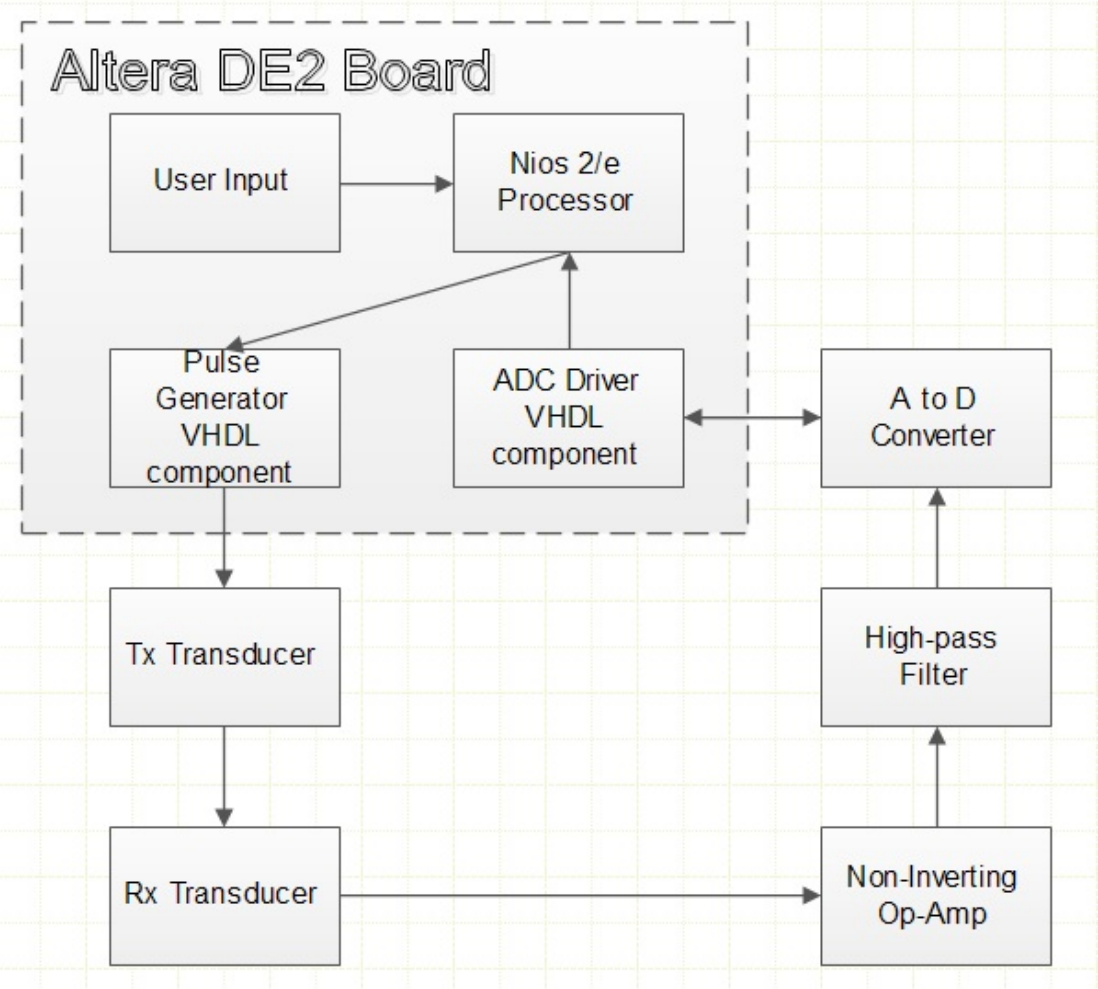
Although the end result in terms of the outputted distance measurements is not always accurate, there were several successful accomplishments in terms of interfacing our custom hardware circuit with the DE2 board. Our custom FPGA components used to drive the Tx transducer functions properly. We also successfully wrote our own ADC driver component in vhdl, ie. it functions independently of software. The ADC integration into our design was a major success in that we reliably sampled a high frequency analog signal and convert it into digital values corresponding to the Rx transducers output. We have also designed the system so that the Rx and Tx transducers modes can switch, ie. the Rx can become the Tx and vice versa, which may be necessary for other applications. We also believe that the system footprint is sufficiently small and energy efficient, satisfying the requirements above.

Design/Operation

In order to implement the above requirements we will need to interface two ultrasonic transducers with each other as well as with the DE2 board. Ultrasonic transducers are devices that convert signals into ultrasonic vibrations (and vice versa) at a specified frequency. By sending a series of square waves at a 3.3Vpk-pk value, we intend on recovering the signal in the second transducer. We can then measure the time from the first transmitted pulse to the first received pulse to reveal our propagation time. Once we have the propagation time, we can reference the speed of sound in air at our altitude(~343 m/s) to calculate the distance between the two transducers.

On the receiving end, the transducer would be receiving the ultrasonic waves and producing analog representations of the signal at 40 kHz. We would then need to amplify the signal to a ~3.3V level. This is done using a non inverting amplifier. This amplified signal is then filtered and converted to a digital signal by a serial ADC sampling at 312.5

kSPS. Once converted to a digital signal, it is interfaced to the DE2 using our custom ADC driver vhdl component via the GPIO ports on the Altera DE2 where it can be analyzed. Our timer is implemented in VHDL and depends on the 50 MHz clock pulses.



1. Hardware block diagram

The hardware component diagram that will interface the transducers to the DE2 board is found in the Appendix as Figure A.

As shown in diagram 1. above, we are using a high-pass filter in hardware that has been designed to pass frequencies >30 kHz. We were unsuccessful in created a functioning band-pass filter that would only pass frequencies at 40 kHz ± 1 kHz. Of course, we would have prefered to implement the filtering in software using the DSP filter but we were unsuccessful. The hardware filter high-pass filter we have designed is still adequate.

