# ULTRASONIC RANGE FINDER USING 8051 MICROCONTROLLER

# PROJECT REPORT

Submitted for the course: Microcontroller and its applications (ECE3003)

By

|  |  |
| --- | --- |
| NAME | PHONE NUMBER |
| M KALAIPRIYAN-16BEC0151 | 7358466004 |
| RAAKESH R V 16BEC0202 | 7373745009 |
| BALA GANESAN A K-16BEC0665 | 9626576069 |
| ADHEIK DOMNIC-16BEC0620 | 8220331914 |
| RAKESH RAMJI S – 16BEC0664 | 9543575251 |

**Slot: L29+L30**

**Submitted to: Dhanabal R**

## (SCHOOL OF ELECTRONICS ENGINEERING)

**ACKNOWLEDGEMENTS**

We would like to thank our faculty Dhanabal R Sir for helping and guiding us through the project. We would also like to thank VIT library for providing us with valuable sources for this project.

We would also like to acknowledge the VIT University Management and our School Dean for giving us an opportunity to carry out this project in the university.

GROUP MEMBERS:

INDEX

|  |  |  |
| --- | --- | --- |
| Serial no. | Contents | Page no. |
| 1 | Aim | 4 |
| 2 | Components Required | 5 |
| 3 | Flow diagram | 6 |
| 4 | Block diagram | 7 |
| 5 | Pin Diagram/ Circuit Diagram | 8 |
| 6 | Instruction Set | 10 |
| 7 | Description | 12 |
| 8 | Design | 13 |
| 9 | LCD interface Diagram | 13 |
| 10 | Code in assembly level language | 14 |
| 11 | Delay Function | 19 |
| 12 | Step by step execution with register contents | 19 |
| 13 | Addressing modes, memory and time required | 24 |
| 14 | Snapshots | 26 |
| 15 | Output | 28 |
| 16 | Hardware | 30 |
| 17 | Applications | 30 |
| 18 | Advantages | 31 |
| 19 | Limitations | 31 |
| 20 | Result | 31 |
| 21 | Protus Output | 32 |
| 22 | Program status word | 32 |
| 23 | I3, i5, i7 |  |

**AIM OF THE PROJECT**

* In this project, we are building an Ultrasonic Rangefinder using 8051 Microcontroller and Ultrasonic Sensor. We have different ways to measure the distance from the range finder to an object. Our way is to use Ultra Sonic Sensor Module for distance measurement. And the measure distance will be displayed in a 16 X 2 LCD display.

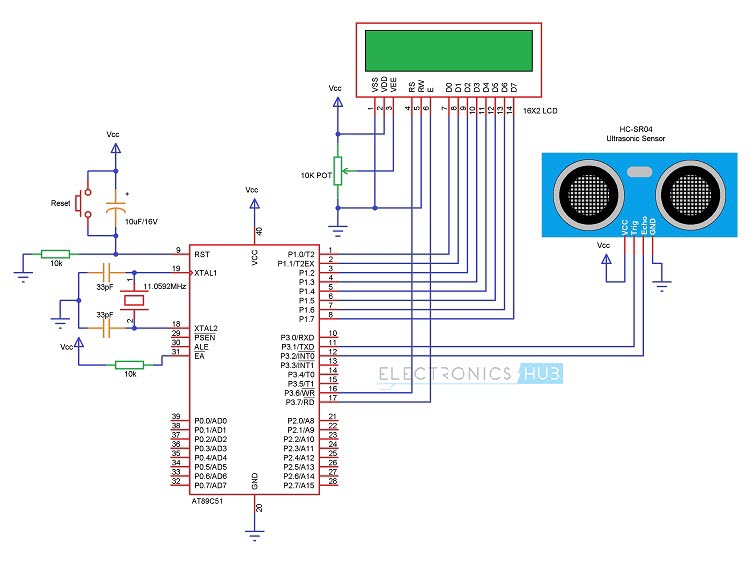
**COMPONENTS REQUIRED**

* AT89C51 Microcontroller
* 8051 Programming board
* Programming cable
* HC – SR04 Ultrasonic Module
* 16 x 2 LCD
* 10KΩ Potentiometer
* 10µF / 16V Electrolytic Capacitor
* 2 x 10KΩ Resistor (1/4 Watt)
* 11.0592 MHz Crystal
* 2 x 33pF Capacitors
* Push Button
* Connecting wires
* Power Supply
* Keil µVision Software

