A picture containing graphical user interface

Description automatically generated

ECE 425

Microprocessor Systems

Final Project

Sonar System

Suren Bursalyan

Sherifat Solebo

Instructor: Aaron Nanas

Spring 2025

1. **Introduction**

The aim of this project is to build and implement a system to detect nearby objects using ultrasonic waves. The information from all the peripherals (i.e., distance and angle) is then transmitted from a microcontroller to a graphic that displays the real-time position of an object in relation to the system - given that an object is inside the operating range.

1. **Background & Methodology**

Sonar (Sound Navigation and Ranging), as opposed to Radar (Radio Detection and Ranging), uses sound waves to detect objects whereas Radar uses radio waves to achieve the same thing. This distinction is important to make since this project implements an ultrasonic sensor to detect objects, hence why it is referred to as sonar.

The ultrasonic sensor will be mounted on a servo motor which will allow the sensor to rotate back and forth by 90 degrees in each direction, resulting in a 180-degree sweep.

The Sonar System described in this report requires an Ultrasonic Sensor, Servo Motor, a 3.3V/5V breadboard power supply, and the TM4C123GH6PM Microcontroller. Furthermore, Keil µVision5 and Processing IDE are used for all of the required software.

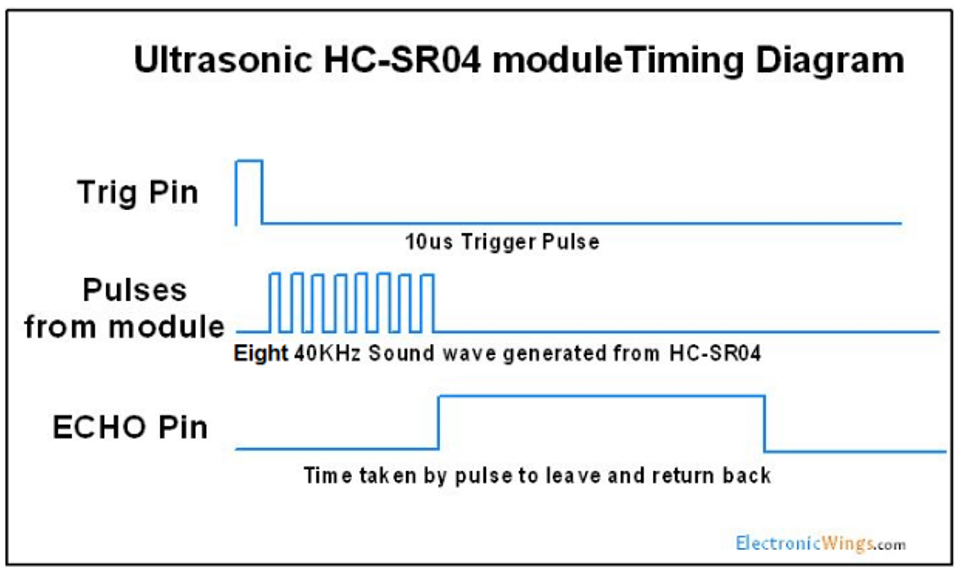
HC-SR04 Ultrasonic Sensor Operation:

For this specific sonar system, the HC-SR04 ultrasonic sensor will be used. The HC-SR04 ultrasonic sensor has an operating range of 2cm – 400cm, and an operating voltage of 5V. It contains a transmitter, receiver, oscillator, and four pins. Namely, VCC, GND, TRIG, and ECHO.

The transmitter works in conjunction with the oscillator and TRIG pin to produce the ultrasonic waves. It contains a piezoelectric disc that is chemically and physically shaped to have a resonant frequency of 40kHz, which is the operating frequency for the HC-SR04. To begin ranging, the TRIG pin must receive a TTL pulse of at least 10µs. The transmitter then utilizes the on-board oscillator to vibrate the piezoelectric disc and in turn it generates 8 ultrasonic waves.

The receiver works with the ECHO pin to capture the ultrasonic waves that bounce back from an object (if applicable). It essentially works like the transmitter in reverse. When the returning ultrasonic waves hit the piezoelectric disc in the receiver, it causes the material to vibrate, generating a voltage as a result. This drives the ECHO pin high and is used as an input to the microcontroller. The pulse width of the ECHO pin can then be used to calculate the distance of an object.

A timing diagram that demonstrates this can be seen in *Figure 1*.



**Figure 1 – HC-SR04 Timing Diagram**

SG-90 Servo Motor Operation:

The SG-90 servo motor in this project will be utilized to sweep the ultrasonic sensor back and forth to detect objects 90 degrees in either direction of the sonar system. The SG-90 has only 3 inputs: VCC, GND, and a signal (or PWM) input.

The servo operates on a 50 Hz PWM signal, which has a period of 20ms. To rotate the servo, the pulse width (or duty cycle) of the signal must be changed. A pulse width of 1ms is required to rotate the servo all the way to the left. A pulse width of 1.5ms is required to rotate the servo to the center position. Finally, a pulse width of 2ms is required to rotate the servo all the way to the right.

In order to create a sweeping effect from the servo, this pulse width must be constantly updated in software and once the servo is all the way to one side the direction must be reversed.

*Figure 2* shows the duty cycles and the resulting position of the servo motor when these duty cycles are provided to it.

A diagram of a graph

AI-generated content may be incorrect.