Signal verification is done in software by using a threshold value, and reading the values that are read in from the ADC. When a value read from the ADC is above this threshold,

we stop a timer. The threshold needs to be sufficiently high so that any lower voltage erroneous noise does not incorrectly stop the timer. It also needs to be low enough so that lower amplitude received signals, due to large transmission distances, are still successfully verified. Because the received signal is a threshold, we will be inheriting error due to the fact that the signal is not actually verified until a small amount of time after the signal was actually received in reality. For example, if the transducers are within close proximity to each other, the threshold will be met more accurately since the amplitude will be substantially greater. As we separate the transducers, we increase the distance but decrease the amplitude of the receive signal. Our threshold would therefore be detected somewhere later in that waveform. Some calculations can be done to estimate what kinds of errors we should expect. We can make the assumption that there will likely be some small delay in terms of the actual receiving of the signal, and when the software interprets a received signal. We can make a conservative estimate that a 250 *u*s error in time may occur. Although most of our testing has been done in imperial units for ease of demonstration, we will do error calculations in metric for simplicity and then convert the result to imperial. We will use a speed of sound in air of 343 m/s. By using the formula:

$$v = \frac{d}{t}$$

Where *v* is velocity, *d* is distance, and *t* is time. We do the calculation as follows for error:

$$343\ m/s\ = \frac{x}{t+250us}$$

By solving for *x* we find *x=343t+0.08575*; where 343t will equal the actual distance and 0.08575 is the error that is added to the measured distance. 0.08575 meters equals to ~3.4 inches. This shows that with a very conservative error assumption, we should still be able to measure distances within our functional requirements.

The design for a first order high-pass filter was done using a resistor and capacitor and referencing the following formula from [7]:

$$f_c = \frac{1}{2\pi RC}$$

Here, f_c is the cut-off frequency. We chose a cutoff frequency of 30kHz to be safe and chose a resistance value of $10k\Omega$. Solving for the capacitance value we calculated C=0.5nF.

The op-amp design involved testing different frequency outputs and determining the required gain for the input of the ADC. It needed to be sufficiently high to ensure a higher range of the measurements, but low enough that close measurements would not generate high voltages (> 5v) going into our ADC (although the ADC input is voltage protected). We referenced the following formula from [8] for non-inverting operational amplifiers:

$$V_{out} = V_{in}(1 + \frac{R_2}{R_1})$$

We measured the voltage of the Rx transducer at a very close distance (~3 inches) and solved for a V_{out} that would not go over 4V. We solved for R_1 and R_2 values to give us a gain of:

$$\frac{V_{out}}{V_{in}} = (1 + \frac{100k}{390}) = A_v = \sim 257$$

Bill of Materials

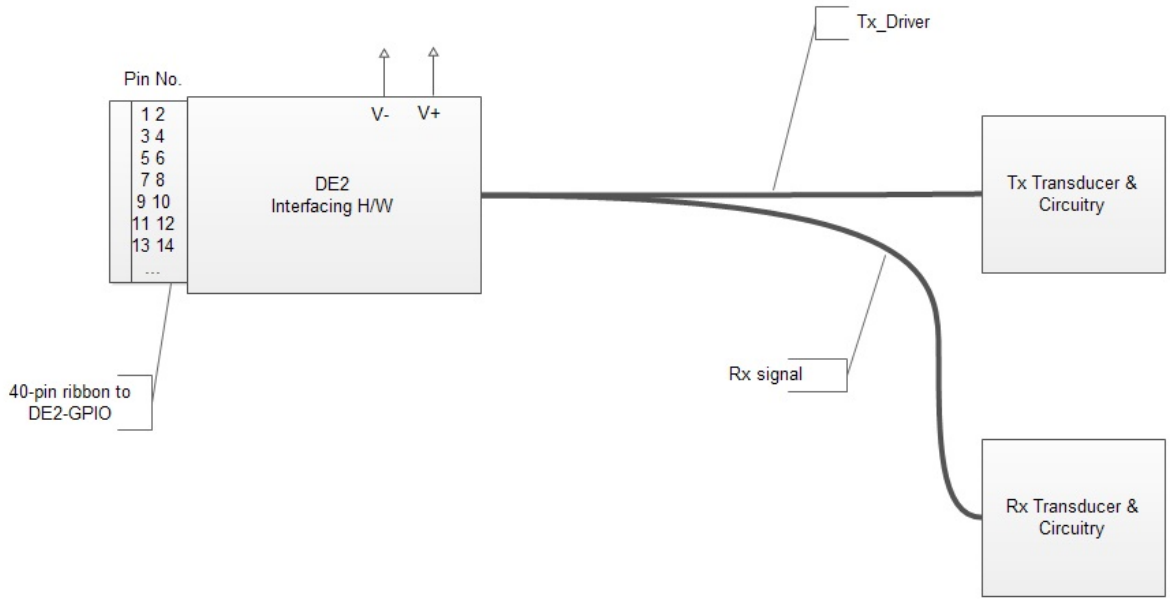
Part	Relevant Specs	Unit Cost	Quantity	Total Cost	Links
Altera Terasic DE2 Board	--	$517.72	1	$517.72	Webpage
OPA209 LOW NOISE OP-AMP	*Supply Voltage:* ±2.25 V - ±18V	$3.15/unit	1	$3.15	Datasheet
400EP250 Prowave transducer	*Resonant Frequency:* 40Khz *MAX* $V_{driving}$*:* 20 V_{rms} *Nominal Impedance:* 300Ω	$15.26/unit	2	$30.52	Datasheet
AD7810 350KSPS SERIAL ADC	10-Bit Serial Output *Sampling Rate*: 350 kSPS *Supply Voltage*: 2.7 V - 5.5 V	$8.00/unit	1	$8.00	Datasheet
LM7805 Voltage Regulator	Output Voltage: 5V Input Voltage: 5-18V	--	1	$0.74	Datasheet
LM7905 Voltage Regulator	Output Voltage: -5V Input Voltage:-5 - -35V	--	1	$0.67	Datasheet
40-pin ribbon cable	--	$10.00/unit	1	$10.00	--
10 ft. PVC pipe	Pipe used to measure distances along	$13.00/unit	1	$13.00	--