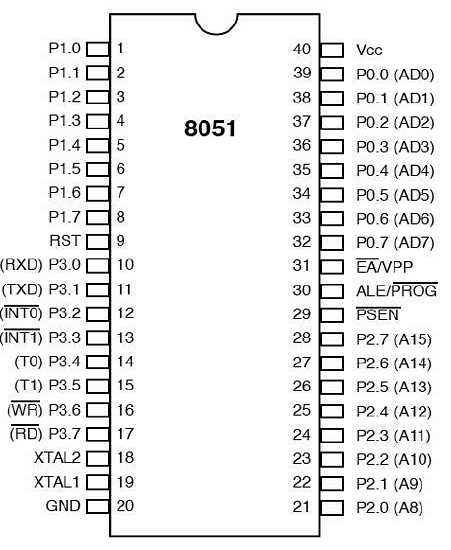
**FLOW DIAGRAM**

****

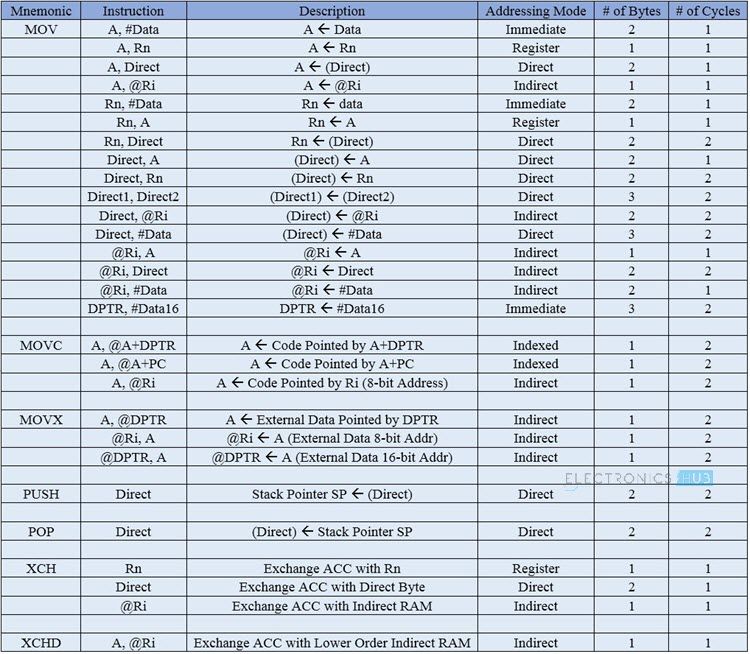
**CIRCUIT DIAGRAM**



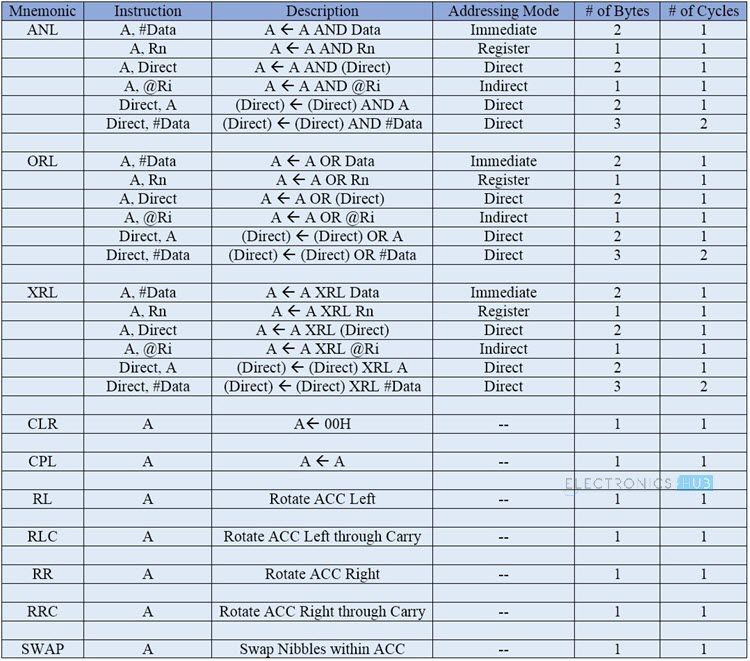
**8051 PIN DIAGRAM**



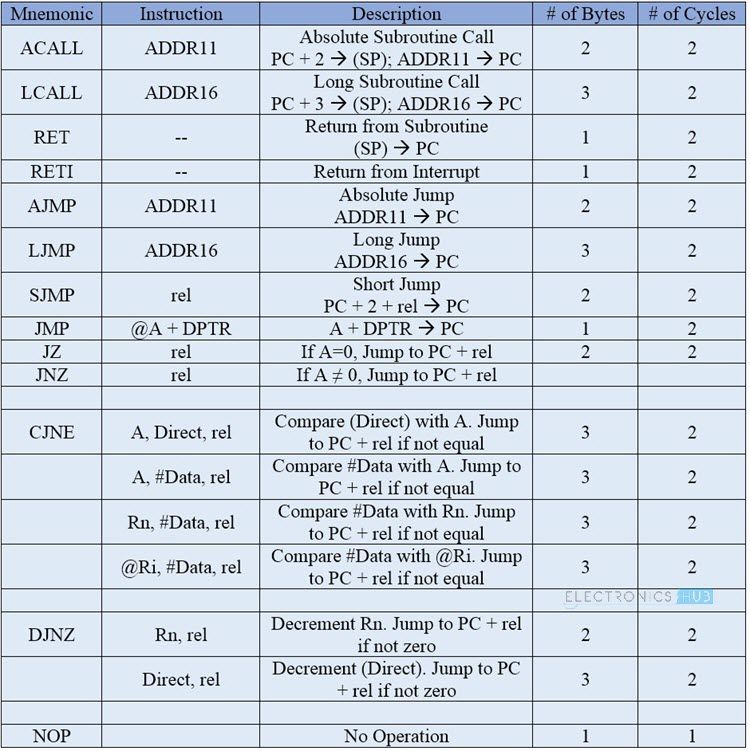
**8051 INSTRUCTION SET**







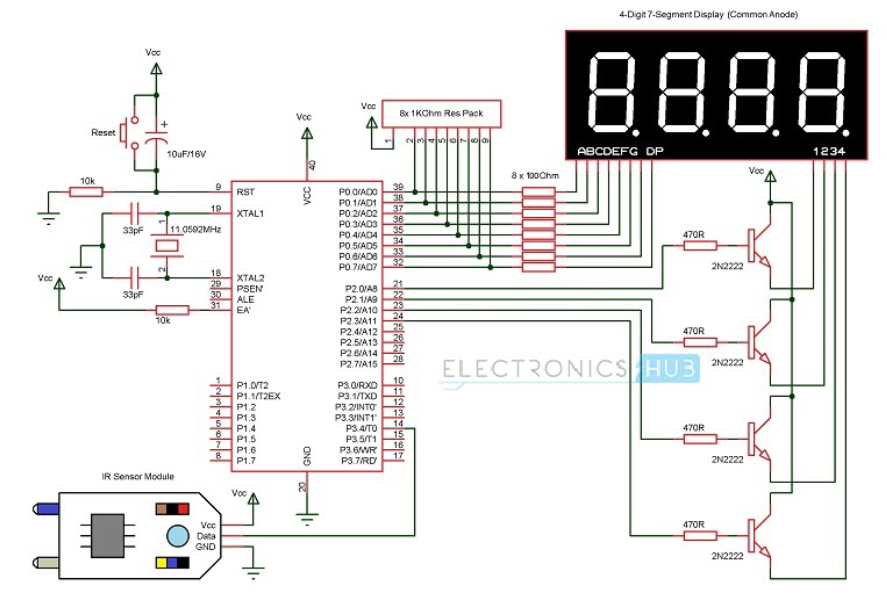




**DESCRIPTION**

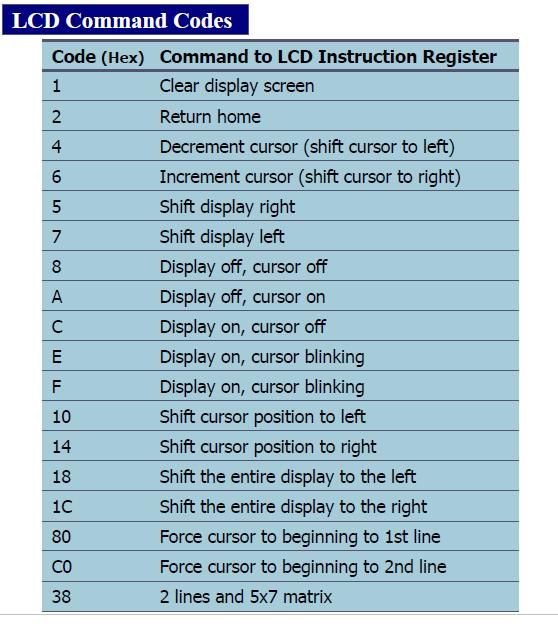
In this project, we are building an Ultrasonic Rangefinder using 8051 Microcontroller and Ultrasonic Sensor. We have different ways to measure the distance from the range finder to an object. Our way is to use Ultra Sonic Sensor Module for distance measurement. And the measure distance will be displayed in a 16 X 2 LCD display.

