# Proximity Alarm System
Abdalrahman Zbeidi

Introduction (Project Description)

The idea of this project is to use the sonar system that detects how close something is to the sonar system acting like a security camera. When something gets too close to the sonar it will detect it and send the message to the buzzer to beep detecting someone or something is close. The goal is to create a simple yet effective system that can be utilized in the real world which can be very practical since it can be used as a security and a safety feature. The project contains the seg7 digit display for the user to realize how far/close they are. The system will be used with both C and assembly coding language and will be used in the CSS Ti software to write and upload the code in.

Purpose   To detect nearby objects using sonar and alert users with visual and sound signals for use in security or safety scenarios.

Inputs   -Sonar sensor (distance input)

Outputs   - Grove buzzer (alarm output)
- 7-segment display (distance output)

Functions   - Continuously monitor distance from the sonar sensor
- Display live distance data on the 7-segment display
- Trigger buzzer if threshold is breached

Performance   - Real-time response (delay < 1 second)
- Accurate distance detection within ±2 cm
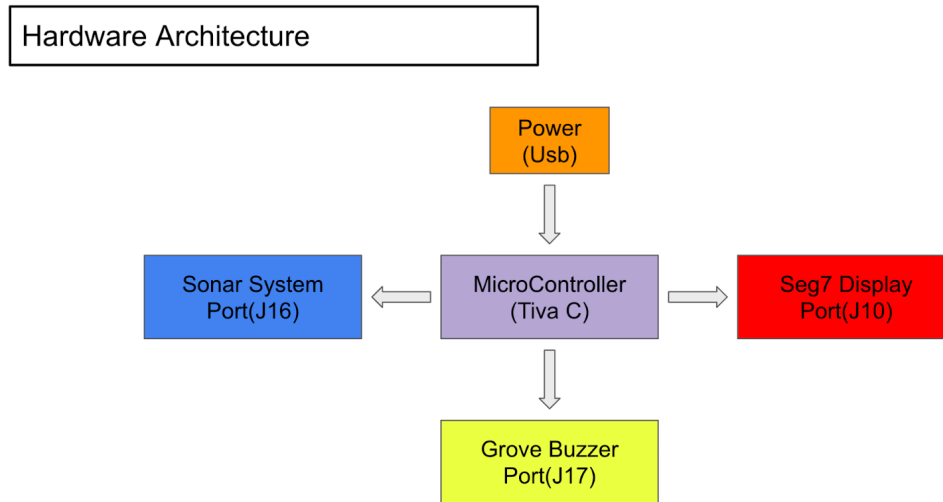- Audible buzzer alert when object is within range

2. Architecture

a. System Architecture
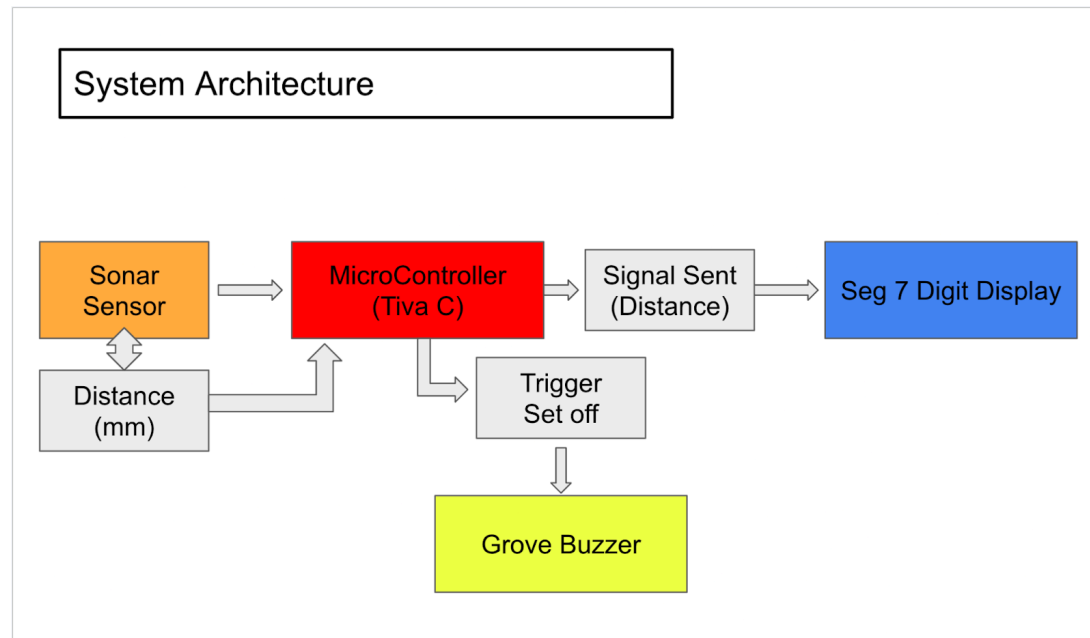
Block Description

- **Sonar Sensor**: Detects the distance betIen the sensor and an approaching object. Sends this raw distance data in millimeters to the microcontroller.

- **Microcontroller (Tiva C)**: Serves as the control unit of the system. It reads the sonar sensor input, processes the data, determines if a distance limit is close or reached, and outputs signals to both the display and the buzzer.

- **Seg 7 Digit Display**: Receives the processed distance data from the microcontroller and displays the live distance reading, giving users real-time visual feedback.