MISC. transducer housing hardware & circuit boards	Housing hardware holding transducers and circuitry	$7.00 + $3.00	2	$20.00	--
Circuit board	Board for hardware interfacing circuitry	$7.00	1	$7.00	--
Pipe mountaing brackets	Brackets used to mount transducers to pipe	Machine shop	2	Machine shop	--
Battery Pack and Batteries	AA Batteries	$13.56	16	$13.56	--
MISC resistors & capacitors		~$1.00	--	$1.00	--
TOTAL	--	--	--	$625.36	--

Reusable Design Units

All I/O to and from the DE2 board can be controlled using the buttons, switches, LEDs, and LCD already present on the board, so it will not be required to write new drivers for external control devices as we don't require any. We will be able to use all previously written drivers/uC code for the Altera DE2's onboard I/O devices. A list of the reused altera core components used on the DE2 board is as follows:

- *NIOS II/e processor*
- *onchip memory (RAM or ROM, 16 kbytes)*
- *sysid peripheral core*
- *interval timer (32 bit counter size)*
- *jtag uart core*
- *optrex 16207 character LCD core*
- *green/red LEDs (PIO core)*
- *ALTPLL (phase locked loop)*
- *sdram controller core (8 Mbytes)*
- *button (PIO core)*
- *CFI flash core (4 Mbytes)*
- *avalon-MM tristate bridge*



2. User-perspective block diagram omitting DE2 board

***see Table III for signal names, descriptions & corresponding GPIO pin assignments**

****complete hardware schematic can be found in the Appendix (Figure A)**

Table I. Measured performance specifications

Measured at: DE2	**Peak**	**Idle**
Current (mA)	0.463	0.463
Voltage (V)	9.03	9.03
Power (mW)	4.18	4.18

Measured at: +5V Rail	**Peak**	**Idle**
Current (mA)	0.065	0.065
Voltage (V)	4.62	4.62
Power (mW)	0.3	0.3

Measured at: -5V Rail	**Peak**	**Idle**
Current (mA)	0.022	0.022
Voltage (V)	-6.99	-6.99
Power (mW)	0.154	0.154

Table II. Operating Conditions

	MAX	MIN	Tested
V+ (V) *based on voltage regulator specs	35	5	9.6V (4 x AA Batteries)
V-(V) *based on voltage regulator specs	-35	5	-9.6V(4 x AA Batteries)
Temperature *based on component benchmarks; NOT TESTED	70□C	-30□C	Not Tested

Table III. Signal names, descriptions, corresponding GPIO pin assignments if applicable

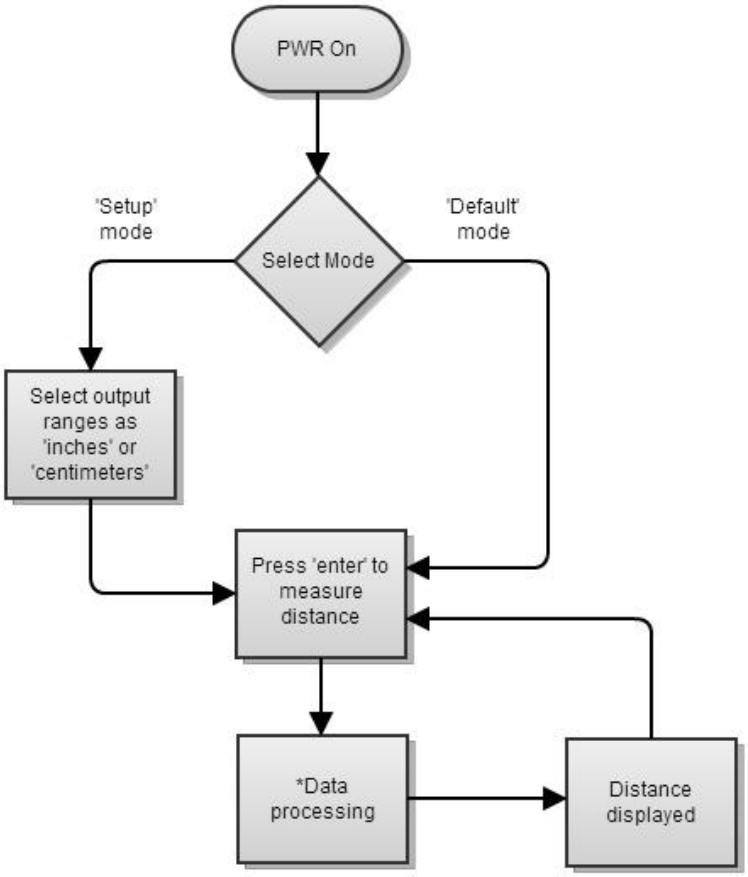
Signal Name	GPIO Pin No.	Description	Location
Tx_Driver	7	Transmitter transducer driver signal	From DE2 via GPIO0-7 through to transmitting transducer
Rx signal	N/A	Amplified received signal output from receiving transducer	From Rx transducer to interfacing H/W
sclk_adc driver	1	sclk signal from DE2 board used to clock out data from ADC	From DE2 to ADC on interfacing H/W via GPIO0-1
convert_start	2	signal from DE2 board used to initiate an ADC conversion	From DE2 to ADC on interfacing H/W via GPIO0-2
MISO_ADC	3	data out of ADC	From ADC on interfacing H/W to DE2 via GPIO0-3
Tx_EN	5	enable signal used to control Tx transducer operation	From DE2 to interfacing H/W via GPIO0-5
GRND	12	ground pin from DE2	From DE2 to interfacing H/W via GPIO0-12
V+	N/A	Positive power	From battery pack to interfacing H/W via power connection

V-	N/A	Negative power	From battery pack to interfacing H/W via power connection

Software Design

User Interaction

There are two aspects of our software design. One being the user interaction/operation code that is somewhat superficial and allows the user to properly use our system, and the other being the critical code involving all transducer & ADC driver controls, data management, and calculations. The program flow for the user operation is as follows:

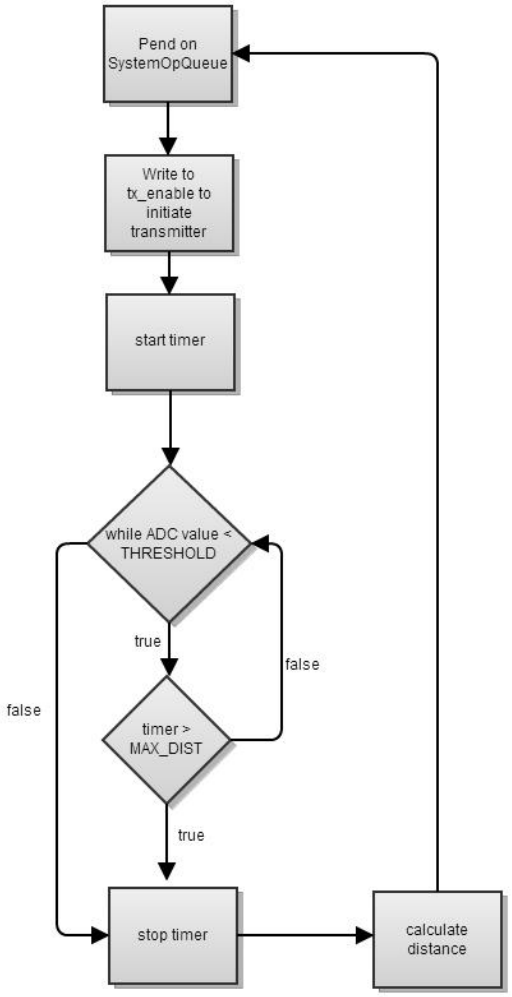


None of the code that applies to this flow chart deals with data acquisition and data processing. A message queue is used to synchronize two tasks; a control task, and and LCD task. Another message queue is used to update the control task based on interrupt driven button presses. The control task pends on a queue that is posted to by the ISR of the

button presses. This queue contains a message that represents which actual button was pressed. The control task will update another message queue that the LCD task pends on. This message queue signals to the LCD what state the system is in so that the proper messages are displayed. Finally, when the system is set-up a final state is reached where the system is ready to operate. The control task posts to a third queue that the System Op task is pending on. The system Op task is the task containing all of our transducer and adc driver commands. This system Op task now communicates with the LCD task for distance displays.

Data Processing

As mentioned, a system op task pends on a queue that is posted to by the control task when system operation is ready to begin, ie. a measurement has been requested. The data flow of the system op task is as follows:



The system op task is simply starting a timer, and reading converted values of our received signal and comparing those values with a threshold value. If a value is read that is above

that threshold the timer is stopped and the distance is converted. We will note here that we are only reading the converted ADC data values at the speed at which the while loop iterates. It is not necessarily reading those values as soon as they are converted. We realize this is a source of error because we may be reading the values slightly after they were actually converted. Our ADC is sampling at 312.5 kSPS, or every 3.2 *u*s, and our *while* loop is not iterating perfectly in conjunction with the samples being read in. In other words, the *while* loop 'sampling rate' will be more than likely out of phase with the ADC's sampling rate. This would ultimately lead to timing discrepancies in analysis of the data. Included below is an assembly breakdown of this loop:

```
while(IORD_16DIRECT(ADC_DRIVER_0_DATAOUT_BASE,0) < THRESHOLD){
 65c:    00804074     movhi    r2,257
 660:    10a43104     addi    r2,r2,-28476
 664:    1080002b     ldhuio    r2,0(r2)
 668:    10800488     cmpgei    r2,r2,18
 66c:    1000061e     bne    r2,zero,688 <tasksystemop+0x104>
if(IORD(MYTIMER_0_TIMEROUT_BASE,0) > MAX_DIST)
 670:    00801034     movhi    r2,64
 674:    10800104     addi    r2,r2,4
 678:    10c00037     ldwio    r3,0(r2)
 67c:    008001b4     movhi    r2,6
 680:    10a95404     addi    r2,r2,-23216
 684:    10fff50e     bge    r2,r3,65c <tasksystemop+0xd8>
break;}
```

*Italic font represents c code

Since we are using the NIOS II/e processor core the number of cycles for this loop to iterate once is 73 cycles. This amounts to 1.4*6u*s with a 50Mhz clock, between each sample read of the ADC. The ADC samples at 312.5 MHZ which amounts to 3.2*u*s per sample. So we are effectively sampling the ADC's sampling rate a maximum of floor(2.19) times/ADC sample.

The data processing algorithm has a couple checks to ensure that the system does not hangup. If the timer value that is read on each while loop iteration is larger than a value we have specified through testing, then the loop breaks and the user is notified that the transducers are out of range.

Test Plan

Hardware Testing:

1. Op Amp Functionality

We need to ensure that our Rx transducer output signal is properly high enough to be analyzed. An op amp with gain greater than ~200 will need to be tested until our signal is of sufficiently high voltage. With such a large gain, it will be prudent to ensure that there is no voltage drift in our output. It is important to note any 1/f noise introduced from the op amp. A potentiometer can be used to change the gain of the OpAmp until we are satisfied that the signal has been amplified sufficiently. This is stage one in ensuring the transducers are operating correctly.

2. ADC Functionality

In order to analyze our signal on board the DE2, we need to convert it to a digital signal. Here an analog to digital converter is used. Since our detection implementation only needs to verify the correct amplitude of the signal, the resolution does not need to be very high. Our 10-bit resolution produces 1023 possible outputs. We also note that in order for the ADC to produce a viable signal, it needs to sample the input at a minimum rate to overcome the Nyquist sampling frequency to prevent aliasing. The Nyquist sampling frequency is equal to 2*B, where B is the highest non-zero energy frequency. In our case, the minimum frequency will be 160 Khz, amounting to 4 samples per waveform . We chose an ADC with a max sampling rate of 350 kSPS and have it set up with our ADC driver .vhdl component to sample it at 312.5 kSPS. The 312.5 KSPS is derived from out system clock at 50 Mhz. The 50 Mhz signal is divided down by integer values resulting in 312.5 KSPS. Initially, we connected a positive constant voltage starting at 0V to the data input of the ADC and printed out the converted data values in hex format. We then would increment the input voltage on the input data pin of the ADC up to 5V. From this stage we input a positively biased low frequency sine wave from the frequency generator and stored the converted values. We read these values and graphed them in excel to ensure that we indeed had a sine wave. Once this was verified we integrated the ADC with our interfacing circuitry and transducers.