**LCD INTERFACE DIAGRAM**



**RS-Register select  
RS=0 ----------Command Register  
RS=1-----------Data Register**

* **RW=0----------Write  
  RW=1-----------Read**
* **E----------------Enable**



**CODE IN ASSEMBLY LEVEL LANGUAGE**

ORG 00H // origin

MOV DPTR,#LUT // moves the address of LUT to DPTR

MOV P1,#00000000B // sets P1 as output port

MOV P0,#00000000B // sets P0 as output port

CLR P3.0 // sets P3.0 as output for sending trigger

SETB P3.1 // sets P3.1 as input for receiving echo

MOV TMOD,#00100000B // sets timer1 as mode 2 auto reload timer

MAIN: MOV TL1,#207D // loads the initial value to start counting from

MOV TH1,#207D // loads the reload value

MOV A,#00000000B // clears accumulator

SETB P3.0 // starts the trigger pulse

ACALL DELAY1 // gives 10uS width for the trigger pulse

CLR P3.0 // ends the trigger pulse

HERE: JNB P3.1,HERE // loops here until echo is received

BACK: SETB TR1 // starts the timer1

HERE1: JNB TF1,HERE1 // loops here until timer overflows (ie;48 count)

CLR TR1 // stops the timer

CLR TF1 // clears timer flag 1

INC A // increments A for every timer1 overflow

JB P3.1,BACK // jumps to BACK if echo is still available

MOV R4,A // saves the value of A to R4

ACALL DLOOP // calls the display loop

SJMP MAIN // jumps to MAIN loop

DELAY1: MOV R6,#2D // 10uS delay

LABEL1: DJNZ R6,LABEL1

RET

DLOOP: MOV R5,#100D // loads R5 with 100D

BACK1: MOV A,R4 // loads the value in R4 to A

MOV B,#100D // loads B with 100D

DIV AB // isolates the first digit

SETB P1.0 // activates LED display unit D1

ACALL DISPLAY // calls DISPLAY subroutine

MOV P0,A // moves digit drive pattern for 1st digit to P0

ACALL DELAY // 1mS delay

ACALL DELAY

MOV A,B // moves the remainder of 1st division to A

MOV B,#10D // loads B with 10D

DIV AB // isolates the second digit

CLR P1.0 // deactivates LED display unit D1

SETB P1.1 // activates LED display unit D2

ACALL DISPLAY

MOV P0,A // moves digit drive pattern for 2nd digit to P0

ACALL DELAY

ACALL DELAY

MOV A,B // moves the remainder of 2nd division to A

CLR P1.1 // deactivates LED display unit D2

SETB P1.2 // activates LED display unit D3

ACALL DISPLAY

MOV P0,A // moves the digit drive pattern for 3rd digit to P0

ACALL DELAY

ACALL DELAY

CLR P1.2 // deactivates LED display unit D3

DJNZ R5,BACK1 // repeats the display loop 100 times

RET

DELAY: MOV R7,#250D // 1mS delay

LABEL2: DJNZ R7,LABEL2

RET

DISPLAY: MOVC A,@A+DPTR // gets the digit drive pattern for the content in A

CPL A // complements the digit drive pattern (see Note 1)

RET

LUT: DB 3FH // look up table (LUT) starts here

DB 06H

DB 5BH

DB 4FH

DB 66H

DB 6DH

DB 7DH

DB 07H

DB 7FH

DB 6FH

END

**DELAY FUNCTION**

DELAY: MOV R7,#250D // 1mS delay

**STEP BY STEP EXECUTION WITH REGISTER CONTENTS**

ORG 00H // origin

MOV DPTR,#LUT // moves the address of LUT to DPTR

MOV P1,#00000000B // sets P1 as output port

MOV P0,#00000000B // sets P0 as output port

CLR P3.0 // sets P3.0 as output for sending trigger

SETB P3.1 // sets P3.1 as input for receiving echo

MOV TMOD,#00100000B // sets timer1 as mode 2 auto reload timer

MAIN: MOV TL1,#207D // loads the initial value to start counting from

MOV TH1,#207D // loads the reload value

MOV A,#00000000B // clears accumulator

SETB P3.0 // starts the trigger pulse

ACALL DELAY1 // gives 10uS width for the trigger pulse

CLR P3.0 // ends the trigger pulse

HERE: JNB P3.1,HERE // loops here until echo is received

BACK: SETB TR1 // starts the timer1

HERE1: JNB TF1,HERE1 // loops here until timer overflows (ie;48 count)

CLR TR1 // stops the timer

CLR TF1 // clears timer flag 1

INC A // increments A for every timer1 overflow

JB P3.1,BACK // jumps to BACK if echo is still available

MOV R4,A // saves the value of A to R4

ACALL DLOOP // calls the display loop

SJMP MAIN // jumps to MAIN loop

DELAY1: MOV R6,#2D // 10uS delay

LABEL1: DJNZ R6,LABEL1

RET

DLOOP: MOV R5,#100D // loads R5 with 100D

BACK1: MOV A,R4 // loads the value in R4 to A

MOV B,#100D // loads B with 100D

DIV AB // isolates the first digit

SETB P1.0 // activates LED display unit D1

ACALL DISPLAY // calls DISPLAY subroutine

MOV P0,A // moves digit drive pattern for 1st digit to P0

ACALL DELAY // 1mS delay

ACALL DELAY

MOV A,B // moves the remainder of 1st division to A

MOV B,#10D // loads B with 10D

DIV AB // isolates the second digit

CLR P1.0 // deactivates LED display unit D1

SETB P1.1 // activates LED display unit D2

ACALL DISPLAY

MOV P0,A // moves digit drive pattern for 2nd digit to P0

ACALL DELAY

ACALL DELAY

MOV A,B // moves the remainder of 2nd division to A

CLR P1.1 // deactivates LED display unit D2

SETB P1.2 // activates LED display unit D3

ACALL DISPLAY

MOV P0,A // moves the digit drive pattern for 3rd digit to P0

ACALL DELAY

ACALL DELAY

CLR P1.2 // deactivates LED display unit D3

DJNZ R5,BACK1 // repeats the display loop 100 times

RET

DELAY: MOV R7,#250D // 1mS delay

LABEL2: DJNZ R7,LABEL2

RET

DISPLAY: MOVC A,@A+DPTR // gets the digit drive pattern for the content in A

CPL A // complements the digit drive pattern (see Note 1)

