

mbed Lab 4: Do You Hear What I Hear?

CS 220

Nov. 13

Introduction

For this lab, you'll use your LCD display with an HC-SR04 Ultrasonic Ranging Module (HCSR04) to build a distance measuring tool. Ultrasonic sensors similar to this one are often used for the parking sensors which are offered as options on new cars today. These sensors are easy to spot — four quarter-sized disks embedded within the front or rear bumper of a car. To give you a bit of experience with writing a device driver, you'll write a simple device driver for the HCSR04, working from the specifications given in the device's data sheet. (Yes, hypothetically you could just use the mbed's HCSR04 library, but anyone experimenting with embedded systems should write at least one device driver themselves. Now's the time.)

Lab Objectives

1. Gain more confidence in reading data sheets, and designing and wiring circuits.
2. Write a simple device driver for an HCSR04.
3. Build a complete prototype embedded system for measuring distances.

Lab Procedure

Read each of the following steps **completely** before acting on them.

1. You won't need to apply power to the KL25Z until later, so set it and the USB cable aside for now.
2. In addition to the HCSR04, you'll be working with an SN74LVC244A ('244A) in this lab. Why the '244A? Simple — the KL25Z uses 3.3V logic. The HCSR04 uses 5V logic. The trigger signal from the KL25Z to the HCSR04 can be read by the HCSR04 as is. The echo signal from the HCSR04 back to the KL25Z must be level-translated from 5V logic levels to 3.3V logic levels to avoid burning-out the KL25Z. The '244A is used to accomplish the level translation.
3. Using information available on the HCSR04's data sheet and pp. 1, 3, and 5 of the '244A's data sheet, complete the wiring diagram (separate sheet). The data sheets are available on the course web site.
4. Have me check your diagram once you've completed it.

5. Keep the GND and 3.3V bus strip wires from the KL25Z to the breadboard in place. Also keep the LCD display wiring in place.

Remove the temperature sensor circuitry from the breadboard. Place the TMP36, capacitors, and resistor into the anti-static bag.