3. Filter Functionality

Due to the fact that noise is an issue with our project, and we were unable to successfully create the necessary filters in software, we decided to design a first order high-pass filter in hardware to filter out any low frequency noise. We used a simple capacitor-resistor design and tested the filter by passing low frequencies through the filter and measuring the output. Low frequency inputs should not pass through the filter, while high frequencies (>30 kHz; as designed) will pass through. Once verified we were ready to integrate this component into our design.

Software Testing:

1. User interface

The code for the user interface can be easily tested using the LCD display. The four buttons on the DE2 will be used as the user input. The LCD can be used to verify that proper program flow occurs based on the user's inputs. The user interface has been verified to be operational.

2. Data processing

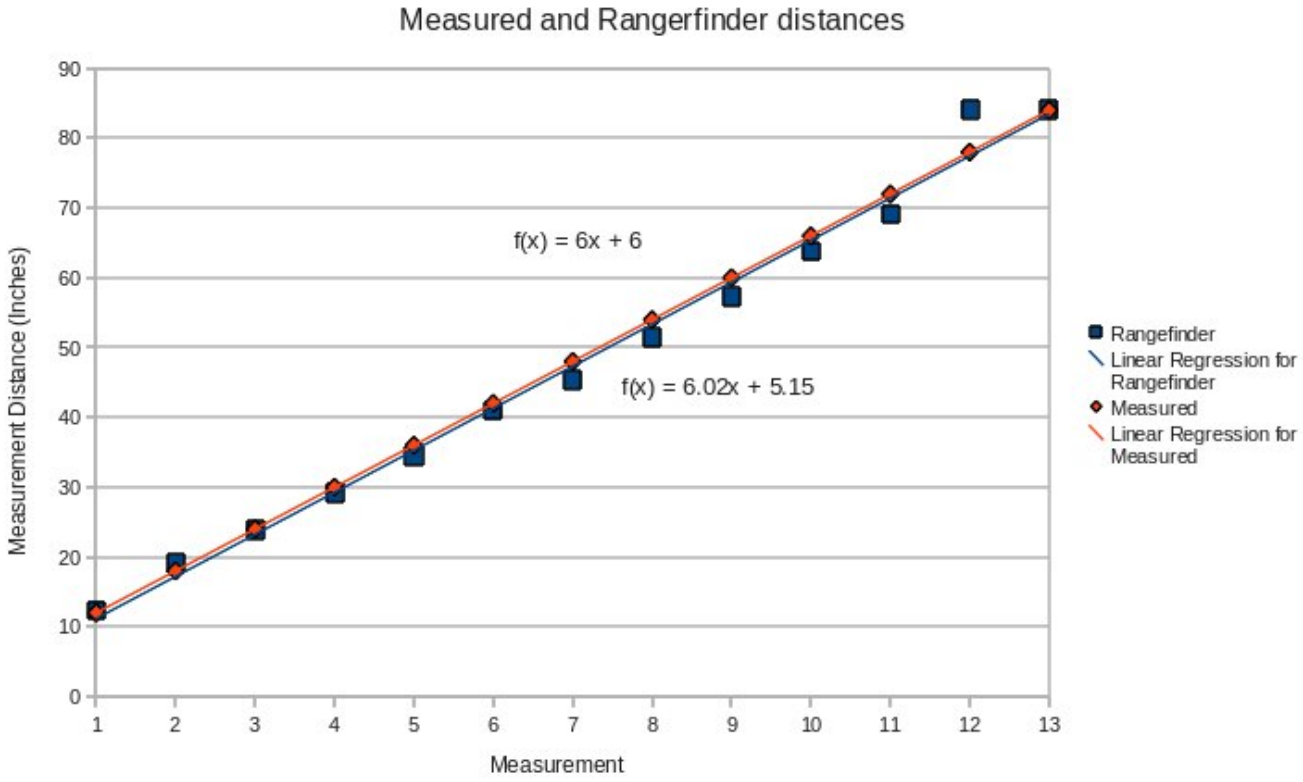
The data processing testing was started after the hardware was all tested and verified to be operating correctly. At first, we simply initiated a Tx signal transmission by writing to our Tx_enable .vhdl component's base address with relevant data. We then polled the ADC and stored the data in an array. After we were finished polling data we printed the results and plotted the ADC output. The plot corresponded to the actual signal that was measured on the Rx transducer using an oscilloscope. Once the transducers were communicating with each other, and relevant data was being piped into our ADC_ driver .vhdl component we wrote a signal verification algorithm, and moved on to total integration testing.

Integration Testing:

When all hardware components were completed their black box testing, they were integrated with each other and system operation was tested. We used the NIOS II IDE to drive our transducers and convert data using the ADC. The complete system was set-up for operation and clock ticks of our distance timer were outputted on the LCD for verification; these clock ticks were then converted to times, and distances were calculated by referencing the speed of sound in air. Data would be plotted to ensure the system operated within our functional requirements, which will be explained in the next section.