RET

LUT: DB 3FH // look up table (LUT) starts here

DB 06H

DB 5BH

DB 4FH

DB 66H

DB 6DH

DB 7DH

DB 07H

DB 7FH

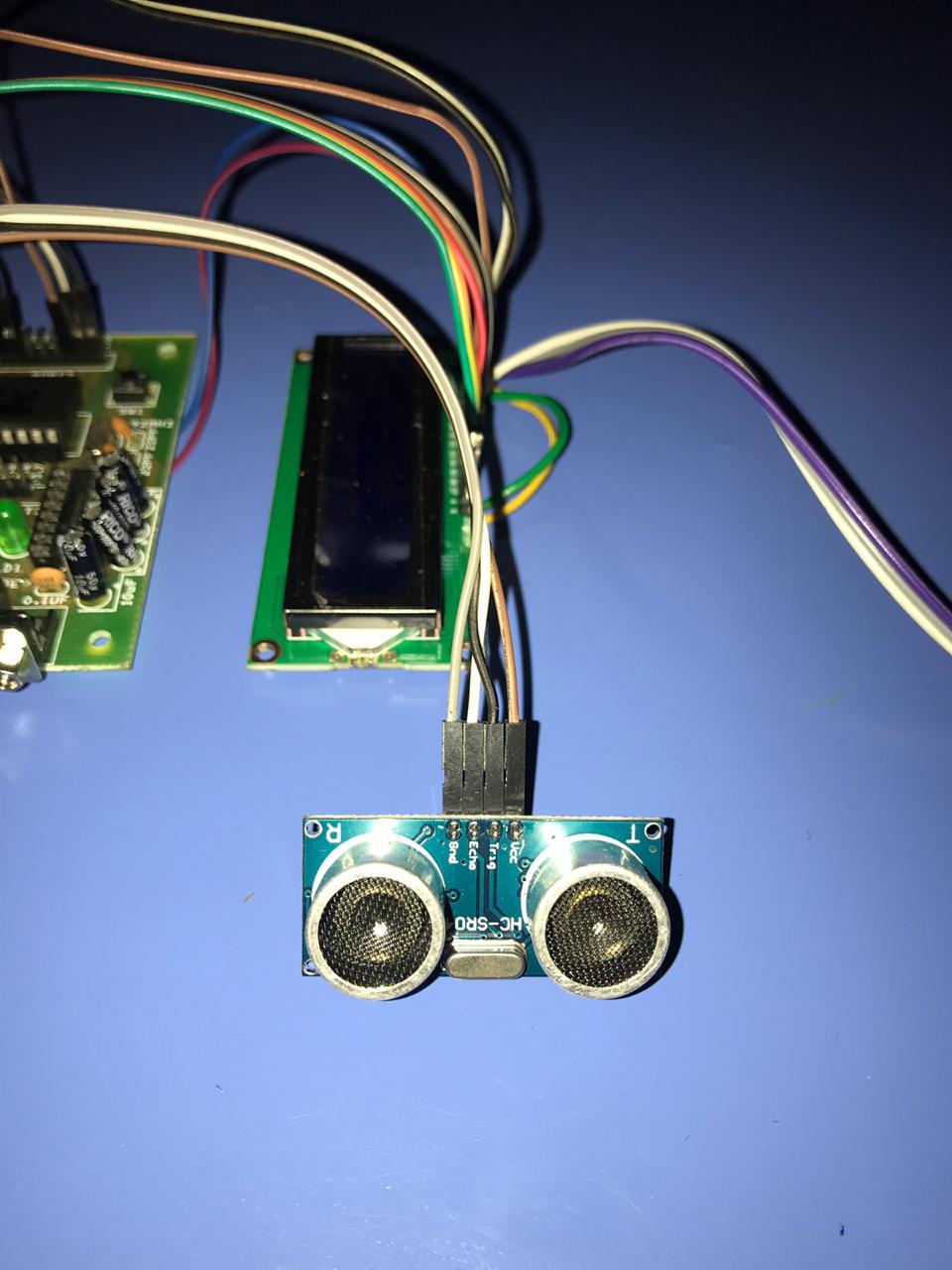
DB 6FH

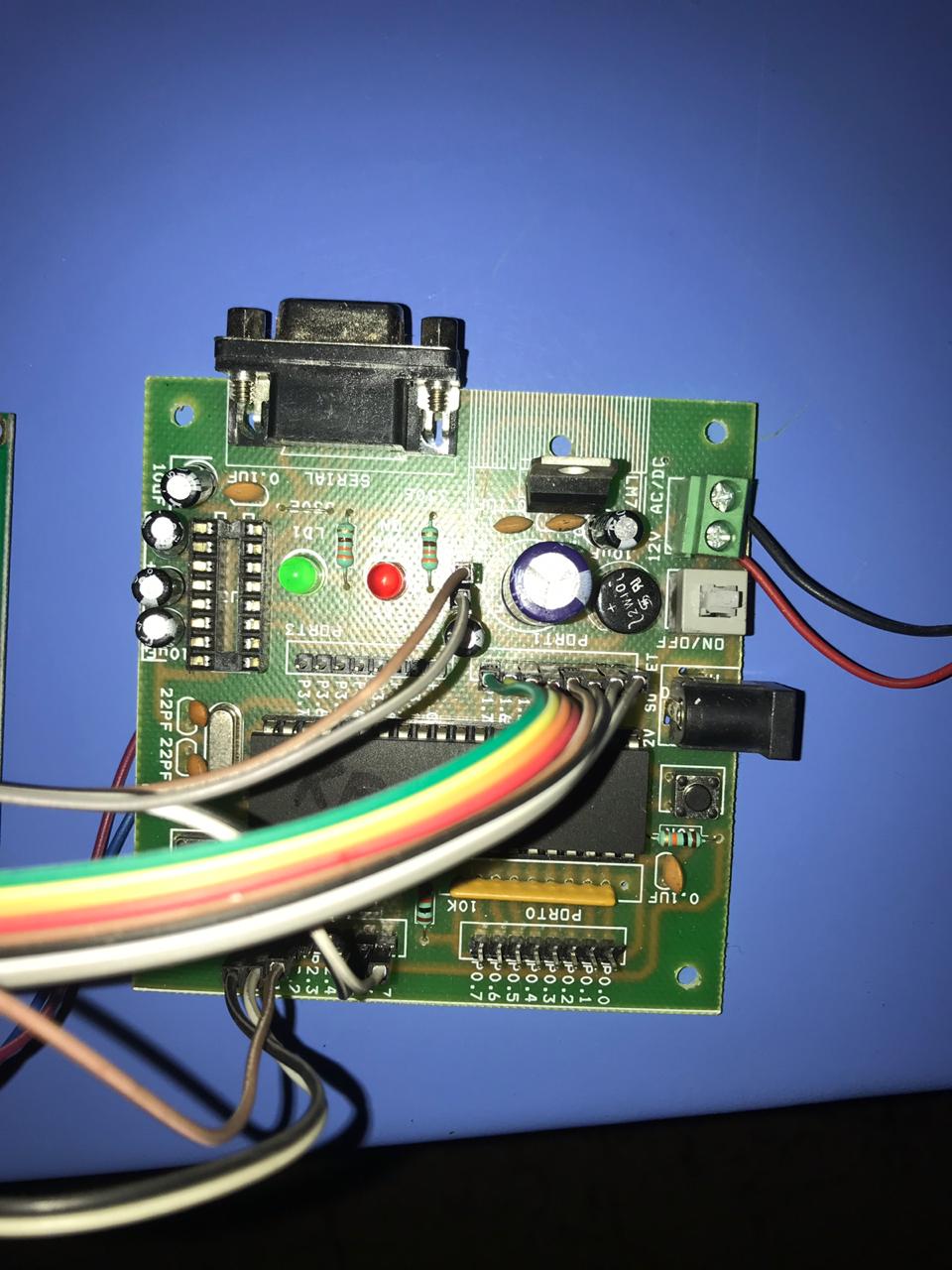
END

**ADDRESSING MODES, MEMORY AND TIME REQUIRED**

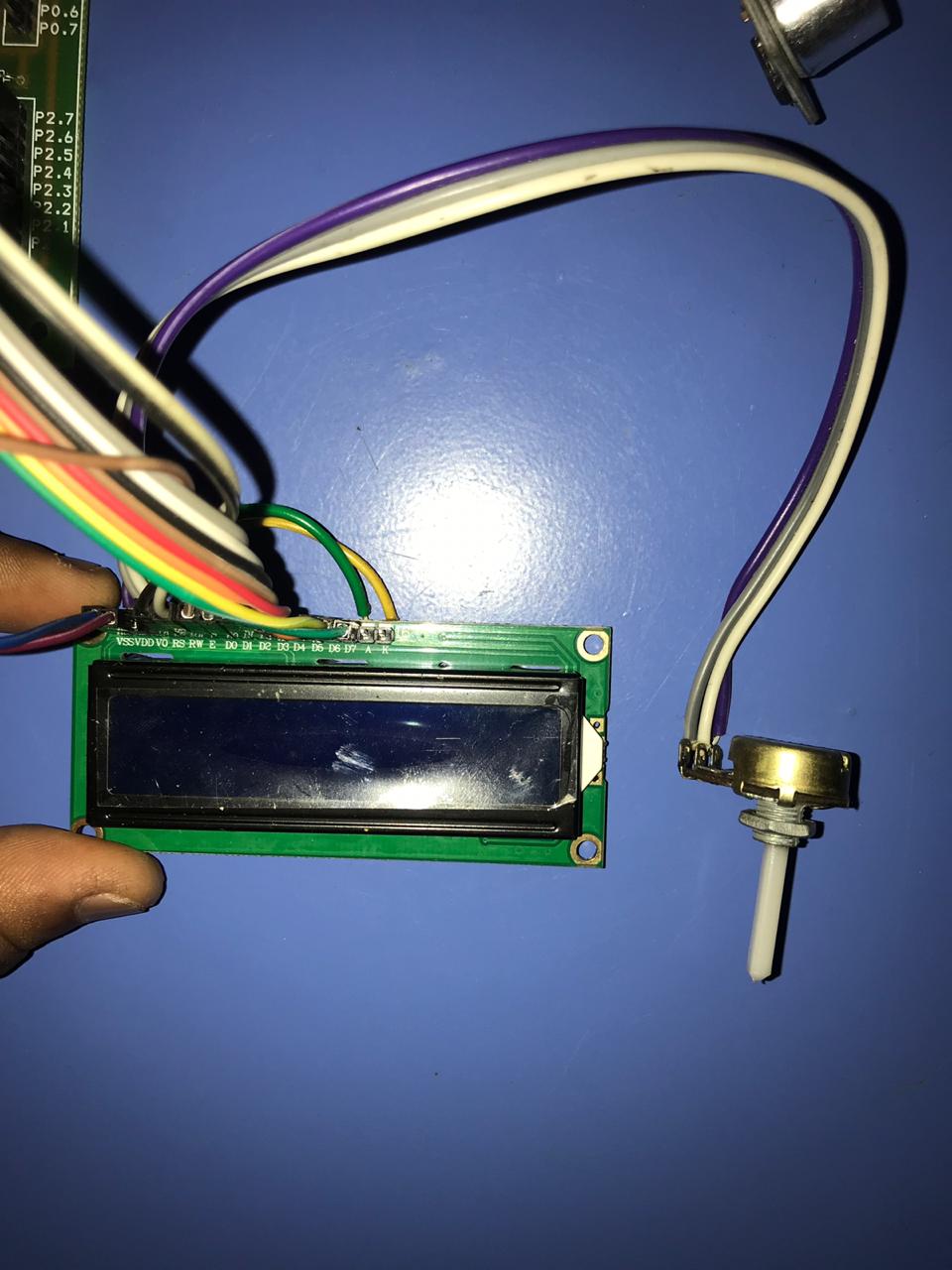
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.No. | Label | Mnemonic | Operands | Addressing mode | Memory required(BYTES) | Time taken(CYS) |
| 1. |  | ORG | 0000H |  |  |  |
| 2. |  | MOV | DPTR | Direct bit | 2 | 1 |
| 3. |  | MOV | P1 | Direct bit | 2 | 1 |
| 4. |  | MOV | P0 | Direct bit | 2 | 1 |
| 5. | MAIN | MOV | R6 | Direct | 3 | 2 |
| 6. |  | SETB | P3.5 | Direct bit | 2 | 1 |
| 7. |  | MOV | TMOD | Direct bit | 3 | 2 |
| 8. |  | MOV | TL1 | Direct bit | 3 | 2 |
| 9. |  | MOV | TH1 | Direct bit | 3 | 2 |
| 10. |  | SETB | TR1 | Absolute | 2 | 2 |
| 11. | BACK | MOV | TH0 | Direct bit | 2 | 1 |
| 12. |  | MOV | TL0 | Immediate | 2 | 1 |
| 13. |  | SETB | TR0 | Absolute | 2 | 2 |
| 14. | HERE: | JNB | TF0 | Absolute | 2 | 2 |
| 15. |  | CLR | TR0 | Immediate | 2 | 1 |
| 16 |  | CLR | TF0 | Absolute | 2 | 2 |
| 17. |  | DJNZ | R6 | Absolute | 2 | 2 |
| 18. |  | CLR | TR1 | Immediate | 2 | 1 |
| 19. |  | CLR | TF0 | Absolute | 2 | 2 |
| 20. |  | CLR | TF1 | Absolute | 2 | 2 |
| 21. |  | ACALL | DLOOP | Absolute | 2 | 1 |
| 22. |  | SJMP | MAIN | Immediate | 2 | 2 |
| 23. | DLOOP | MOV | R5 | Absolute | 2 | 2 |
| 24. | BACK1 | MOV | A ,TL1 | Immediate | 2 | 1 |
| 25. |  | MOV | B,#100D | Immediate | 2 | 2 |
| 26. |  | DIV | AB | Absolute | 2 | 2 |
| 27. |  | SETB | P1.0 | Immediate | 2 | 1 |
