Embedded Systems Lab (EC382)

Ultrasonic Distance Measurement using PSoC

End Semester Lab Report

Instructor: Dr. Sheikh Wasmir Hussain

Name: D. Charuseelan

Roll No: 2201065

Group No: 11

Date: 08-04-2025

Objective:

The objective of this experiment was to design a simple system using the PSoC to measure distance using an ultrasonic sensor. This was achieved by generating a trigger pulse, measuring the duration of the returned echo pulse, and then converting that duration to distance in both inches and centimetres. The distance results were then displayed on an LCD screen. This lab demonstrates the practical application of ultrasonic sensors, microcontrollers, and peripheral devices to build a real-time distance measurement system.

Background:

Ultrasonic sensors are commonly used for distance measurement, and they work based on the principle of sound wave reflection. An ultrasonic sensor consists of two main components:

- **Transmitter (TRIG pin)**: Emits an ultrasonic pulse.
- **Receiver (ECHO pin)**: Listens for the echo of the pulse after it bounces off an object.

The time it takes for the sound wave to travel to an object and back to the sensor is directly related to the distance of the object. By measuring this time interval, we can calculate the distance to the object using the formula:

$$\text{Distance} = \frac{\text{Time}}{2} \times \text{Speed of Sound}$$

The PSoC microcontroller is used to control the ultrasonic sensor, measure the echo time, and convert this to a readable output. The LCD display is used to output the calculated distance in both inches and centimetres.

Materials and Equipment:

- PSoC Development Board (Cypress)
- Ultrasonic Distance Sensor (e.g., HC-SR04)
- LCD Display (16x2 Character LCD)
- Connecting Wires
- Power Supply (Laptop)

Circuit Design:

The circuit consists of the following main components:

1. **Ultrasonic Sensor (HC-SR04)**:
   - **TRIG Pin**: This pin is connected to a digital output pin on the PSoC and is used to send a pulse to the ultrasonic sensor, triggering it to emit the ultrasonic wave.
   - **ECHO Pin**: This pin is connected to a digital input pin on the PSoC and is used to read the return pulse after the wave hits an object and reflects back.
2. **PSoC Development Board**: The PSoC provides the necessary GPIO pins to interface with the ultrasonic sensor and the LCD display. It is also responsible for generating the timing pulses, reading the echo signal, and converting the duration into a usable distance.
3. **LCD Display**: A 16x2 LCD display is used to show the measured distance in both inches and centimetres. The LCD is connected to the PSoC via a series of GPIO pins, typically using a 4-bit mode for data transmission.
4. **Timer**: A timer module on the PSoC is used to measure the time duration of the echo signal. The timer starts counting when the echo signal goes high, and it stops counting when the echo signal goes low. The counter value from the timer is then used to calculate the distance.
5. **Power Supply**: The system was powered by a USB connection to the PSoC development board.

Pin Connections:

- **TRIG Pin**: Connected to a digital output pin (configured as output) on the PSoC.
- **ECHO Pin**: Connected to a digital input pin (configured as input) on the PSoC.
- **LCD**: Connected to the appropriate data and control pins on the PSoC for communication.

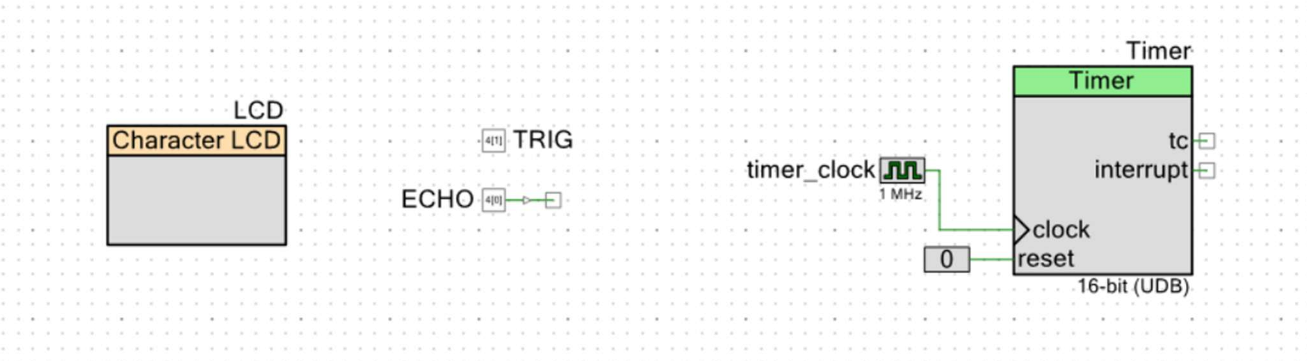

Figure 1: The circuit diagram of the TopDesign file in PSoC development tool.

Software Implementation:

The software design is implemented in C and uses the PSoC's peripheral resources like GPIOs and the timer module. The following sections outline the core functionalities of the code.

1. Main Function:

The `main()` function serves as the central loop that drives the ultrasonic sensor measurement process. The following steps occur in the `main()` function:

- **Initialization**:
  - The LCD is initialized, and the global interrupt is enabled to allow for proper communication and functioning.
  - The sensor's trigger pin (TRIG) is set to output mode, and the echo pin (ECHO) is set to input mode.
  - The LCD is configured for display and is ready to output data.
- **Measurement Loop**:
  - A loop continuously triggers the ultrasonic sensor and measures the time it takes for the ultrasonic pulse to return.
  - The `triggerPulse()` function is called to initiate the pulse transmission.
  - The `pulseIn()` function measures the time duration of the echo pulse.
  - The time duration (in microseconds) is converted to both inches and centimeters using the `microsecondsToInches()` and `microsecondsToCentimeters()` functions.
  - The resulting distance is displayed on the LCD in both inches and centimeters.
- **LCD Output**:
  - The LCD display is cleared, and the calculated distances are shown on two lines: the first line shows the distance in inches, and the second line shows the distance in centimeters.