Experimental Results

Once we verified that the system was operating properly we ran a series of tests to calibrate the system and make it as accurate as possible over a range of 0.5 to 6.5 ft. distances. Once calibrated, we logged the outputted values at half foot increments with 3 measurements taken and averaged the values. We also measured the actual distances using a tape measure. The following is a plot of the actual distances and the corresponding averaged outputted distances of our system:



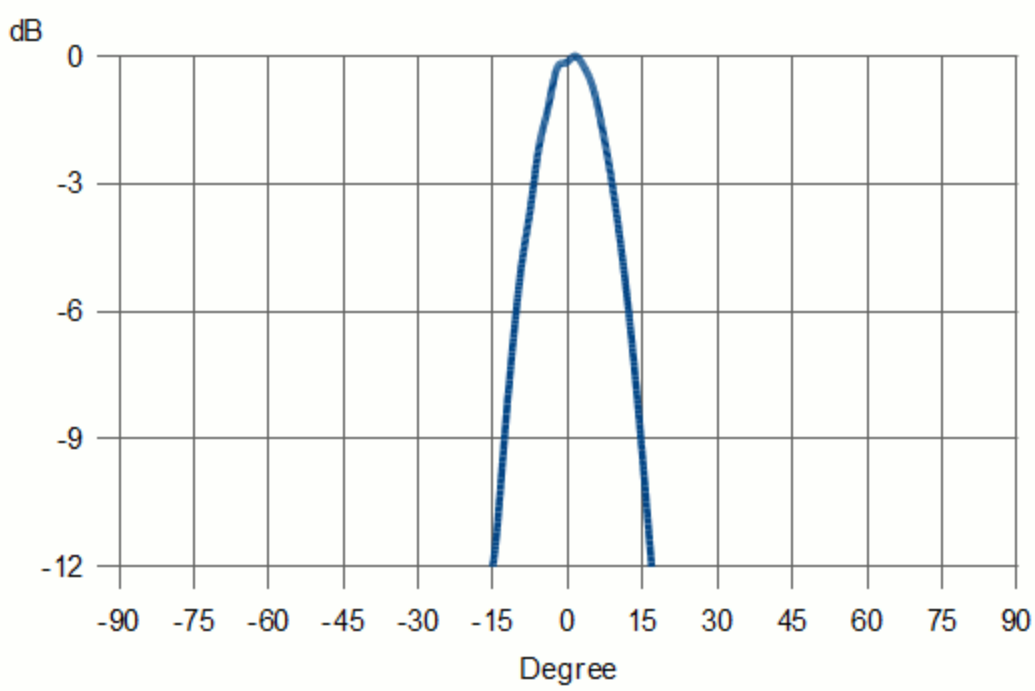
The plot shows that the outputted distances corresponds with the actual distances quite well with the slope of the two plots differing only by 0.02 inches per measurement, on average.

Although our project demo has the transducers mounted on a pipe, ensuring they are pointing directly at each other, we also tested the device for accuracy and reliability when the transmitting transducer is set up at angles larger than 0 degrees with respect to the signal path. At distances of 12 and 24 inches we took 15 measurements at each angle and recorded the average.

Table IV. Tx transducer @ angles results

Distance (inches)	Avg output @ 0°	Avg output @ 15°	Avg output @ 30°	Avg output @ 45°
12	12.15	16.67	13.86	12.22
24	23.11	N/A	N/A	N/A

Below is a chart from the 400EP250 datasheet regarding the beam angle of the transducer at 40 kHz:



3. Beam angle of 400EP250 transducers according to datasheet

By referencing this chart, we would assume that with angles >15° that the reliability of the system would decrease. This is reinforced with our testing. At distance of 1 ft. we were still able to produce somewhat reliable distances at angles from 0-45 degrees. Although precision was certainly reduced at larger angles at 1 ft. the system was still somewhat reliable. At 2 ft., however, at angles >0° the system almost never produced a reliable result. Averages were not attainable because over 50% of the time the system produced an 'Out of Range' output.

Citations

[1] T. Tanzawa, “The ultrasonic range finder for outdoor mobile robots,” in *Intelligent Robots and Systems 95*, Pittsburg, USA, 1995, pp. 368-373.

[2] MATLAB version 7.12.0. Natick, Massachusetts: The MathWorks Inc., 2010

[3] Analog Devices, AN-877 Application Note: Interfacing High Speed ADCs via SPI, 2005-2007.

[4] Lawrence C. Lynnworth, “Industrial Applications of Ultrasound-A Review,” IEEE Trans. on Sonics and Ultrasonics, vol. 22, no. 2, pp. 71-101, March, 1975.

[5] Safety Code 25, “Guidelines for the Safe Use of Ultrasound”, Health Canada, 1991.

[6] Prowave Electronics, AP050830 Application Note: Selection and use of Ultrasonic Ceramic Transducers, 2005.

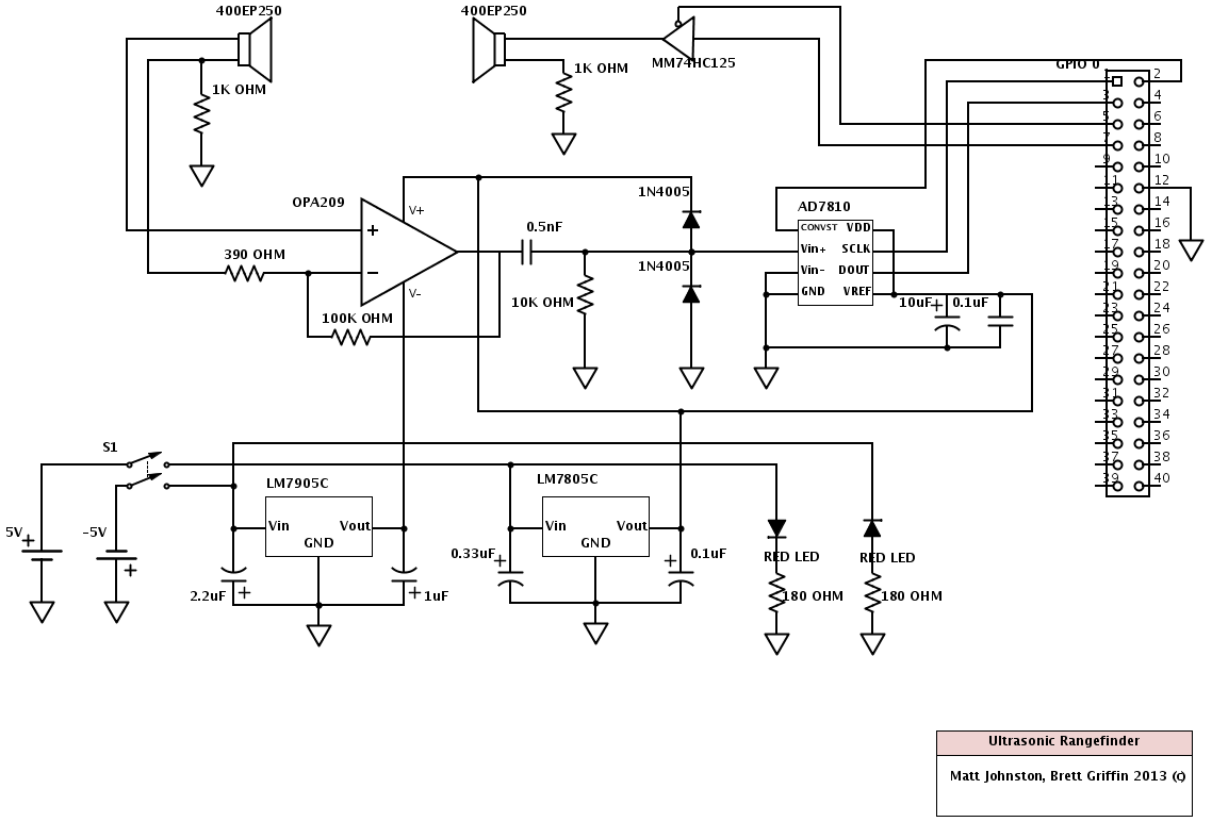
[7] “High Pass-Filter.” Internet: http://www.electronics-tutorials.ws/filter/filter_3.html [Mar 2013]