| 28. |  | ACALL | DISPLAY | Absolute | 2 | 2 |
| 29. |  | MOV | P0,A | Immediate | 2 | 2 |
| 30. |  | ACALL | DELAY | Absolute | 2 | 1 |
| 31. |  | ACALL | DELAY | Absolute | 2 | 2 |
| 32. |  | MOV | A,B | Immediate | 2 | 2 |
| 33. |  | MOV | B,#10D | Immediate | 2 | 1 |
| 34. |  | DIV | AB | Absolute | 2 | 2 |
| 35. |  | CLR | P1.0 | Absolute | 2 | 2 |
| 36. |  | SETB | P1.1 | Immediate | 2 | 1 |
| 37. |  | ACALL | DISPLAY | Absolute | 2 | 2 |
| 38. |  | MOV | P0,A | Indirect | 2 | 2 |
| 39. |  | ACALL | DELAY | Absolute | 2 | 1 |
| 40. |  | ACALL | DELAY | Absolute | 2 | 2 |
| 41. |  | MOV | A,B | Indirect | 2 | 2 |
| 42. |  | CLR | P1.1 |  | 2 | 2 |
| 43. |  | SETB | P1.2 | Direct bit | 2 | 1 |
| 44. |  | ACALL | DISPLAY | Immediate | 2 | 1 |
| 45. |  | MOV | P0,A | Immediate | 2 | 2 |
| 46. |  | ACALL | DELAY | Absolute | 2 | 2 |
| 47. |  | ACALL | DELAY | Immediate | 2 | 1 |
| 48. |  | CLR | P1.2 | Absolute | 2 | 2 |
| 49. |  | DJNZ | R5,BACK1 | Absolute | 2 | 2 |
| 50. |  | RET |  |  |  |  |
| 51. | DELAY: | MOV | R7,#250D | Absolute | 2 | 2 |
| 52. | DEL1: | DJNZ | R7,DEL1 | Absolute | 2 | 2 |
| 53. |  | RET |  |  | 2 | 1 |
| 54. | DISPLAY: | MOVC | A,@A+DPTR | Immediate | 2 | 2 |
| 55. |  | CPL | A | Absolute | 2 | 2 |
| 56. |  | RET |  |  | 2 | 1 |
| 57. | LUT: | DB | 3FH | Absolute | 2 | 2 |
| 58. |  | DB | 06H |  | 2 | 2 |
| 59. |  | DB | 5BH |  | 2 | 1 |
| 60. |  | DB | 4FH |  | 2 | 2 |
| 61. |  | DB | 66H |  | 2 | 2 |
| 62. |  | DB | 6DH |  | 2 | 1 |
| 63. |  | DB | 7DH |  | 2 | 2 |
| 64. |  | DB | 07H |  | 2 | 2 |
| 65. |  | DB | 7FH |  | 2 | 1 |
| 66. |  | DB | 6FH |  | 2 | 2 |
| 67. |  | END |  |  |  |  |