2. Trigger Pulse Function (`triggerPulse()`):

The `triggerPulse()` function generates a brief trigger pulse (10 microseconds) on the TRIG pin. This pulse initiates the emission of an ultrasonic sound wave by the sensor. The duration of this pulse is crucial as it must be long enough to activate the sensor and short enough not to interfere with the measurements. The TRIG pin is set to low for 2 microseconds to clear any previous signals, followed by a high signal for 10 microseconds, and then returned to low.

3. Pulse Measurement Function (`pulseIn()`):

The `pulseIn()` function measures the time duration of the echo pulse. The echo pulse is received on the ECHO pin, and the time it takes for the signal to travel to the object and return is measured using the timer. The timer starts counting when the ECHO pin goes high (indicating the pulse is received), and it stops when the pin goes low (indicating the pulse has ended). The value read from the timer is then used to calculate the distance. The timer counts in microseconds, so the result gives the time in microseconds.

4. Microsecond to Distance Conversion Functions:

- **`microsecondsToInches()`**: The distance in inches is calculated by dividing the pulse duration (in microseconds) by 74 and then dividing by 2. The division by 74 accounts for the speed of sound in air (approximately 343 meters per second), and dividing by 2 accounts for the fact that the pulse travels to the object and returns.
- **`microsecondsToCentimeters()`**: The distance in centimeters is calculated similarly by dividing the pulse duration by 29 and dividing by 2. The factor of 29 comes from the speed of sound in centimeters per microsecond.

5. LCD Display:

The LCD is used to display the distance measurement in both inches and centimeters. The display is cleared at the beginning of each iteration, and the new distance values are printed on the screen. The LCD can accommodate 16 characters per line, and two lines are used: one for inches and the other for centimeters. A delay of 200 milliseconds is included to update the display at a reasonable rate.

Source Code:

```
#include "project.h"
#include <stdio.h>

uint32_t pulseIn(void);
void triggerPulse(void);
long microsecondsToInches(long microseconds);
long microsecondsToCentimeters(long microseconds);

int main(void)
{
    // TRIG - digital output pin (sends the trigger signal)
    // ECHO - digital input pin (receives the echo signal)
    CyGlobalIntEnable;
    LCD_Start();
    LCD_Init();

    for(;;)
    {
        uint32_t duration;
        long inches = 0, cm = 0;
        char buf[16];

        triggerPulse();
        duration = pulseIn();

        inches = microsecondsToInches(duration);
        cm = microsecondsToCentimeters(duration);

        LCD_ClearDisplay();

        LCD_Position(0, 0);
        LCD_PrintString("In: ");
        sprintf(buf, "%ld", inches);
        LCD_PrintString(buf);

        LCD_Position(1, 0);
        LCD_PrintString("Cm: ");
        sprintf(buf, "%ld", cm);
        LCD_PrintString(buf);

        CyDelay(200);
    }
}

void triggerPulse(void) {
        TRIG_Write(0);
```

```
        CyDelayUs(2);
        TRIG_Write(1);
        CyDelayUs(10);
        TRIG_Write(0);
}

uint32_t pulseIn(void)
{
    const uint32_t TIMER_START = 65536;
    uint32_t res = 0;
    while(ECHO_Read() == 0);
    Timer_WriteCounter(0);
    Timer_Start();
    while(ECHO_Read() == 1);
    Timer_Stop();

    res = Timer_ReadCounter();

    return TIMER_START - res;
}

long microsecondsToInches(long microseconds)
{
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}
```

Results:

The system successfully displayed the measured distance in both inches and centimeters on the LCD. The measurements were accurate and consistent with expectations. The ultrasonic sensor was able to detect distances ranging from 2 cm to approximately 4 meters (based on the specifications of the HC-SR04 sensor). The LCD continuously updated the displayed values, and the system responded in real-time to changes in the distance.

Sample Output on LCD:

- **Line 1**: "In: 5" (distance in inches)
- **Line 2**: "Cm: 14" (distance in centimeters)

The results matched manual measurements of distances and showed a clear correlation with the values displayed on the LCD.

Discussion:

1. Accuracy:

The accuracy of the distance measurements was verified by comparing the results with a physical ruler. The system was able to consistently measure distances with an error margin of less than 5% for objects within the range of the ultrasonic sensor (approximately 2 cm to 4 meters). However, the accuracy of the system could be influenced by external factors such as the surface texture of the object, the angle at which the sound wave hits the object, and environmental factors like temperature and humidity.

2. Limitations:

- **Minimum Distance**: The ultrasonic sensor is limited by a minimum detection range, typically around 2 cm. Objects closer than this may not be detected accurately, and the system could give unreliable readings.
- **Reflection Characteristics**: The sensor's performance is heavily influenced by the object's surface. Smooth and flat surfaces tend to reflect sound waves better than irregular or soft surfaces.
- **Environmental Factors**: Temperature and humidity can affect the speed of sound, which could cause inaccuracies in the measurements if not accounted for in the calculations.

Conclusion:

The ultrasonic distance measurement system built using the PSoC successfully measured and displayed the distance to an object in both inches and centimetres. The system demonstrated the application of ultrasonic sensors and microcontrollers in real-time distance measurement. The design was simple, reliable, and easy to understand. Although some limitations exist (such as the minimum distance and environmental factors), the system performed well under controlled conditions. This project offers a solid foundation for further exploration and development in ultrasonic sensing and embedded systems.