



Standard Embedded Diploma - Mini Project 4

Project Title:

Car Parking Sensor Using ATmega32, Ultrasonic Sensor, LCD Display, LEDs, and Buzzer.

Objective:

The aim of this project is to design a simple car parking sensor system that helps drivers detect obstacles and avoid collisions while parking. The system measures the distance between the car and an obstacle using an ultrasonic sensor and displays the distance on an LCD screen. LEDs and a buzzer provide visual and auditory warnings based on the proximity of the object.

Overview:

This project uses an ATmega32 microcontroller to process distance data from an ultrasonic sensor and display the results on an LCD. The system includes three LEDs (Red, Green, and Blue) and a buzzer for indicating the proximity of an obstacle. The LEDs light up progressively and the buzzer activates as the obstacle gets closer, helping drivers park safely.

Features:

- **Ultrasonic Distance Measurement:** The ultrasonic sensor measures the distance between the car and any nearby object.
- **LCD Display:** Displays the distance measured by the ultrasonic sensor in centimeters. Also displays "Stop" if the distance is less than 5 cm.
- **LED Indicators:** Three LEDs (Red, Green, and Blue) light up according to the proximity of the object.
- **Buzzer Alert:** The buzzer sounds when the object is too close (≤ 5 cm).
- **Dynamic LED Behavior:** LEDs turn ON/OFF based on the following distance ranges.

Hardware Components:

- ATmega32 Microcontroller
- HC-SR04 Ultrasonic Sensor
- 16x2 LCD Display
- Red, Green, and Blue LEDs
- Buzzer

Pin Connections:

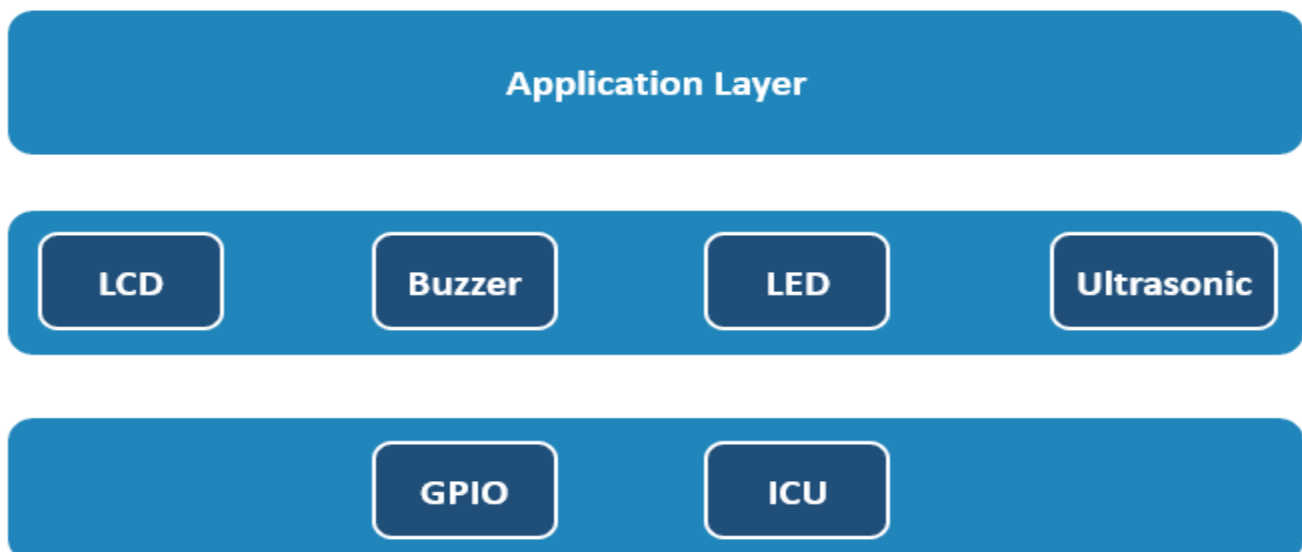
- **ATmega32:**
 - **Port A (PA1):** Connected to LCD RS Pin
 - **Port A (PA2):** Connected to LCD E Pin
 - **Port A (PA3 to PA6):** Connected to LCD pins for data (D4-D7).
 - **Port B (PC0 to PC2):** Connected to Red, Green, and Blue LEDs respectively.
 - **Port C (PC5):** Connected to the transistor controlling the buzzer.
 - **Port D (PD6):** Echo pin for Ultrasonic sensor.
 - **Port D (PD7):** Trigger pin for Ultrasonic sensor.
- **Ultrasonic Sensor:**
 - **Echo:** PD6.
 - **Trigger:** PD7.
- **LCD Display:**
 - 4 bits mode.
 - **RS:** PA1.
 - **RW:** Ground.
 - **E:** PA2.
 - **D4, D5, D6, D7:** PA3, PA4, PA5 and PA6.
- **LEDs:**
 - **Red LED:** PC0.
 - **Green LED:** PC1.
 - **Blue LED:** PC2.
- **Buzzer:**
 - Connected to PC5.

