**Ultrasonic Distance Sensor using 8051**

**1. Abstract**

This project demonstrates a real-time distance measurement system that employs an HC-SR04 ultrasonic sensor to detect objects. The measured distance is displayed on a pre-soldered I2C 16×2 LCD, interfaced with an 8051 development board. The 8051 is programmed using an Arduino Uno acting as an ISP (In-System Programmer) with AVRDude. This document provides detailed hardware connections, code with comments, and an explanation of the working principles.

**2. Objective**

* **Measure Distance:** Use the HC-SR04 ultrasonic sensor to measure the distance of an object.
* **Display Results:** Show the distance measurement (in centimeters) on the I2C LCD.
* **Microcontroller Programming:** Program the 8051 development board via an Arduino Uno used as an ISP.
* **I2C Communication:** Implement I2C communication on the 8051 by connecting SDA to P0.7 and SCL to P0.6.

**3. Components Used**

* **8051 Development Board**
* **I2C 16×2 LCD Module** (pre-soldered; uses SDA and SCL)
* **HC-SR04 Ultrasonic Sensor**
* **Arduino Uno** (used as ISP for programming the 8051)
* **Jumper Wires**

*Note:* Most passive components (oscillator, capacitors, resistors) are integrated on the development board.

**4. Circuit Connections**

**4.1. I2C LCD Module**

* **VCC:** Connected to 5V on the 8051 development board
* **GND:** Connected to Ground
* **SCL:** Connected to **P0.6**
* **SDA:** Connected to **P0.7**

**4.2. HC-SR04 Ultrasonic Sensor**

* **VCC:** Connected to 5V
* **GND:** Connected to Ground
* **Trig:** Connected to **P3.3**
* **Echo:** Connected to **P3.2**

**4.3. Arduino Uno ISP Programming Interface**

* **Arduino Uno D10 (RST):** Connected to the 8051’s Reset pin
* **Arduino Uno D11 (MOSI):** Connected to the 8051’s MOSI (data line for ISP)
* **Arduino Uno D12 (MISO):** Connected to the 8051’s MISO (data line for ISP)
* **Arduino Uno D13 (SCK):** Connected to the 8051’s SCK
* **Common Ground:** All GND lines tied together

**5. Working Principle**

1. **Ultrasonic Measurement:**
   * The HC-SR04 sensor emits an ultrasonic pulse when the trigger pin (P3.3) is activated.
   * The pulse travels until it encounters an object and then reflects back to the sensor.
   * The time interval between sending and receiving the pulse is measured using Timer0 in the 8051.
2. **Distance Calculation:**
   * The time measured is converted to distance (cm) using the formula:

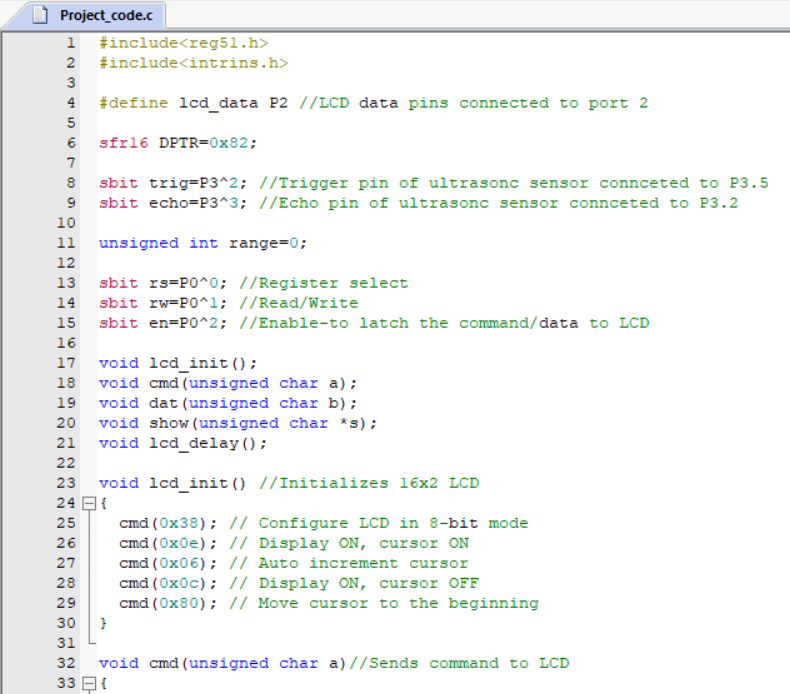


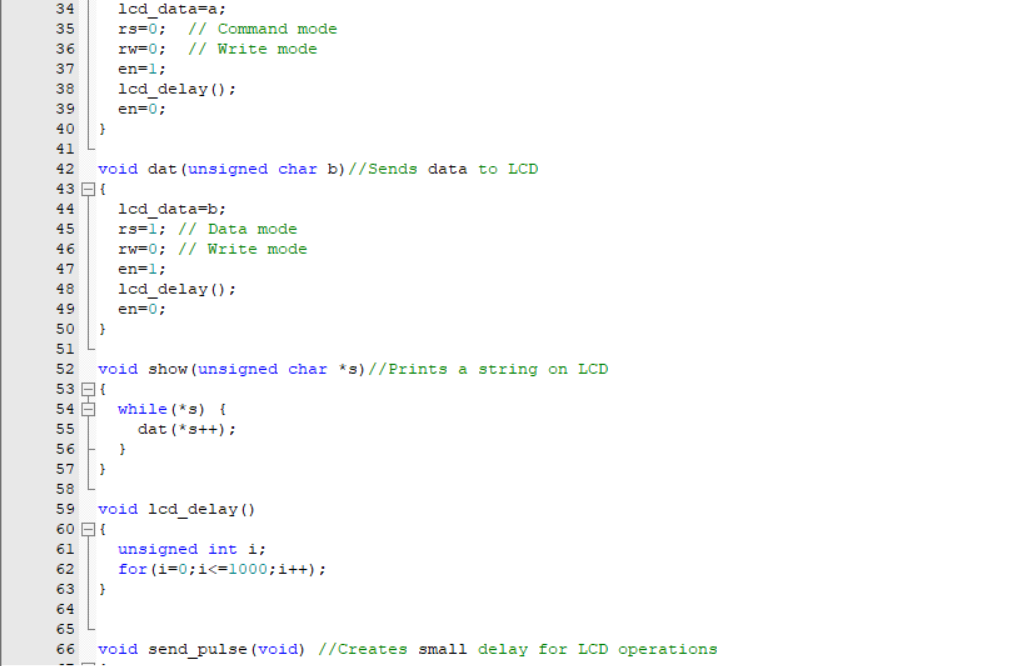
* + This conversion factor (59) accounts for the round-trip travel time of the pulse in air.

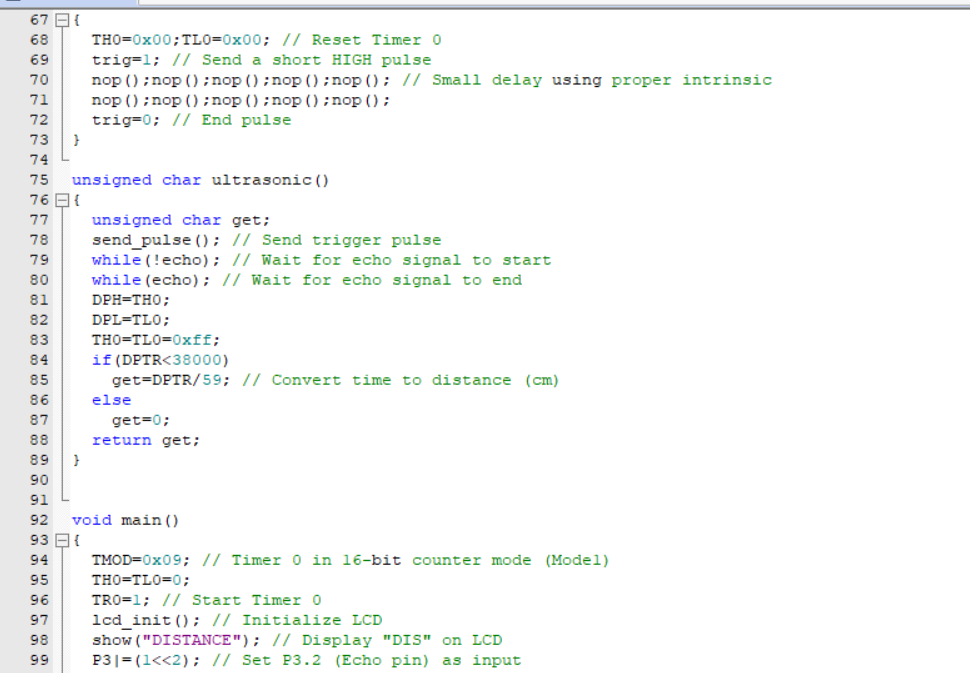
1. **I2C LCD Display:**
   * The measured distance is sent to an I2C 16×2 LCD using a bit-banged I2C protocol with SDA on P0.7 and SCL on P0.6.
   * The LCD displays the text “DISTANCE” and updates the distance value continuously.
2. **Programming via Arduino Uno:**
   * The compiled code is uploaded to the 8051 using an Arduino Uno configured as an ISP with AVRDude.

