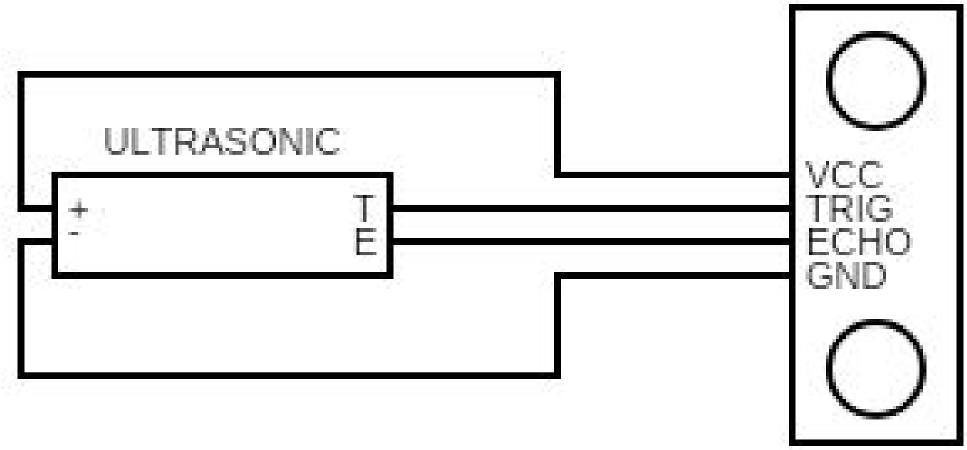
Raymond Au
916672216
Judasilfarion

In Assignment 2 of CSC615 Unix Programming Fall 2020, we wired an ultrasonic sensor to our Raspberry Pi and wrote code for it to measure distance using ultrasonic echolocation. When I started the assignment I was initially rather confused on how to set up the wiring. All of the tutorials online were done without the SB motor shield, but since I'm a hardware manager I knew I had to use the ultrasonic interface on my version of the Raspberry Pi. I tried setting up the ultrasonic sensor using the BCM and resistors and connecting them to the ultrasonic interface, but none of my code was working so I always just assumed that my wiring was wrong. This is one issue that I repeatedly ran into.

Writing the code was really easy. There were many examples of code for the ultrasonic sensor online, I just had to translate it from Python to C. This was quite simple, since all I had to do was look at the C and wiringPi libraries and look for functions that were direct parallels to the ones in the Python version of the code. If no direct parallel existed I'd just create my own. Unfortunately, while my code in theory should've been perfect, for some reason it wasn't working.

Fortunately, I was able to solve these two issues that were preventing my program from functioning correctly. Reading the manual for the SB motor shield taught me that I didn't need to use the BCM at all, I could just wire the ultrasonic sensor directly to the ultrasonic interface on my Raspberry Pi. That solved the physical wiring aspect of my problem. The second issue was solved by asking on Slack for clarification on how the pins were supposed to be set up. I assumed that the manual told me that the pins were supposed to be for the wiringPi library, but in actuality they were the physical pins. This solved the coding aspect of my problem. After that was done, my ultrasonic sensor was working fine.

Hardware Diagram:


ULTRASONIC
+
-
T
E
VCC
TRIG
ECHO
GND