Operation Description:

- The ultrasonic sensor emits an ultrasonic wave and listens for its reflection from a nearby object.
- The ATmega32 microcontroller calculates the distance by measuring the time taken for the wave to return to the sensor.
- The measured distance is displayed on the LCD in centimeters.
- The system continuously updates and recalculates the distance, providing real-time feedback to the driver.
- **Distance to LED and Buzzer Mapping:**
 - **Distance \leq 5 cm:** All LEDs are flashing (Red, Green, Blue), Buzzer sounds, LCD shows "Stop."
 - **6 cm \leq Distance \leq 10 cm:** All LEDs ON (Red, Green, Blue), No buzzer.
 - **11 cm \leq Distance \leq 15 cm:** Red and Green LEDs ON, Blue LED OFF.
 - **16 cm \leq Distance \leq 20 cm:** Only Red LED ON, others OFF.
 - **Distance $>$ 20 cm:** All LEDs OFF, Buzzer OFF.

System Requirements:

- **Microcontroller:** ATmega32.
- **System Frequency:** 16 MHz
- **The Project should be implemented using the below layered model architecture**



Drivers Requirements:

GPIO Driver Requirements:

- Use the same GPIO driver implemented in the course.

ICU Driver Requirements:

- Use the same ICU driver implemented in the course.
- The ICU should be configured with a frequency of **F_CPU/8** and set to detect the **rising edge** as the first edge.
- ICU_init and ICU_setCallBack functions should be called inside the **Ultrasonic_init** function.

Buzzer Driver Requirements

- The buzzer will serve as an alert system in case of the distance is less than or equal to 5 cm.
- **Buzzer Driver Functions:**
 - **void Buzzer_init(void)**
Initializes the buzzer pin direction and turn off the buzzer.
 - **void Buzzer_on(void)**
Activates the buzzer.
 - **void Buzzer_off(void)**
Deactivates the buzzer.
- **Additional configurations:**
 - The buzzer pin should be configurable to be connected to any microcontroller ports and pins.

LCD Driver Requirements:

- Use the same LCD driver implemented in the course with 4-bit data mode. The LCD is used to display the distance in centimeters and If the distance is less than 5 cm, the LCD should display the word "**Stop**" as a warning.⁴
- **Display:** 16x2 LCD in 4-bit mode
- **Pin Configuration:**
 - **RS:** PA1
 - **RW:** Ground.
 - **Enable:** PA2
 - **R/W:** Ground (always in write mode)
 - **D4, D5, D6, D7:** PA3, PA4, PA5 and PA6.

Ultrasonic Driver Requirements:

- Implement a full Ultrasonic driver using the ATmega32 ICU driver.
- The ultrasonic driver must have the following functions:
 - **void Ultrasonic_init(void)**
 - **Description:**
 - Initialize the ICU driver as required.
 - Set up the ICU callback function.
 - Set the direction for the trigger pin as output through the GPIO driver.
 - **Inputs:** None.
 - **Return:** None.
 - **void Ultrasonic_Trigger(void)**
 - **Description:**
 - Send the trigger pulse to the ultrasonic sensor.
 - **Inputs:** None.
 - **Return:** None.
 - **uint16 Ultrasonic_readDistance(void)**
 - **Description:**
 - Send the trigger pulse by using the Ultrasonic_Trigger function.
 - Start the measurement process via the ICU driver.

- **Inputs:** None.
- **Return:** The measured distance in centimeters.
- **void Ultrasonic_edgeProcessing(void)**
 - **Description:**
 - This is the callback function called by the ICU driver.
 - It calculates the high time (pulse time) generated by the ultrasonic sensor.
 - **Inputs:** None.
 - **Return:** None.
- **Additional configurations:**
 - The trigger pin should be configurable to be connected to any microcontroller ports and pins.

Video Reference:

<https://youtu.be/bhtyP1ZBiJg>

Thank You
Edges For Training Team