EC 450 Final Project Report

Team 10: Mertcan Cokbas & Dong Hyun Kim

- ## Goal and Design of the project

The goal of this project was to build a system that would measure the distance of a near object and the rate of change in distance. We also aimed to display both of those data for the user. To accomplish this goal, we designed our MSP432 board to connect to an ultrasonic chip (HC-SR04) that would allow us to measure how far or close an object in centimeters and inches.

- ## Description of the implementation including system level and component level interaction

In order to first connect the ultrasonic chip to MSP432, we had to match the voltage levels of these devices; MSP432 works with 3.3 volts, whereas the ultrasonic chip works with 5 volts. To match the voltage levels, we used a bi-directional logic level converter (BOB-12009), which required a 6-pin header. A small problem arose here where the level converter seemed not to be working because it was not touching the header. This problem was solved by soldering the 6-pin header to the level converter so that they would be completely connected. The way the ultrasonic chip works is that once it receives a trigger signal from the MSP432 board, ultrasonic chip sends 8 cycles of 40KHz ultrasound wave to the environment; after the 8th cycle, the echo signal become HI, and it stays HI until the ultrasound wave comes back. Knowing this process, our challenge was to first trigger a pulse and accurately measure the length of the echo pulse.

To trigger a 15 micro-second pulse (greater than 10 micro-second pulse was enough according to the data sheet), we first tried to use Watchdog Timer, but WDT gave a pulse only every 800 milli-second, which was too long to obtain frequent samples. As a result, to be able to generate pulse at every 100 milli-second, we opted for Timer A, which better handles PWM signals. We chose to send a pulse at every 100 milli-second because the data sheet suggested to use greater than 60 milli-second in order to prevent echo and trigger signals from overlapping. We connected

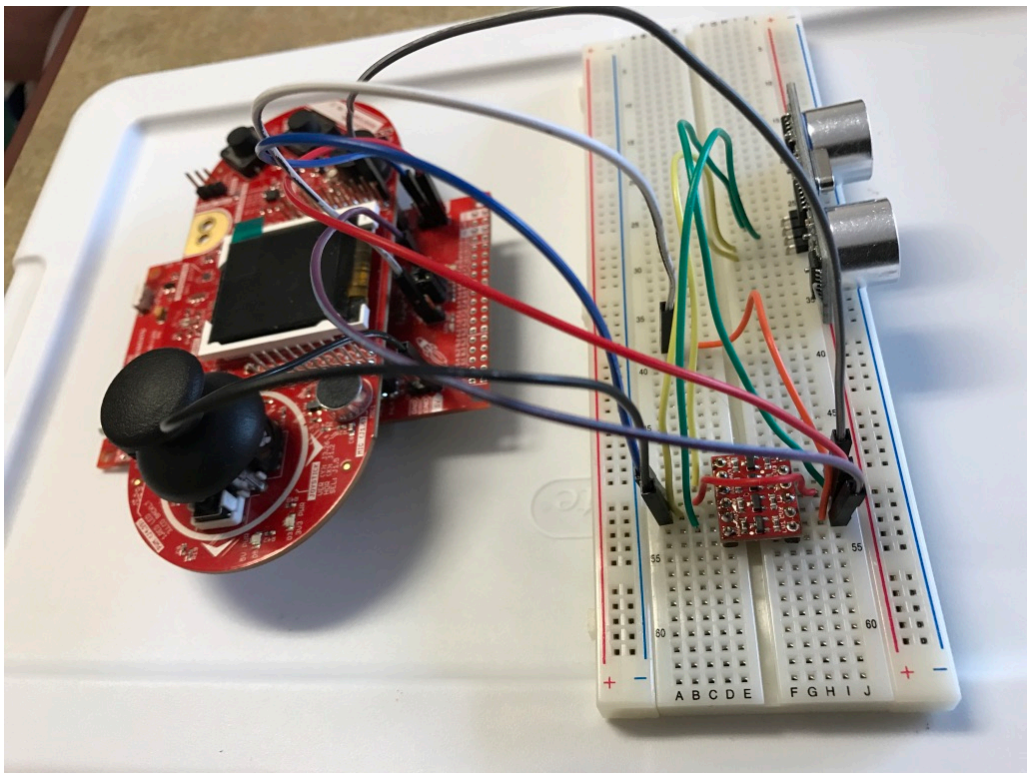
P3.6 to the LO side of the level converter, and the HI to the trigger pin of the HC-SR04.