**SNAPSHOTS OF THE PROJECT WORKING EXPLANATION STEP BY STEP**

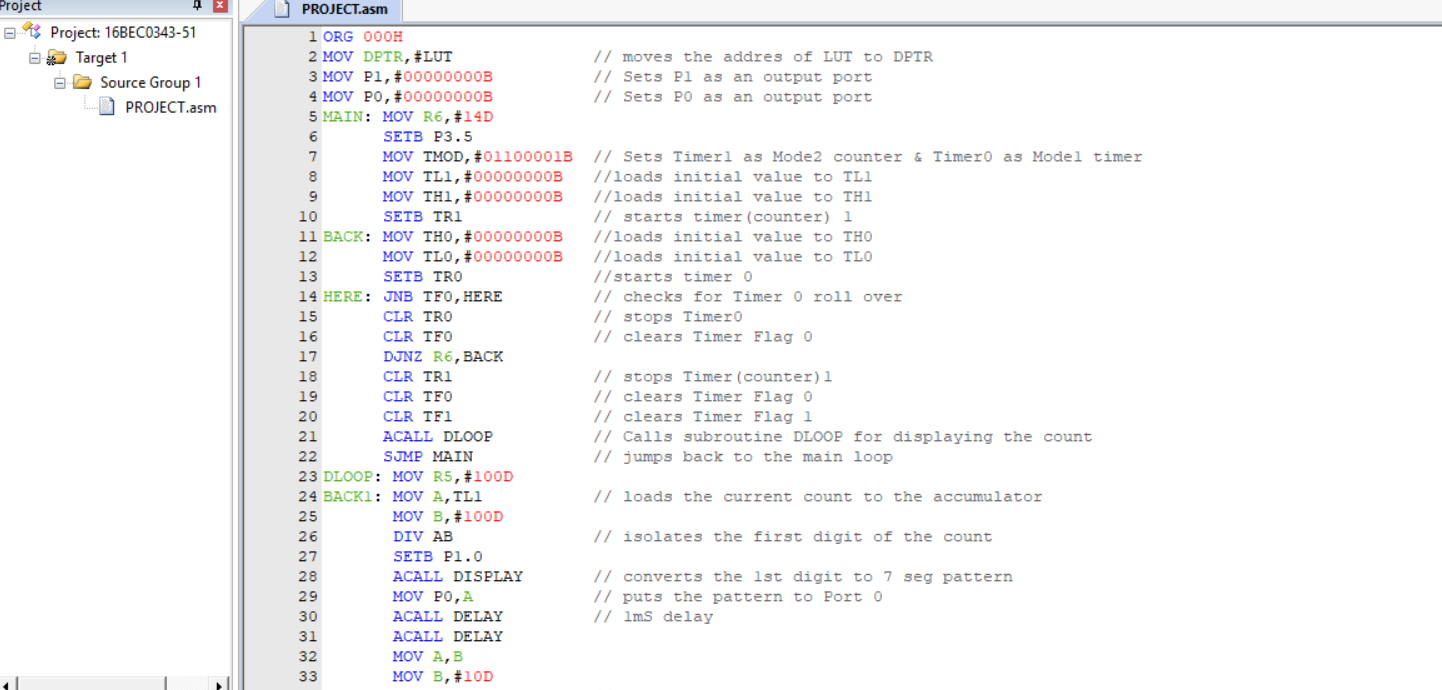
**** The heart component of this project is Ultrasonic Module which associates in measuring the distance between this module and the obstacle that hinders the sensor

