Ultrasonic Distance Measurement Using STM32 & 16×2 LCD

Project Title
Distance Measurement Using Ultrasonic Sensor and 16×2 LCD with STM32H563

Objective
To measure the distance of an object using an **HC-SR04 ultrasonic sensor** and display the measured distance on a **16×2 I²C LCD** using **STM32H563 (Sourya Board)**.

Components Used

Component	Quantity
STM32H563 (Sourya Board)	1
16×2 LCD with I²C Module	1
HC-SR04 Ultrasonic Sensor	1
ST-Link V2	1
5V AC–DC SMPS	1
Jumper Wires	As required

Pin Connections

LCD Connections

LCD Pin	Connection
VCC	5V from SMPS
GND	GND of SMPS & STM32
SDA	SDA1 of STM32
SCL	SCL1 of STM32

Ultrasonic Sensor Connections

HC-SR04 Pin	STM32 Pin
VCC	5V from SMPS
GND	GND of SMPS & STM32
TRIG	PC8
ECHO	PC9

Note:
ECHO pin gives 5V output. Use a **voltage divider** to protect STM32 GPIO.

ST-Link V2 Connections

ST-Link Pin	STM32 Pin
Pin 1	3.3V
Pin 7	SWDIO
Pin 9	SWCLK
Pin 12	GND

STM32CubeIDE Project Setup

Step 1: Create Project

- Open **STM32CubeIDE**
- Select **STM32H563ZIT6**
- Name project:
- ultrasonic_sensor

Step 2: Configure I2C

Pin	Function
PB6	I2C1_SCL
PB7	I2C1_SDA

Enable **I2C1** from Connectivity tab.

Step 3: Configure GPIO

Pin	Mode
PC8	GPIO Output (Trigger)
PC9	TIM3_CH4 (Echo)

Step 4: Timer Configuration (TIM3)

Parameter	Value
Mode	Input Capture
Channel	CH4
Prescaler	72
Counter Period	65535
Interrupt	Enabled

➞ Timer runs at **1 MHz** for microsecond accuracy.

Ultrasonic Working Principle

1. TRIG pin sends 10µs pulse
2. Ultrasonic wave is transmitted
3. Echo pin receives reflected signal
4. Time difference is measured
5. Distance calculated using:

Distance = (Time × Speed of Sound) / 2
Distance = Time × 0.034 / 2

Code Implementation

Delay Function (Microseconds)

```
void delay(uint16_t time)
{
  __HAL_TIM_SET_COUNTER(&htim3, 0);
  while (__HAL_TIM_GET_COUNTER(&htim3) < time);
}
```

Timer Input Capture Callback

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
  if (htim->Instance == TIM3 &&
    htim->Channel == HAL_TIM_ACTIVE_CHANNEL_4)
  {
    if (Is_First_Captured == 0)
    {
      IC_Val1 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_4);
      Is_First_Captured = 1;
      __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_4,
                    TIM_INPUTCHANNELPOLARITY_FALLING);
    }
    else
    {
      IC_Val2 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_4);

      if (IC_Val2 > IC_Val1)
        Difference = IC_Val2 - IC_Val1;
      else
        Difference = (0xFFFF - IC_Val1) + IC_Val2;

      Distance = (float)Difference * 0.034f / 2.0f;

      Distance_Ready = 1;
      Is_First_Captured = 0;

      __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_4,
                    TIM_INPUTCHANNELPOLARITY_RISING);
      __HAL_TIM_DISABLE_IT(&htim3, TIM_IT_CC4);
    }
  }
```

```
}
```

Ultrasonic Trigger Function

```
void HCSR04_Read(void)
{
  Is_First_Captured = 0;
  __HAL_TIM_SET_COUNTER(&htim3, 0);
  __HAL_TIM_ENABLE_IT(&htim3, TIM_IT_CC4);

  HAL_GPIO_WritePin(GPIOC, GPIO_PIN_8, GPIO_PIN_SET);
  HAL_Delay(1);
  HAL_GPIO_WritePin(GPIOC, GPIO_PIN_8, GPIO_PIN_RESET);
}
```

LCD Driver Files

📁 **lcd_i2c.c**

Refer to the 16*2 LCD Interfacing

📁 **lcd_i2c.h**

Refer to the 16*2 LCD Interfacing

Main Function Code

Global Variables

```
volatile uint32_t IC_Val1 = 0;
volatile uint32_t IC_Val2 = 0;
volatile uint32_t Difference = 0;
volatile uint8_t Is_First_Captured = 0;
volatile float Distance = 0;
volatile uint8_t Distance_Ready = 0;
```

In main() before while loop:

```
HAL_TIM_IC_Start_IT(&htim3, TIM_CHANNEL_4);

lcd_init();
lcd_put_cur(0, 0);
lcd_send_string("Distance:");
```

Inside while(1):

```
HCSR04_Read();
HAL_Delay(60);

if (Distance_Ready)
{
  Distance_Ready = 0;
```

```
    int dist = (int)(Distance * 100);

    char lcdBuffer[16];
    sprintf(lcdBuffer, "%d.%02d cm", dist / 100, dist % 100);

    lcd_put_cur(1, 0);
    lcd_send_string("            ");
    lcd_put_cur(1, 0);
    lcd_send_string(lcdBuffer);
}
```

Output

LCD Display:
Distance:
23.45 cm

✓ Updates continuously

✓ Accurate distance measurement

✓ Real-time display

Conclusion

This project successfully demonstrates:

✓ Ultrasonic distance measurement

✓ STM32 timer input capture

✓ I2C LCD interfacing

✓ Real-time embedded system application

The system accurately measures distance and displays it on a 16×2 LCD.

Applications

- Obstacle detection
- Parking assistance
- Distance measurement systems
- Robotics
- Automation projects