**Figure 2 – Required Duty Cycles to Rotate Servo**

This specific implementation of Sonar requires the use of 6 key embedded systems concepts outlined below:

* UART
* SysTick
* General Purpose Input-Output (GPIO)
* Pulse Width Modulation (PWM)
* Polling
* Real-Time Systems

1. **Block Diagram**

A block diagram of the sonar system is shown in *Figure 3*. A table describing the inputs and outputs of each component is also provided in *Table 1*.

A diagram of a computer system

AI-generated content may be incorrect.

**Figure 3 – Block Diagram of Sonar System.**

**Table 1: Description of Inputs and Outputs.**

|  |  |  |
| --- | --- | --- |
| **Signal** | **Description/Purpose** | **Input/Output**\* |
| TRIG | Input to the ultrasonic sensor. Microcontroller provides 10µs pulse to initiate ranging. | Output |
| ECHO | Output from the ultrasonic sensor to the MCU. Provides the MCU with a pulse that can be used to calculate distance of a detected object. | Input |
| PWM | Provides the servo motor with a PWM signal to rotate and change direction of servo motor. | Output |
| UART | Transmits current distance of detected object and current angle of servo motor to computer. | Output\*\* |
| 5V | Provides the necessary 5V supply to both the ultrasonic sensor and servo motor. | N/A |

\* Input/Output is from the perspective of the microcontroller.

\*\* Shown as bidirectional signal in block diagram because code is uploaded from the computer to the microcontroller.

1. **Components Used**

**Table 2: List of all Components Used for Project.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Model** | **Manufacturer** | **Purpose** |
| Ultrasonic Sensor | HC-SR04 | N/A (Generic) | Object detection |
| Servo Motor | SG-90 | Tower Pro | Sweep ultrasonic sensor back and forth. |
| Power Supply | 3.3V/5V Breadboard Power Supply | MakerSpot | Provide 5V to servo and ultrasonic sensor |
| Microcontroller | TM4C123GH6PM | Texas Instruments | Control I/Os |

1. **Pinout Used**

The table below lists all of the pins used for this project and their purposes:

**Table 3: Pinouts Used for Sonar System**

|  |  |  |
| --- | --- | --- |
|  | **Microcontroller** |  |
| **Pin Designation** | **Input/Output** | **Connected To** |
| **PA2** | Output | Ultrasonic Sensor (TRIG) |
| **PA3** | Input | Ultrasonic Sensor (ECHO) |
| **PF2** | Output | Servo Motor (PWM) |
|  | **Power Supply** |  |
| **Pin Designation** | **Connected To** |  |
| **VCC** | Servo Motor |  |
| **VCC** | Ultrasonic Sensor |  |
| **GND** | Servo Motor  (Shared with MCU) |  |
| **GND** | Ultrasonic Sensor  (Shared with MCU) |  |

1. **Analysis & Results**

Successful implementation of the sonar system graphic will have the outputs shown in *Figure 4* and the servo will sweep the sensor back and forth as shown in the demo video linked below:

* [Sonar System Project](https://youtube.com/shorts/GWr8HmAAaNE?feature=share)

A green radar screen with a black background

AI-generated content may be incorrect.A green and red radar screen

AI-generated content may be incorrect.

**Figure 4 – Clean Sweep on Sonar Graphic (Left) Vs. Object Being Detected (Right)**

As is shown, when an object is within the operating range of the system, the graphic displays an “Object: In Range” message and provides the user with the current angle of the servo along with the distance of the detected object.

Implementation Issues:

During implementation of this sonar system, there were a number of bugs and other implementation issues with the majority of them being software. These included issues moving the servo motor, making functions that take an angle as input rather than number of ticks to determine the current duty cycle and therefore rotate the servo, and issues getting the ultrasonic sensor to work with the servo motor.

Servo Issues:

The first obstacle for this sonar system was establishing communication with the servo. During the first attempt, rather than sending a PWM signal using the dedicated PWM pins on the TM4c123GH6PM Microcontroller, the signal was manually generated via a GPIO pin and a PWM signal was simulated by driving a pin high and low at the required frequency. This ended up over-complicating the project, so PWM was later implemented.

When a proper PWM signal was initially implemented, the function that generated the PWM signal took ticks of the clock as an argument to determine the current duty cycle being sent to the servo. Later, the function was revised to take the desired angle of the servo as an argument, which helped with clarity of the code and made it easier to go back and update movement of the servo as needed throughout the process.

Ultrasonic Sensor Issues:

Once the servo motor was working as intended, work began on the ultrasonic sensor. At first, there were issues getting the sensor to work with the servo motor. Every time the function to read distance was called in the main file, the servo would cease to rotate. This made it very clear that the issue had to reside within the “read distance” function. It was soon realized that the code was getting stuck in one of the while loops inside the function.

The issue was that when there was no object in front of the sensor to detect, the program would infinitely wait for the ECHO signal to go high, which would cause much of the processing power within the microcontroller to be used for detecting an object that did not exist. This was remedied by implementing a time limit in the while loop, which (if reached) will return a distance value that is out of range for the sonar system. Once this time limit was added, the system began to operate as intended.

1. **Conclusion**

As a result of constructing this sonar system, a deeper understanding of the embedded systems concepts highlighted in this report was obtained. Furthermore, the background physics of how ultrasonic sensors are designed, made, and operated was learned, providing for a clearer view of how everything in this sonar system works. A link to the GitHub repository for this project is provided below.  
  
[GitHub Repository](https://github.com/surenb1/Sonar-Project.git)