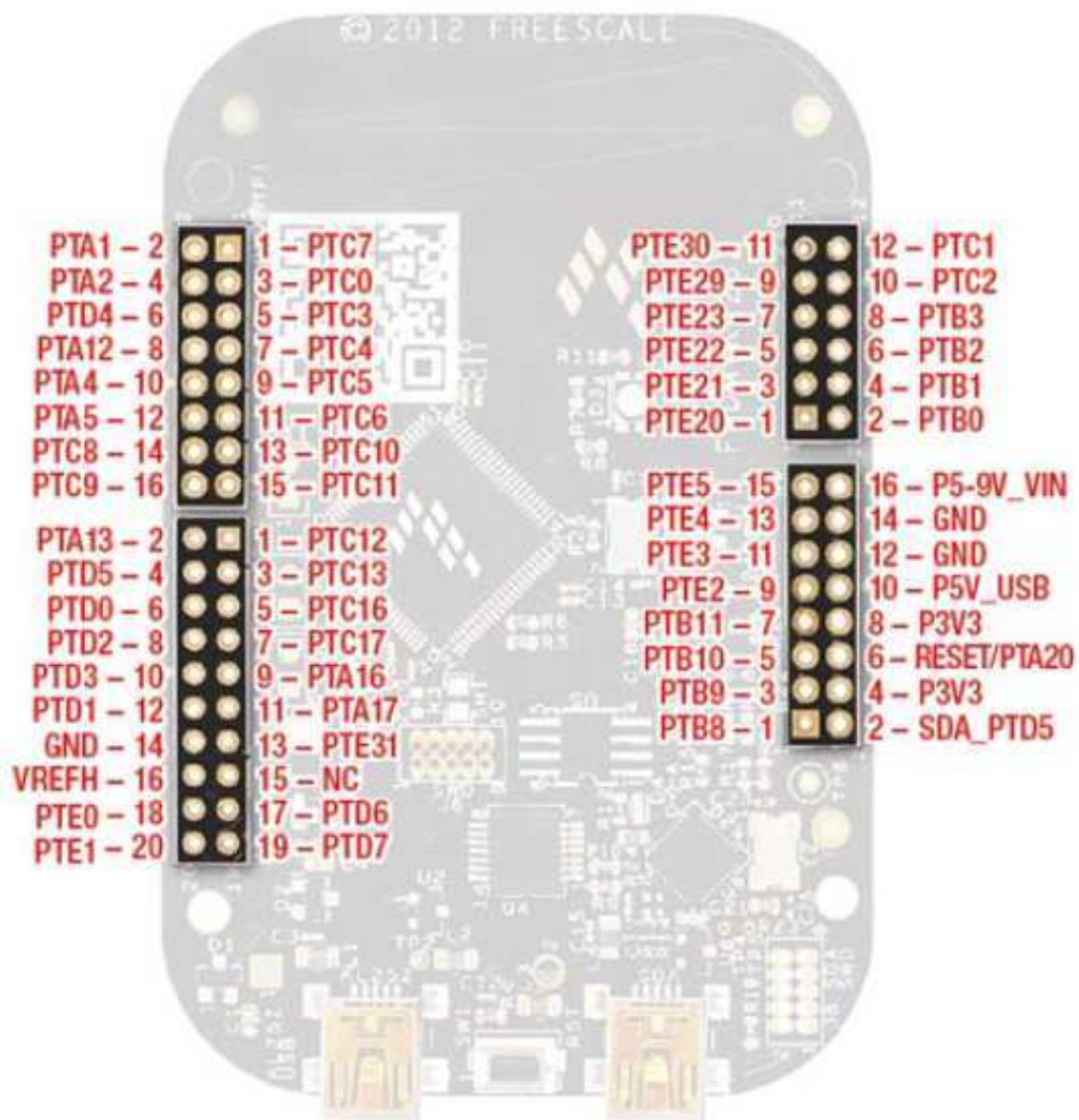
6. You'll find the '244A in the anti-static bag. Insert it into the breadboard with the U-shaped indentation facing toward row 1 of the breadboard. (This is an engineering convention and will make it easier for me to check your wiring.) The chip should straddle the channel running down the center of the breadboard. (Refer to the photos on the course home page if this isn't clear.)
7. Insert the HCSR04 into its own set of rows on the breadboard.
8. Wire your circuit. If you're still a bit unsure about your wiring ability, have me check your work.
9. Go to the mbed development site. Clone your LCD/temperature sensor project and name it "Rangefinder" or some name of your own choosing. Remove the temperature sensor code.
10. Now it's time to think about your device driver function. Here's the basic flow in pseudo-code form:

```
float measureDistance()
{
    create a timer; // see the mbed's Timer interface
    reset the timer;
    make sure the HCSR04 is idle;
    send a trigger pulse;
    wait for the echo signal to go high;
    start the timer;
    wait for the echo signal to go low;
    stop the timer;
    read the timer value in microseconds;
    if the timer value exceeds the HCSR04's maximum range delay
        return -1.0;
    else
        return the distance in inches;
}
```

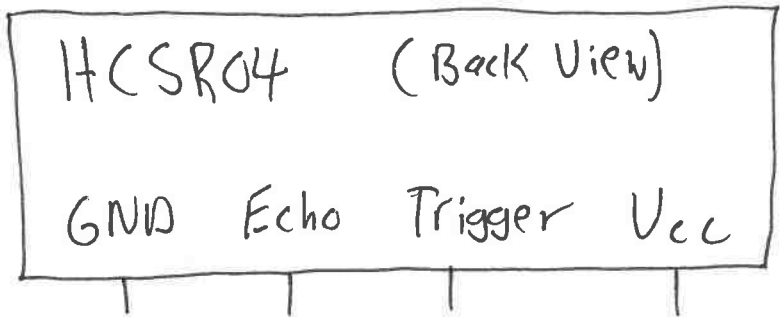
Refer to the HCSR04 data sheet for timing details.

11. Implement your device driver. You'll need to use the mbed's `DigitalOut` (trigger; PTB9) and `DigitalIn` (echo; PTB8) interfaces in the driver.
12. Write a main program that prints the measured distance in inches once per second, to two digits after the decimal point.
13. Connect the KL25Z to your host computer. Download and test your design.
14. Measure some distances!!!

KL25Z Pin Diagram



USB connector-end of the KL25Z



Wiring
Diagram