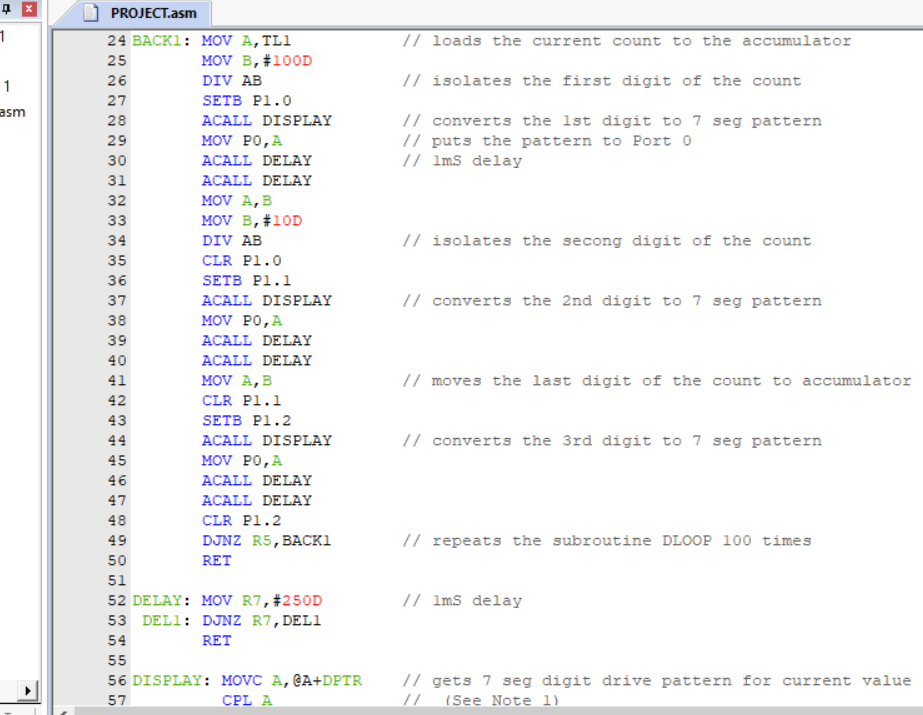
.

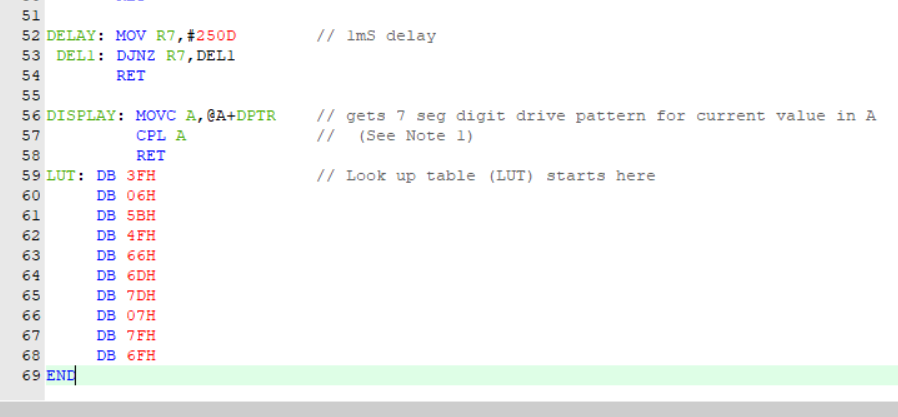
The microcontroller 8051 programmed accordingly in order to convert analog signal of the ultrasonic module to digital signal which is to be displayed in LCD.

Potentiometer here is used to adjust the brightness of the LCD.