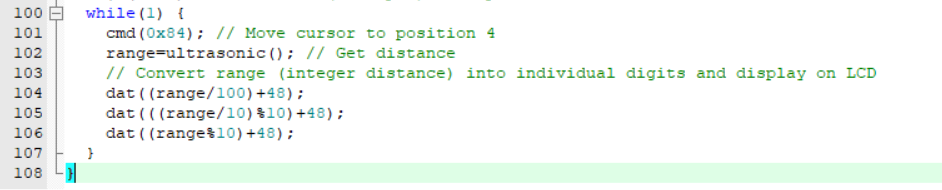
**6. Embedded C Code**

Below is the complete embedded C code with detailed comments. This version implements basic I2C routines and integrates ultrasonic sensing.









**Code Explanation:**

* **I2C Routines:** i2c\_start(), i2c\_stop(), and i2c\_write() implement a simple bit-banging I2C master protocol to communicate with the LCD module.
* **LCD Functions:** lcd\_send\_nibble(), lcd\_send\_byte(), lcd\_cmd(), and lcd\_data() break down data into nibbles for transmission, including proper enable pulsing and RS control.
* **Ultrasonic Functions:** send\_pulse() triggers the sensor and ultrasonic() calculates the distance in centimeters using Timer0.
* **Main Routine:** Initializes the timer and LCD, continuously reads the distance, and updates the display.

A computer screen with white text

AI-generated content may be incorrect.

* To upload the HEX file to the AT89S52 microcontroller, the avrdude.exe tool is used in combination with the avrdude.conf configuration file and an Arduino Uno acting as the programmer. The command executed in the Command Prompt includes several key parameters: -c stk500v1 specifies the programmer type, where stk500v1 refers to the STK500 version 1 protocol, which is compatible with the Arduino Uno’s bootloader and enables it to communicate with the target microcontroller. The -P COM9 option designates the appropriate serial port to which the Arduino is connected, and -b 19200 sets the baud rate for communication. The -U flash:w:"path\_to\_file.hex":a flag instructs avrdude to write the specified HEX file to the flash memory of the microcontroller, using the correct file format. Upon executing the command, avrdude initiates the programming process by communicating with the Arduino, which then transfers the HEX file to the AT89S52 via SPI. If the setup is correct and the toolchain is properly configured, the process concludes successfully with a confirmation message. However, it is important to note that since the AT89S52 belongs to the 8051 family and is not natively supported by avrdude, this method may require a modified version of the tool or is alternatively performed using a Python-based uploader for more consistent results.

**7. Software Tools and Programming**

* **Keil uVision:** For writing, compiling, and debugging the embedded C code.
* **AVRDude with Arduino Uno (ArduinoISP):** Used to upload the compiled code (.hex file) from Keil to the 8051 development board.

**8. Applications**

* **Obstacle Detection:** For robotics or automotive parking assistance.
* **Proximity Sensors:** In security systems or automated lighting control.
* **Level Measurement:** In industrial applications, such as monitoring liquid levels.

