

Introduction to Arduino

Objective

To understand the basic working of Arduino through interfacing with Ultrasonic sensor and LCD display.

Arduino Functions Reference

pinMode()

Configures the specified pin to behave as an input (INPUT) or an output (OUTPUT).

Syntax: `pinMode(pin, mode);`

digitalWrite()

Writes a HIGH (5V) or LOW (0V) value to a digital output pin.

Syntax: `digitalWrite(pin, value);`

delay()

Pauses the program for the amount of time specified in milliseconds (ms).

Syntax: `delay(ms);`

delayMicroseconds()

Pauses the program for the amount of time specified in microseconds (μ s).

Syntax: `delayMicroseconds(us);`

pulseIn()

Reads a pulse (either HIGH or LOW) on a pin. It waits for the pin to go to the specified state, starts timing, waits for the pin to go to the opposite state, and stops timing. The return value is the length of the pulse in microseconds.

Syntax: `pulseIn(pin, value);`

Serial.begin()

Starts the serial data transmission at the specified data rate (baud rate). The data rate is given in bits/sec. The standard rates are 9600, 57600 and 115200 bits/sec. The Arduino IDE's Serial Monitor is usually configured to open at 9600 baud by default

Syntax: `Serial.begin(speed);`

Serial.print()

Prints data to the serial port as human-readable text. It keeps the cursor on the same line.

Syntax: `Serial.print(data);`

Serial.println()

Prints data to the serial port as human-readable text, followed by a newline character to move the cursor to the next line.

Syntax: `Serial.println(data);`

lcd.begin()

Initializes the LCD. Specifies the dimensions of the display (e.g., 16 columns and 2 rows). It must be called once in `setup()`.

Syntax: `lcd.begin(cols,rows);`

lcd.clear()

Clears the entire display and immediately sets the cursor position back to the home position (0, 0). This is used to refresh the screen content in each loop iteration.

Syntax: `lcd.clear();`

lcd.setCursor()

Sets the position of the LCD cursor. Subsequent text will be printed starting at this location. The indexing starts at 0 index.

Syntax: `lcd.begin(cols,rows);`

lcd.print()

Displays text or numeric data on the LCD screen at the current cursor position. This is the equivalent of `Serial.print()`, but for the physical LCD screen.

Syntax: `lcd.print(data);`

Part 1: Ultrasonic Sensor with Serial Monitor Output

Overview

In this experiment, we will learn to program simple applications using an **HC-SR04 Ultrasonic Sensor** connected to an **Arduino UNO** board.

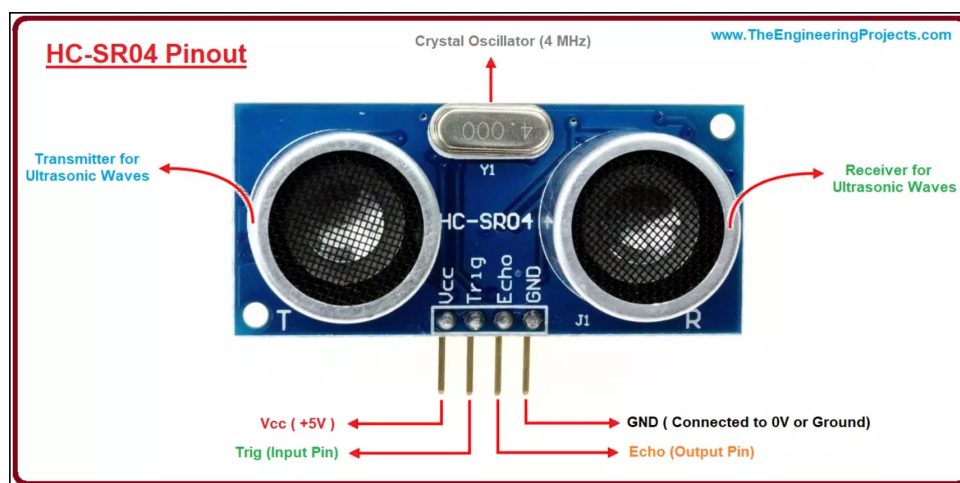


Figure 1: Pin diagram of HC-SR04

Components Required

- Arduino UNO and USB Cable
- Breadboard
- HC-SR04 Ultrasonic Sensor
- Jumper Wires

Circuit Connections:

- Echo pin → 12
- Trigger pin → 13
- VCC → 5V
- GND → GND

Procedure:

1. Upload the code to the Arduino.
2. Place obstacles in front of your Ultrasonic Sensor and vary the distance as you like.
3. Note the readings on the Serial Monitor.