- **Grove Buzzer**: Acts as the alert system. When the microcontroller detects that an object is too close which I set it to a certain distance, it sends a trigger to the buzzer to create the sound.

b. Hardware Architecture

```
┌─────────────────────────────────┐
│ Hardware Architecture           │
└─────────────────────────────────┘
```

```
                        ┌──────────────┐
                        │    Power     │
                        │    (Usb)     │
                        └──────────────┘
                               │
                               ▼
┌──────────────┐       ┌──────────────┐       ┌──────────────┐
│ Sonar System │  ⟸    │MicroController│  ⟹   │ Seg7 Display │
│  Port(J16)   │       │   (Tiva C)   │       │  Port(J10)   │
└──────────────┘       └──────────────┘       └──────────────┘
                               │
                               ▼
                        ┌──────────────┐
                        │ Grove Buzzer │
                        │  Port(J17)   │
                        └──────────────┘
```

Block Descriptions:

- **Sonar Sensor**: Connected via GPIO pins on J6 pin, it sends distance data using trigger and echo pulses.

- **Tiva C Microcontroller**: Central MicroController for reading inputs and controlling outputs. It receives sonar data, processes it, and drives both the display and the buzzer.

- **Grove Buzzer**: Connected to the Tiva C to generate sound alerts when a nearby object is detected.

- **7-Segment Display**: Connected to Port(J10) to send signals and to display information.

- **PoIr (Usb)**: The system receives poIr and programming through the USB port and is connected to the Tiva C LaunchPad.

c. Software Architecture
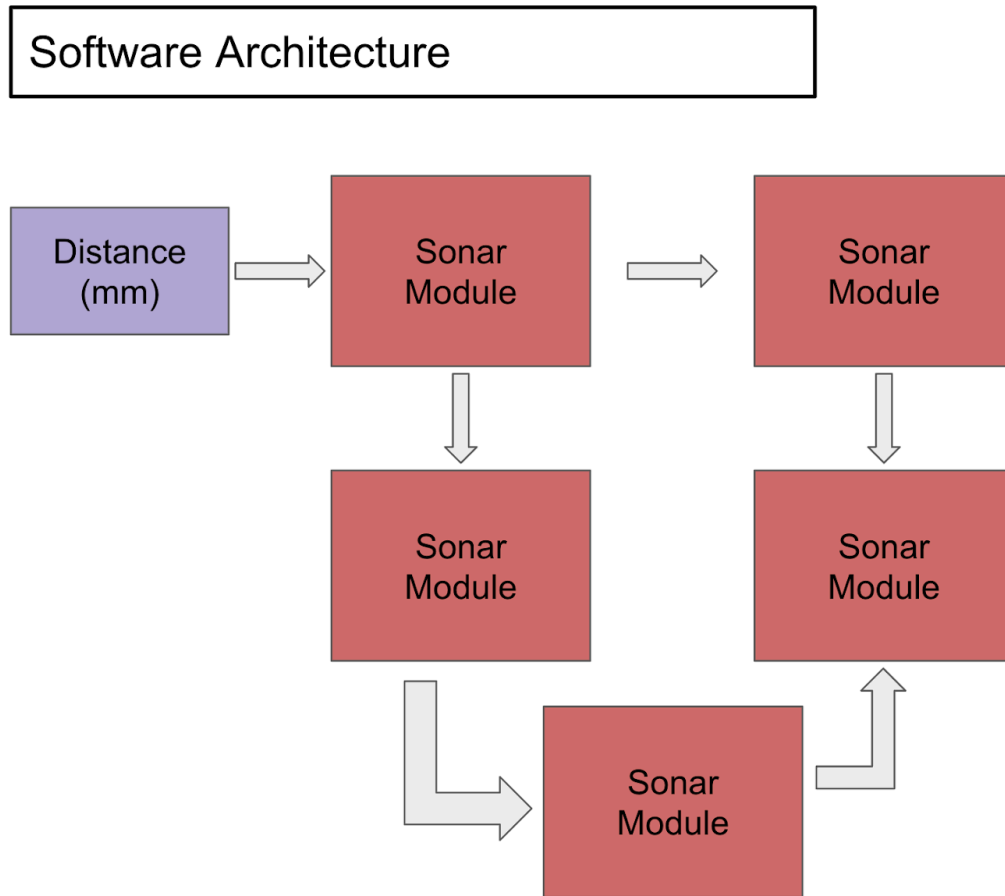
Block Descriptions:

- **Sonar Module**: Triggers the sonar sensor and calculates distance based on echo.

- **Distance Evaluation**: Compares measured distance to a threshold.

- **Display Output**: Sends the distance to the seg 7 display.

- **Buzzer Control**: Activates the buzzer if distance is close or at the target.

- **Main Timer Loop**: Handles repeated cycles and updates.

Summary of Project Status

While working on the project, I encountered a minor issue where the buzzer would activate slightly above the intended threshold. HoIver, the margin of error was small, and I determined it was likely due to hardware limitations, specifically with the ultrasonic sensor accuracy of measurements. Lastly, the project used 19,065 bytes (7%) of the available Flash memory for instructions and 6,116 bytes (18%) of the available SRAM for data storage. Overall, it took around 6–7 hours of focused work to complete and prepare the project.