The Bat Machine

3D Ultrasonic “Radar” Scanner and Signal Visualizer
Design Document, CSEE 4840 Embedded Systems

Nico Bykhovsky (nb3227), Lourdes Sanchez-Medina (als2408)
Spring 2025

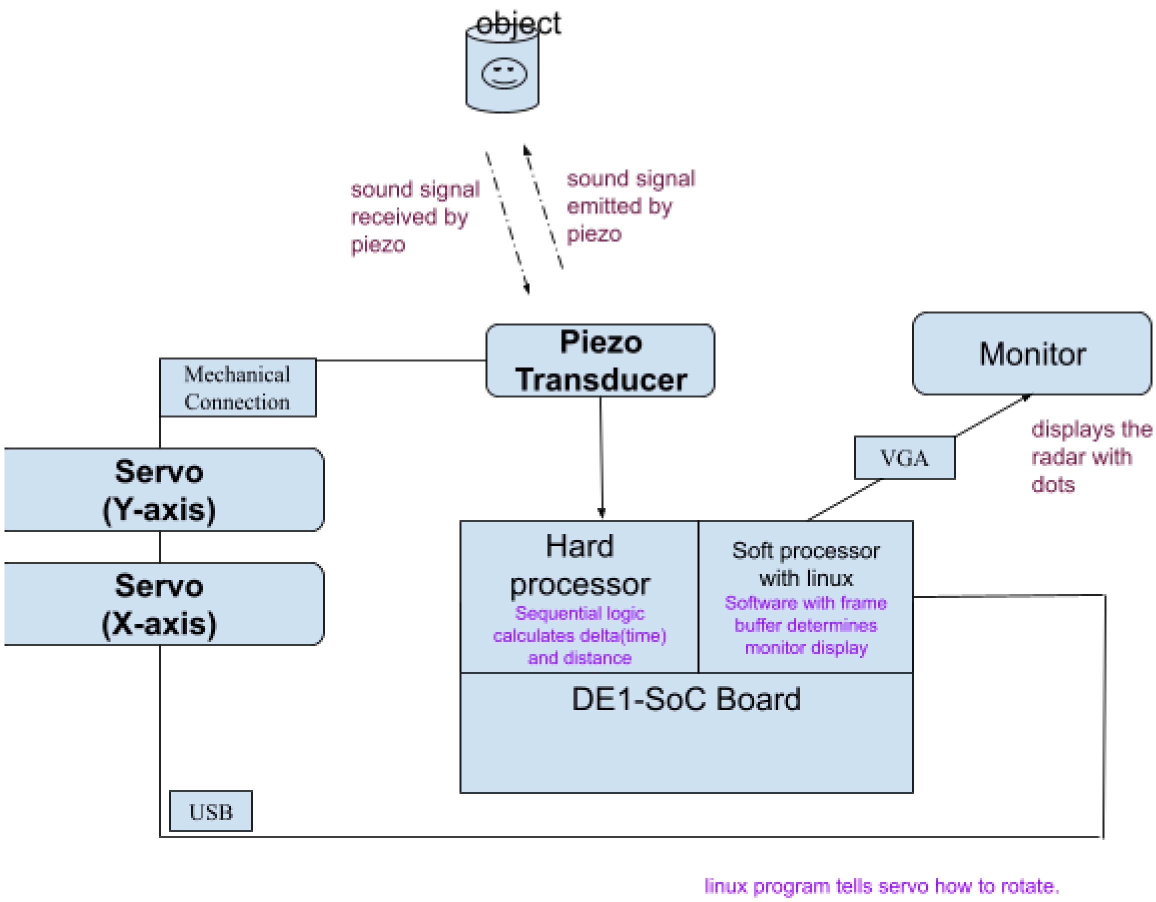
Contents

Introduction

This project uses a sonar sensor controlled by the FPGA to scan its surroundings from a given point and display a real-time “radar screen” of nearby objects. The FPGA will trigger the ultrasonic ‘ping’ and the sonar sensor measures the echo time to compute the distance between itself and the nearby object. This will be done via a Piezo Transducer. By rotating the sensor on a servo (or possibly by using multiple sensors positioned at different angles), the system can sweep across an area, varying both the direction it's facing and the height of the plane it’s sweeping. The result can be shown on a circular radar-style graph on a monitor connected to the FPGA–objects appear as dots on top of concentric circles at various ranges. The goal here is to build a Radar sensor and track the distances of objects on a video interface.

We took inspiration for this project from here github.com. A basic setup on an FPGA showed this concept by scanning left/center/right with an ultrasonic sensor and plotting the detected objects on a VGA screen, playing a tone via speaker based on distance.

System Block Diagram



Algorithms & Hardware-Software Interface

Hardware: Our project utilizes the FPGA and System Verilog to measure time using clock cycles and sequential logic and the control of sensor operation. First, we generate a trigger pulse to the piezo transducer, so it emits an ultrasonic signal. From here, we start the timer. Once the piezo transducer senses the echo of the sound signal (as in, its return), the FPGA receives a wire signal UP and the 'timer' is stopped: the number of clock cycles are counted between when the transducer emitted and received the sound signal. This number is cut in half, then multiplied by the speed of sound to get the distance. We will also explore fast fourier transform by taking an off the shelf implementation for the hard processor to extract the correct frequency from the Piezo Transducer on its return signal. This delay feeds into a simple function to find the distance.

From here, the distance number will be sent to the soft processor, which runs a multithreaded C program in Linux that, similarly to lab 2, has a frame buffer (similar to lab 2) that one thread uses to display a radar, use the distance data from the hard processor to know where to draw a point within the frame buffer, and display it. Additionally, another will signal the servo to move and keep track of its the piezo transducer's position; the program will also use this information to determine the position of the blip on the screen.

For a 3D element, the monitor screen will update every 1 second to show different vertical planes. The program controls 2 servos–one for vertical scanning, and one for horizontal scanning.

Resource Budgets

Low memory is needed–we need the SD card to run a C program in Linux and display the radar. We will need 1 piezo transducer and 2 servos (one of us has 2 servos).