Part 2: Interfacing an LCD Screen

Overview

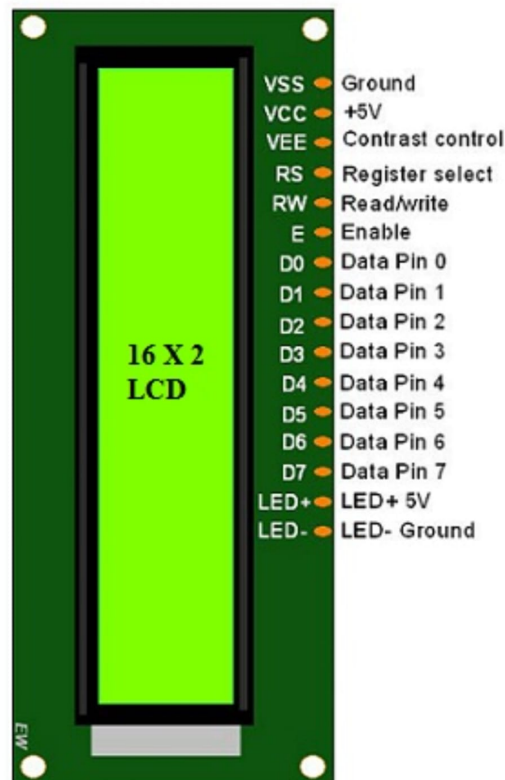
In this part, we will learn how to use an **LCD screen** as a display for showing text and data using Arduino.

Components Required

- Arduino UNO and USB cable
- Breadboard
- 16×2 LCD display
- Potentiometer (for contrast control)
- Jumper wires

Procedure

1. Open the Arduino IDE and use the built-in **LiquidCrystal** library (not `LiquidCrystal I2C`).
2. Load the example sketch: File > Examples > LiquidCrystal > HelloWorld.
3. Set the correct pin configuration in the code: `LiquidCrystal lcd(12, 11, 5, 4, 3, 2);`
4. Upload the code to your Arduino board.
5. Adjust the potentiometer until the text on the LCD is clearly visible.
6. Modify the code to display custom messages for testing.



LCD-16x2-pin-diagram

Figure 2: Pin diagram of LCD screen

LCD Connections

LCD to Arduino Connections (4-bit Mode)

LCD Pin	Label	Connection
1	GND	Connect to Arduino GND
2	VCC	Connect to Arduino 5V
3	VEE	Connect to the middle pin of potentiometer ; connect the potentiometer ends to 5V and GND (for contrast adjustment)
4	RS	Connect to Arduino Digital Pin 12
5	RW	Connect to GND (for write mode)
6	E	Connect to Arduino Digital Pin 11
11	D4	Connect to Arduino Digital Pin 5
12	D5	Connect to Arduino Digital Pin 4
13	D6	Connect to Arduino Digital Pin 3
14	D7	Connect to Arduino Digital Pin 2
15	LED+ (A)	Connect to 5V through a 220Ω resistor (for LCD backlight, if available)
16	LED- (K)	Connect to GND (backlight return)

Part 3: Integration – LCD + Ultrasonic Sensor

Overview

In this part, we will integrate the **ultrasonic distance sensor** with the **LCD display** to show the measured distance in real time.

Procedure

1. Connect the ultrasonic sensor and LCD to the Arduino as described in the earlier sections.
2. Include the necessary libraries at the top of your code
3. Define the pin connections in your code
4. In the `setup()` function, initialize the LCD and sensor pins
5. In the `loop()` function, measure the distance and display it
6. Upload the code to your Arduino.
7. Observe the LCD display — it should continuously update the distance measured by the ultrasonic sensor.