[8] “Non-inverting Operational Amplifier.” Internet: http://www.electronics-tutorials.ws/opamp/opamp_3.html [Mar 2013]

Appendix

Hardware:

Figure A-Hardware Schematic



Software:

Figure B-UI software flow

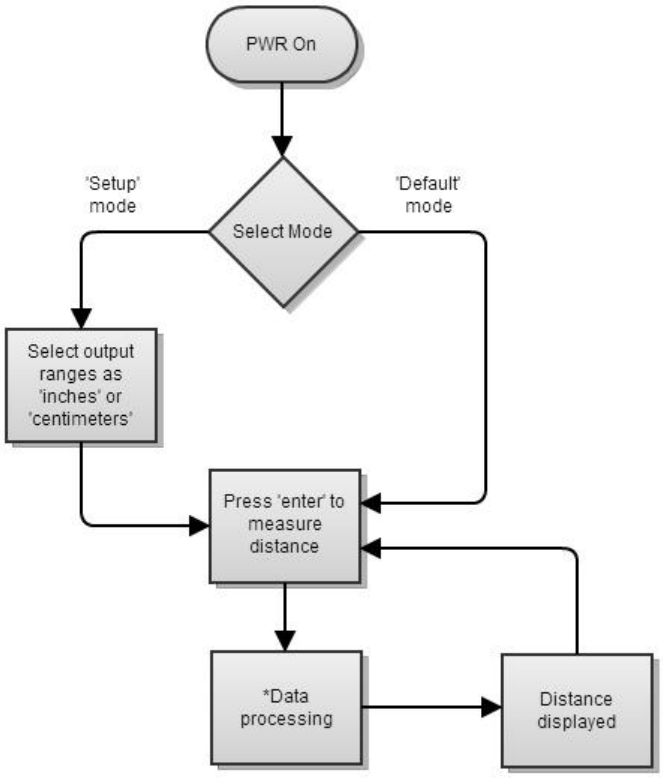
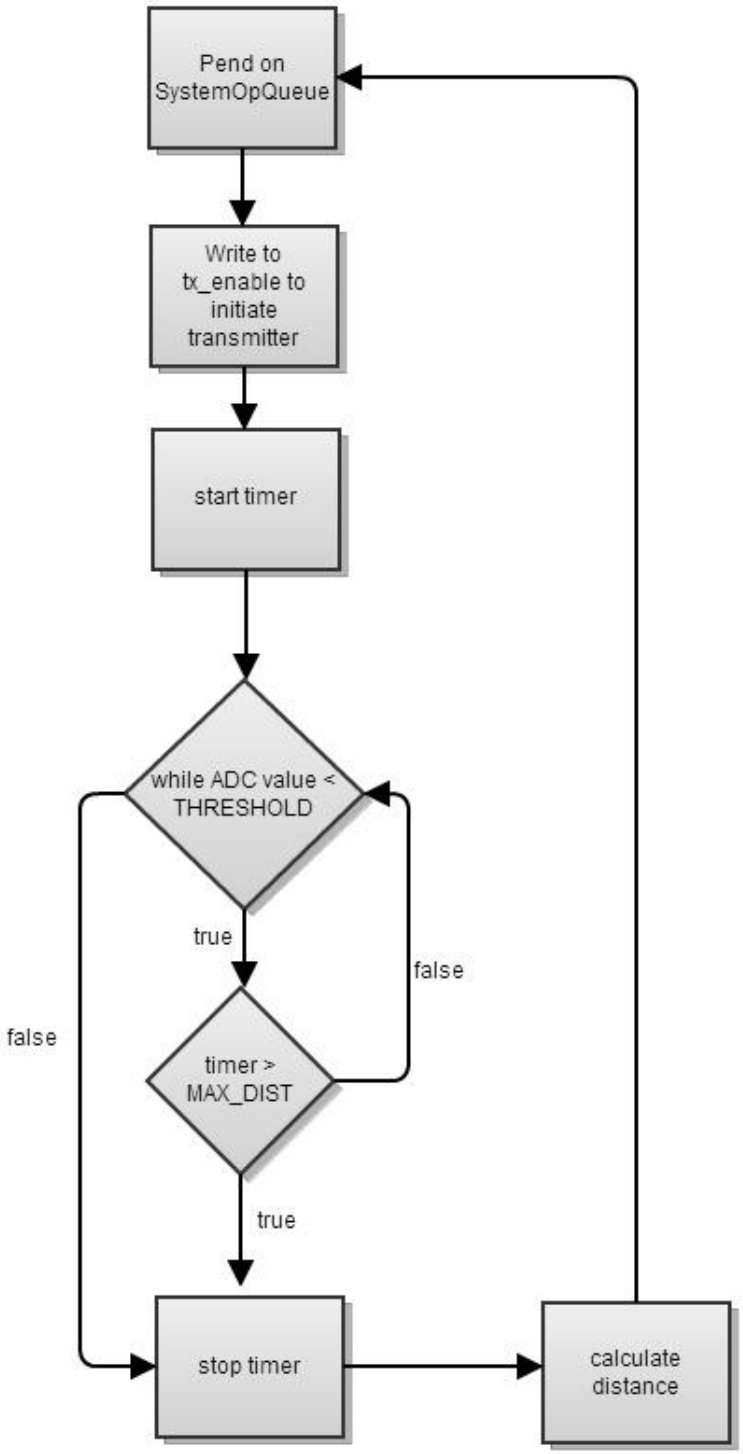


