UNIVERSITY OF JORDAN

Embedded Systems Lab

Ultrasonic Range Finder with Buzzer

Preferred Group Size	Grading	Project Due Date
(2) Two is the allowed group size. The students in one group MUST be from the same class	**(15)** 15 marks is the project weight	**(18th)** The deadline for the project is the week starting Sunday, December 18th 2016

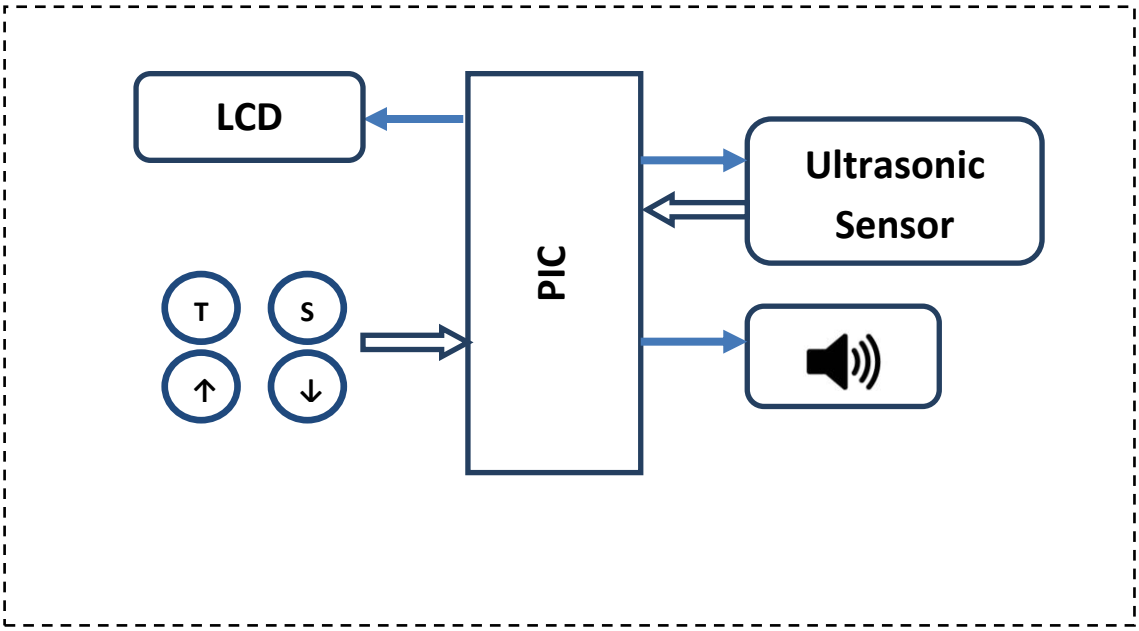
Project Description:

In this project you are required to implement a Range Finder using an ultrasonic sensor. The device should detect objects that are in the range of "2 cm" and up to "400 cm" from the sensor. When an object is detected, the distance from the sensor should be displayed on an LCD in centimeters. The user of the device should have the ability to save at least **two** threshold ranges (e.g. 25–50 cm, 51-100 cm).

When an object is detected within one of the threshold ranges, a buzzer should make a certain sound. The sound for each range should be different and it should reflect the closeness of the object (i.e. closer threshold ranges should have sounds with higher frequencies). Keep in mind that human ear is more sensitive to sound frequencies between "1 KHz" and "4 KHz".

Hardware Layout

This project requires one PIC microcontroller attached to different input and output devices as shown in the following figure:



The PIC is connected to four push buttons (T, S, ↑, and ↓), LCD, ultrasonic sensor, and a buzzer. It is recommended that you use **HC-SR04** ultrasonic sensor which has the same required range and can be easily interfaced with the microcontroller without the need for Analog-to-Digital conversion.

Functionality

In this section, we describe the operation process of the Range Finder. First, the ultrasonic sensor should **continuously** search for close objects that are within the detectable range (i.e. 2 cm to 400 cm). When an object is detected, the PIC should compute the distance of the object from the device assuming the speed of sound is 340 m/s. The computed distance should be displayed on the LCD in centimeters.

The user can setup two threshold ranges and store them in the microcontroller. In order to do that, the user should press the "T" button in order to enter the setup menu. The setup menu should appear on the

LCD and instruct the user to enter the minimum and maximum limit of each threshold range. The user can use the increase "↑" and decrease "↓" buttons to determine the values of limits. After each limit is determined, the user should press the "S" button to save the current limit and move to the next. At any time, the user can press the "T" button to exit the setup menu. When the device is in the setup mode, the ultrasonic sensor is deactivated and no objects will be detected.

After the threshold ranges are set, if an object is detected in any of these ranges the buzzer should go off (i.e. should make a sound) **as long as the object is in the threshold range**. Each threshold range must cause a unique buzzer sound that reflects the closeness of the object.

You can control you ultrasonic sensor/buzzer manually with timers or using the PWM module. The datasheet of the microcontroller contains all the information that you need about timers and PWM modules.

Every group should add an extra feature to the device. For example, one group can add a mechanism to check if the entered threshold ranges are wrong or out of range. Another group can add an RGB LED and use different LED colors for objects detected in different threshold ranges. The latter feature can be useful for users with hearing problems.

Important Notes

- Start as early as possible on your project, though the project description sounds simple, there is inherent complexity in both hardware and software aspects, so do not underestimate the time it needs, you will have many problems along the way which you will have to resolve!
- Never think of buying a model or commissioning someone to do it for you, not only will you get a zero in the project, but also your act will be considered as a direct violation to JU laws and your actions shall be reported as cheating in the final exam!
- Code sharing between groups is not allowed!
- If you acquire a ***part*** of your software from a book, website, etc … kindly reference it properly, else it will be considered as plagiarism.
- You are only allowed to base your project on PIC16F877A or PIC16F84A.
- All programming must be done in **PIC ASSEMBLY** language only; using high level languages in the project will get you a Zero.
- Your submitted work must be professional:
    1. Hardware: your implementation should be neat and easy to trace.
    2. Software: your work should be fully documented, all inputs/outputs should be listed, and each subroutine/macro should be fully documented! Use functional comments! Refer to the last section in experiment 3 regarding documentation.
- Students are not allowed to move between groups once they are formed, so choose your group carefully from the beginning! ***We are not responsible if your colleague in the group chose to drop the class, we will not allow you to join another group!***
- Divide the work such that each student is responsible for a specific task, **YET EVERY** student is required to answer for **ANY QUESTIONS** in relation to any submitted work of the project.

Good Luck and Have Fun Building the Project