Speaker

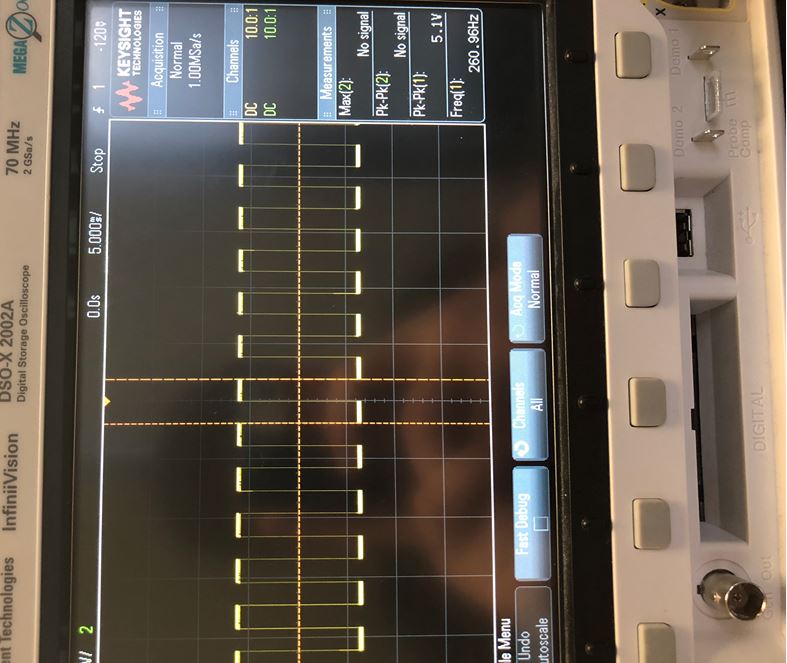
System architecture, speaker.

2.1 A: The speaker uses a ground and an input wire, it has an internal resistance of 8 ohms and a power dissipation of 0.5 Watts. This speaker uses a PWM to generate sounds corresponding to different tones, and for this project we used square waves to generate pulse waves, but it can also be driven using sinewaves. The higher the frequency the higher the pitch. To calculate the desired frequency we used this formula provided by the ATmega644A data sheet Fout = Fclk/(2\*pre-scale \* OCR0A). The speaker interacts with echolocator because the outputted tone depends on the distance found by the echolocator. The ATMega644A also interacts with the speaker as it supplies the correct pulse waves depending on the distance found by the echolocator.

3.1 A: When using the speaker, initially we found a lot of interference from the speaker when outputting a tone, because of this we decided to use a 0.47 µ Farad capacitor. The Intended use of this capacitor was to smooth out the signal to eliminate the background noise while maintaining the correct tone. We knew the speaker had an internal resistance of 8 ohms and using this information we could create a low pass filter to eliminate the noise. We attached the speaker in series with the capacitor to ground, as to create a low-pass filter. We chose to use a low pass filter because the range of frequencies was between 261 - 493 Hz which was relatively low. We calculated the cut-off frequency using the formula Fc = (1/(2πRC)) to be Fc = 42.3 KHz. This allows all frequencies below the cut-off frequency to pass, as well as the capacitor smooths the signal, which eliminates most of the noise. The expected behaviour of the speaker is to output the appropriate frequency for the correct distance within a margin of error +/- 10 Hz

3.1 B Program

For the speaker code we decided that we wanted to have PB3 as the output pin to accommodate for the limited space on the board, as to make it easy to wire on the board. In the main function we did this by writing only PB3 as an output, we then initialized the counter by writing TCNT0 to 0. Next, we wrote TCCR0A to equal 1 on WGM01 and a 1 on COM0A0 as to enable CTC mode. We then pre-scaled by 1024 by writing a 1 to CS02 and CS00 on TCCR0B. the last step was to test the code using a value, in this case we tested this with OCR0A = 27 which was 261 Hz. We verified this program worked by first setting up an oscilloscope to test if the frequency was correct.



Based on the oscilloscope the frequency was the same as what we had calculated and predicted. We also wanted to verify the accuracy of the speaker, so we also outputted the tone and checked the value through a frequency detector, and the frequency was the same as expected. For the final code we had the same initializations as in the individual code, but this time we wrote it in its own function and called the function in the while loop. We also added if statements corresponding to the specific distances for the specific tones, so when the distance was a certain value the speaker would output the correct frequency for the distance, we verified this using the same methods as mentioned previously, and the results were the same.

Flow chart:

System powered

Is Trigger pressed

Yes No

State = ranging

State = Waiting no frequency emitted

Is an object detected

Yes No

Is the distance between 1 and 36 cm

Yes No

OCR0A = 27, Output frequency of 261 Hz.

Is the distance between 37 and 72 cm

Yes No

Is the distance between 146 and 181 cm

Is the distance between 110 and 145 cm

Is the distance between 73 and 109 cm

OCR0A = 27, Output frequency of 261 Hz.

Is the distance between 182 and 218 cm

Yes No

OCR0A = 27, Output frequency of 261 Hz.

Is the distance between 219 and 254 cm

Yes No

OCR0A = 27, Output frequency of 261 Hz.

Is the distance between 255 and 290 cm

Yes No

OCR0A = 27, Output frequency of 261 Hz.

Is the distance between 291 and 327 cm

Yes No

OCR0A = 27, Output frequency of 261 Hz.

Is the distance between 328 and 363 cm

Yes No

OCR0A = 27, Output frequency of 261 Hz.