**9. Observations and Results**

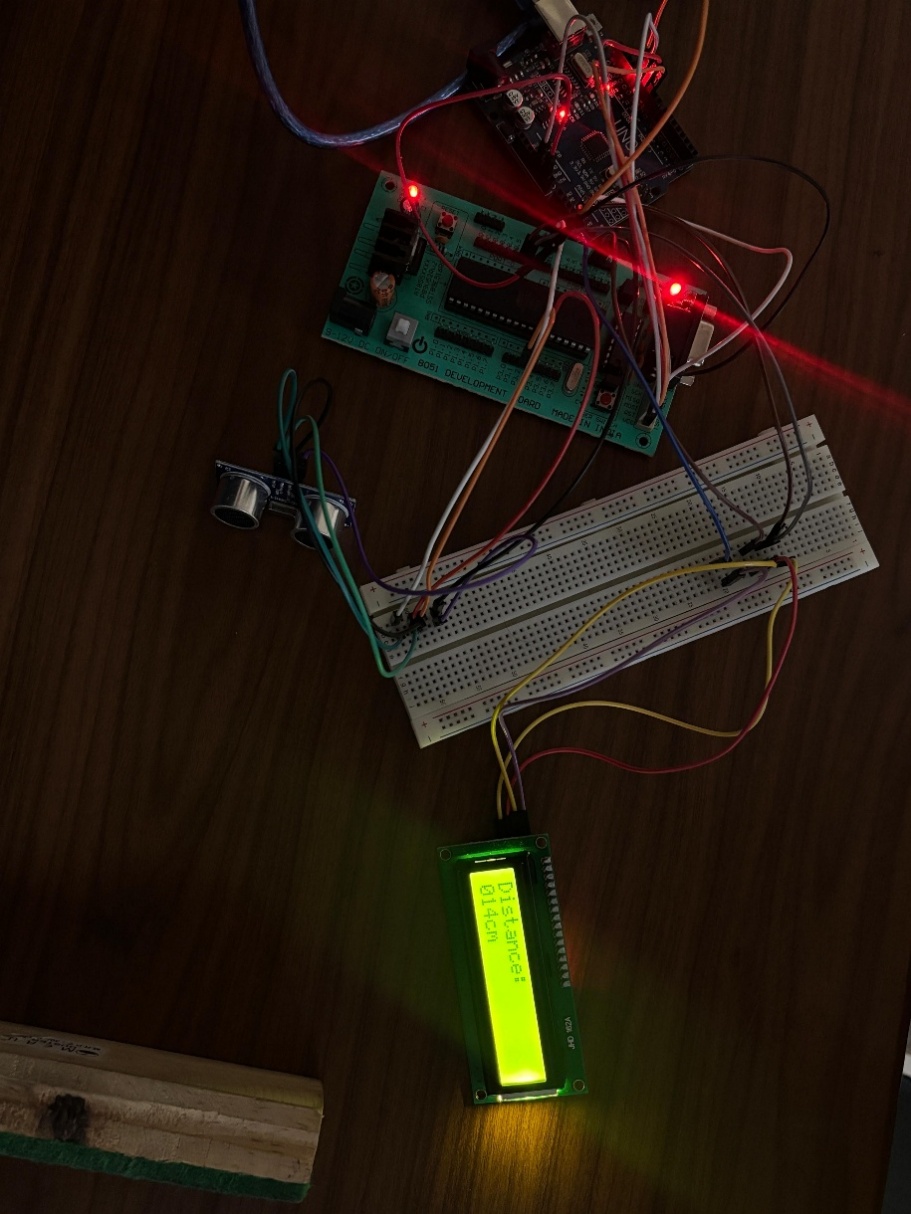
The system successfully measured distances within the HC-SR04 range (typically 2cm to 400cm), with an acceptable error margin. The LCD displayed a stable reading that updated approximately every 200 milliseconds.

**10. Conclusion**

This project successfully demonstrates an embedded solution for distance measurement using an ultrasonic sensor and an 8051 microcontroller, with data output via an I2C 16×2 LCD. The use of an Arduino Uno as an ISP shows a cost-effective and flexible method to program legacy microcontrollers. Future improvements could include adding alarms for specific distances and integrating more robust I2C communication libraries.

**11. Appendices**

**11.1. Project Photos**



**11.2. References**

* 8051 Microcontroller documentation
* HC-SR04 Ultrasonic Sensor datasheet
* I2C LCD module user manuals