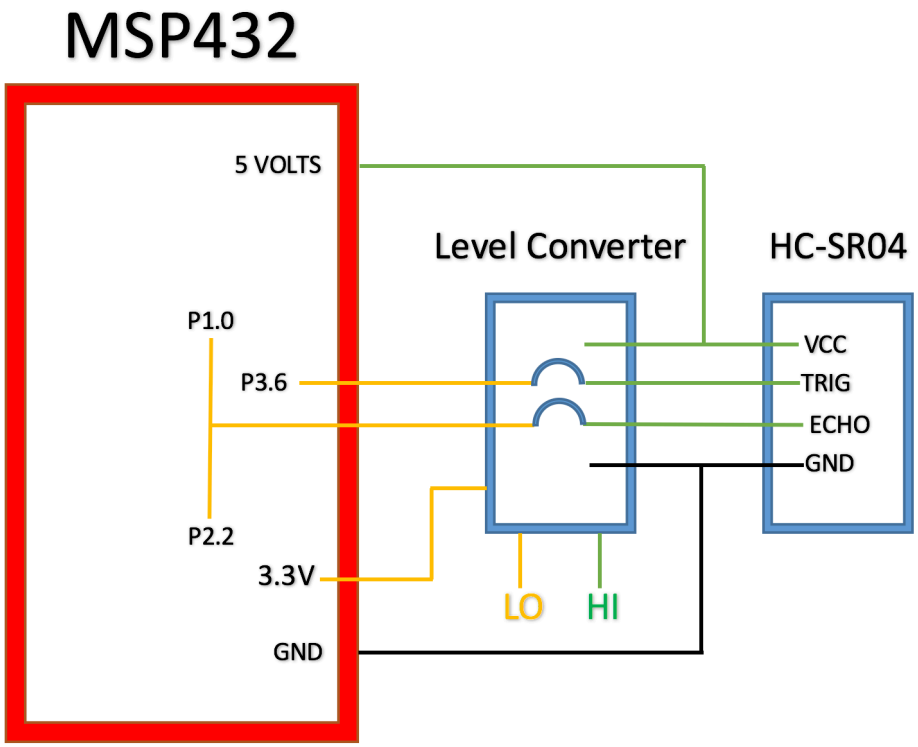
Now to measure the length of the ECHO signal, we first tried to use TimerA Capture Mode, which happened to be perfect for our application as it could sense both the rising and the falling edge of the signal, but we encountered some unknown errors during implementation. Our solution was to instead use two different GPIO pins, one for rising(P2.2) and the other for falling edge(P1.0). With the help of Timer32, P2 Handler stored the start time of the pulse in a global variable in, and P1 Handler stored the finish time in another global variable. Also in P1 Handler, the value of the start time variable is subtracted from the finish time value, and the resulting difference is divided by the speed of the sound (340m/s), and another factor of 2 to account for the wave travelling back and forth. In addition, a global variable within P1 Handler is set to 1 to tell the main program that there is a new result to be displayed.

As a side, the calculation for the speed and the update for the LCD is done in the main().

- **Schematics**

  Note that MSP432 supplies both the 5 volts and 3.3 volts. However, the output pins can only generate 3.3 volts, so a logic level converter has to be used. In our schematics, the green lines show the 5 volts connection, and the yellow lines show the 3.3 volts. Also note that the echo line splits into two to connect to P1.0 and P2.2, one for rising edge and the other for falling edge.




MSP432
5 VOLTS
Level Converter
HC-SR04
P1.0
P3.6
VCC
TRIG
ECHO
GND
P2.2
3.3V
LO
HI
GND

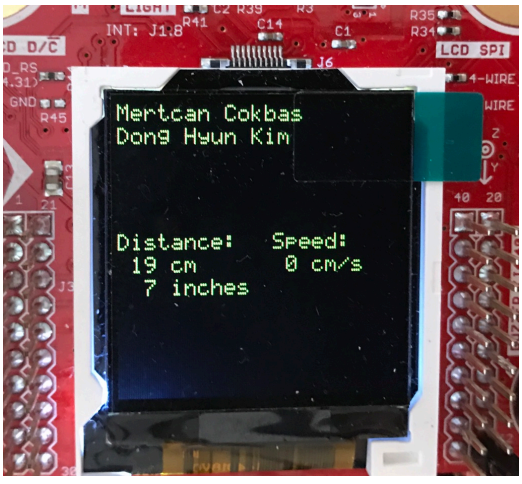
- **Assessment of Success**

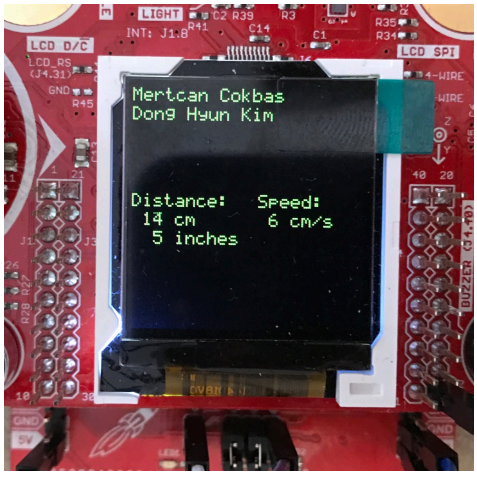
  In terms of our implementation and connection of the MSP432 to the ultrasonic chip, we were mostly successful. From soldering the level converter to using alternate peripherals, we overcame numerous obstacles along the way to make the reading on the LCD work. The hardest challenge of all was reaching the conclusion that our hardware was malfunctioning; although we were certain of our implementation to have 10 readings per second, the ultrasonic chip we had originally was giving readings every 5 seconds, which was largely frustrating. After we tried four different implementations to make the readings more frequently, we still failed and realized that our hardware might actually be problematic. With little hope, we ordered a new ultra sound chip, and almost like a miracle, everything worked as planned.

- **Next steps you might take to make it more successful**

  To improve the project, we should have a more powerful ultrasonic chip that can measure the distance more frequently and accurately. The reading of the speed tends to be shaky after 3 meters, and there should be an algorithm to smooth out the change in speed. We actually tried a few algorithms such as the exponential averaging, but it had almost no effect. Currently, the speed is 'jumping' too quickly.

- **Code and listing for the project**





- **Summary of team members' contributions to the project**

  Throughout the project, Mertcan was responsible for trying out different methods of implementation in terms of code and connecting the components. In the meanwhile, Brian managed the hardware components, helped with debugging the code, and documented the progress from start to finish.