Figure C-Software data processing flow

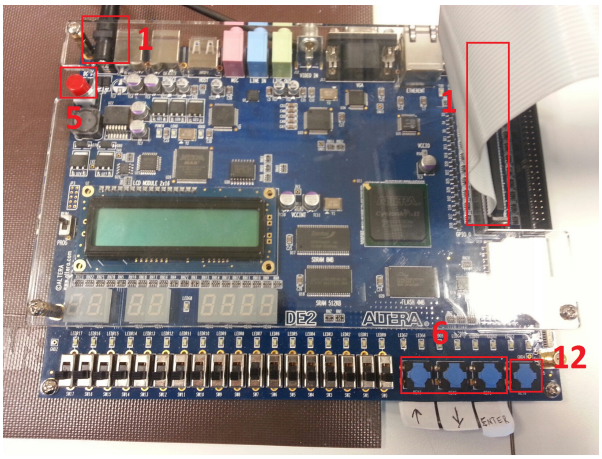


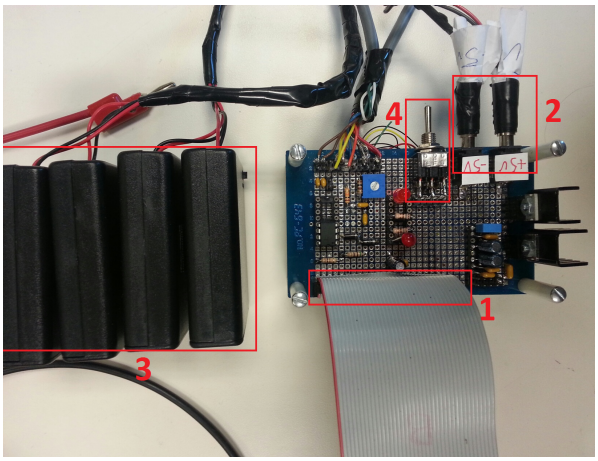
Note: All source code has been submitted as a .zip archive

Quick-Start User Guide

1) Plug in DE2 board and connect ribbon cable from the DE2 GPIO-0 to connection on off-board circuitry
2) Plug positive rail battery pack to '+5V' connector on off-board circuitry, and negative rail battery pack to '-5V' connector on off-board circuitry
3) Switch battery pack switches to 'ON' on all switches
4) Switch on off-board circuitry by flicking switch down towards power connectors
5) Turn on DE2 by pressing red power button
6) Press enter at start-up screen
7) Choose 'Setup' or 'Default' mode; if 'Default' mode chosen, skip to step 9)
8) Choose output units of inches or centimeters and press enter
9) Ensure ear-protection for operator is in place, and warn others to be >2m away from transmitting transducer
10) Point transducers at each other and press 'enter' to measure a distance.
11) Press-enter again to measure more distances.
12) Press reset button to start-over

***If output reads "Out of Range" either try again until distance is displayed, or move transducers closer





Future Work

After gaining an understanding of how to integrate a system involving ultrasonic transducers with the Altera DE2 board we feel there is a wide variety of possibilities involving extensions and alterations of design based on different applications. An interesting extension of our design would be to use it as a pipeline leak detector. The theory is all the same, and the hardware setup is the same. By mounting our transducers on the outside of a pipe, at an angle pointing at each other and measuring the time it takes for a signal to pass through the pipe, as well as the fluid medium inside the pipe, we should be able to measure when the velocity of the fluid inside the pipe changes. Changing transmission times of the signal would imply a changing velocity of the fluid in which the signal is travelling. A changing velocity of the fluid could imply a leak in the pipeline.

Although the hardware setup of our system is suitable for this application, there would need to be many improvements in terms of DSP and signal verification. As of right now, our system is susceptible to noise, and also does not have the precision necessary to be able to detect fluctuating transmission times at the precision that would be required. Coupling the transducers to the pipe would also be an issue. Any frequencies resonating inside the pipe that are not generated by the transmitting transducers would be amplified greatly by the receiving transducer coupled directly to the pipe. A very robust, and reliable filter would need to be implemented to ensure that all signals received are those generated by our 40 kHz transducer. It would be advantageous to write this filter component in .vhdl so that it could be implemented at the hardware level and be independent of software for enhanced speed and precision. A cross-correlation algorithm would likely be a good starting point for signal verification, but this would be very challenging to implement and also debug as a hardware implementation. A cross-correlation algorithm would increase the signal verification regardless of the noise in the signal. By sweeping the transmit signal between the low and high bandwidth of the transducer, a unique signal can be generated. This signal is then correlated with a discrete representation of itself in a given window size. Care would have to be taken, to ensure that the window size is not too large, which would cause large delays in the calculation.
In terms of adding features to the rangefinder system we would still like to increase the signal processing and signal verification processes. As of right now, the system is susceptible to noise and whether we implement better filtering in software or hardware, it is definitely something we want to improve on. There are a few software features and hardware modifications we would also like to add. One of these would be a calibration mode that the user can choose that would calibrate the system. The theory is that the user would set up the transducers at 1 ft. increments, the device would measure what the distances measured were and if necessary the system could add a correction factor to the

distances so that they are more accurate during normal operation.To allow for a greater measuring distance, and enhanced accuracy, the driving voltage could be increased, granted the proper safety precautions are taken.