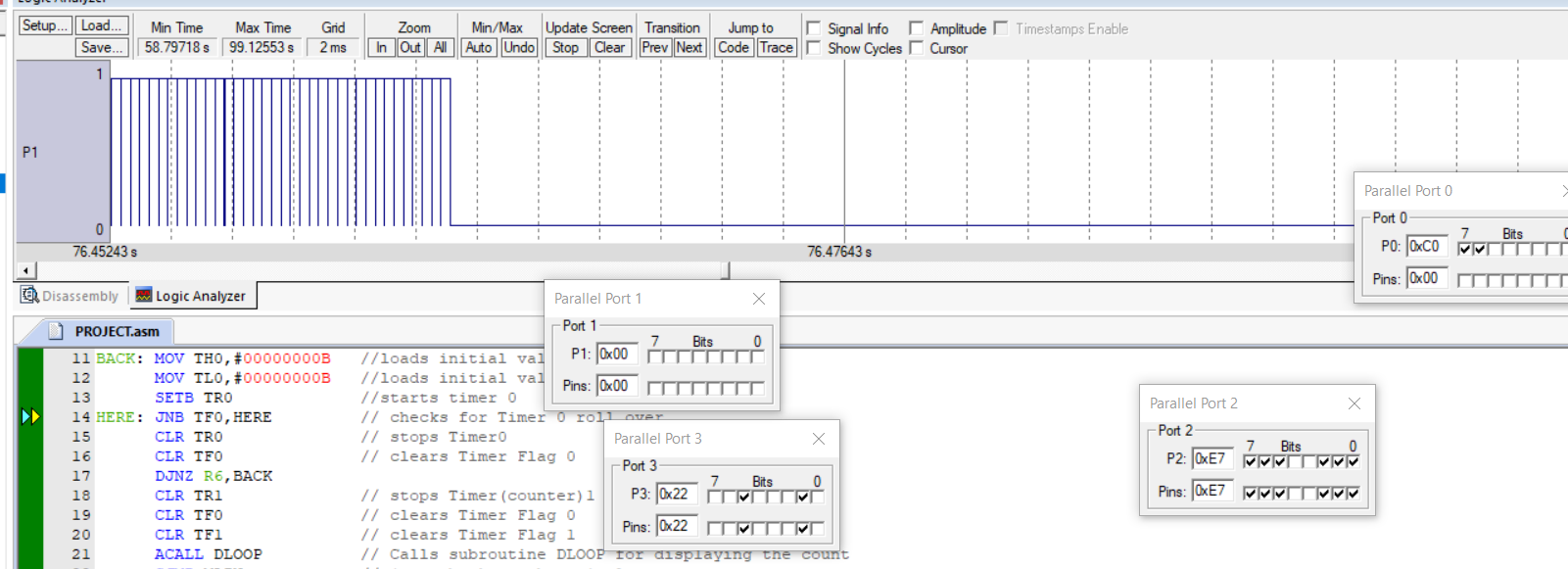
****

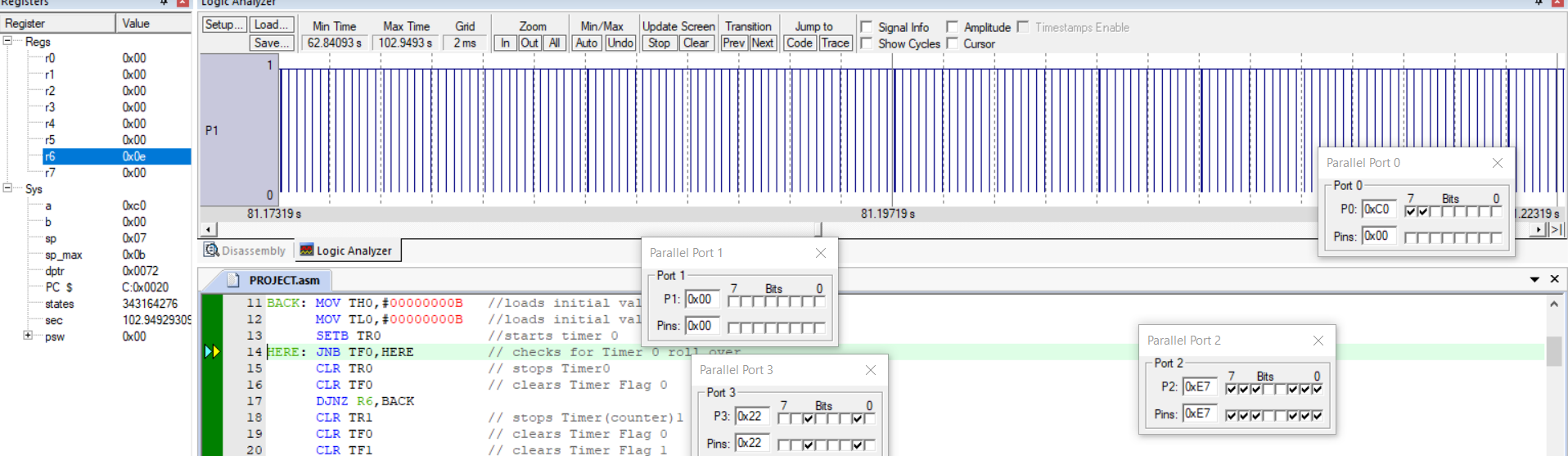
****

****

**OUTPUT**

For software purpose the LCD code has been replaced with the respective ports values. The screenshots of software working have been given below:





**HARDWARE**

****

**APPLICATION**

* RADAR
* **Liquid Level Sensing**
* **Vehicle Detection for Car Washes**
* **Automotive Assembly**
* **Parking Garage Applications**
* **Uses in Production Lines**
* **Interference Detection**

**ADVANTAGES**

1. The ultrasonic sensor has high frequency, high sensitivity and high penetrating power therefore it can easily detect the external or deep objects.
2. These sensors easily interface with 8051 microcontroller.
3. These sensors have greater accuracy then other methods for measuring the thickness and depth of parallel surface.
4. These sensors could easily sense the nature, shape and orientation of that specific objects which is within the area of these sensors.
5. There sensors are easy to use, not dangerous during operation for nearby objects, person, equipment or material.
6. Since the code is given at the microcontroller level, it is easy to change or alter the working according to the user.

**LIMITATIONS:**

* The ICs used in this circuit are CMOS devices and are highly static, making it impossible to touch them with bare hands.
* It has limited life time due to use of battery for powering the circuit.
* Speed calculation may be affected by the varying duty cycle of the timer.

**INFERENCE**

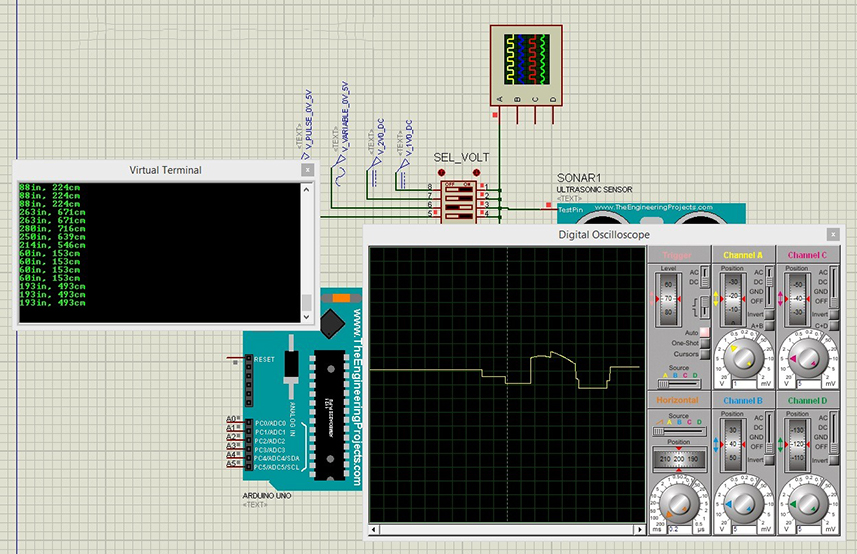
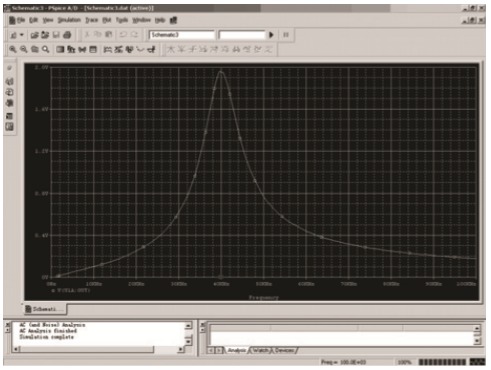
* The distance of obstacle is detected by the ultrasonic sensor module and the data is sent to the microcontroller.
* When the data is collected by the microcontroller this goes through the code

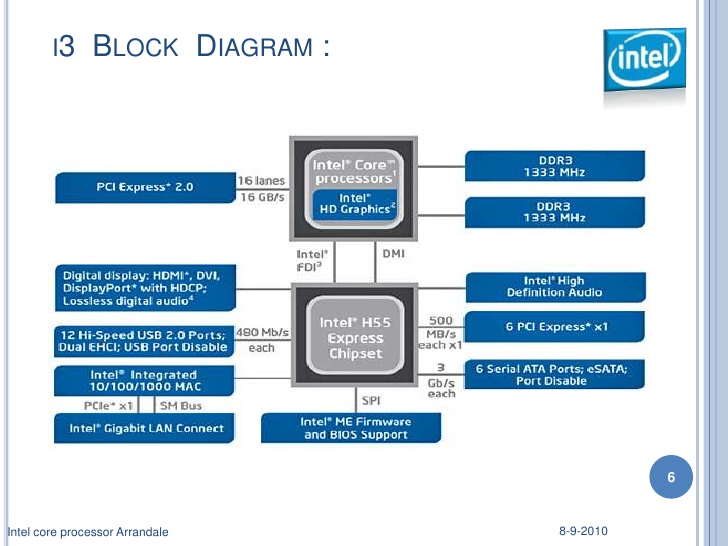
And finds the distance between Ultrasonic module and the obstacle which is thendisplayed by the LCD display.

**RESULT**

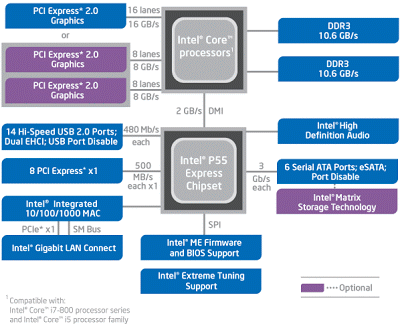
* An assembly level code for the system of ultrasonic range finder has been written and executed in Keil software.
* A hardware model of Ultrasonic Range Finder Circuit has also been implemented.

**Protus Output**

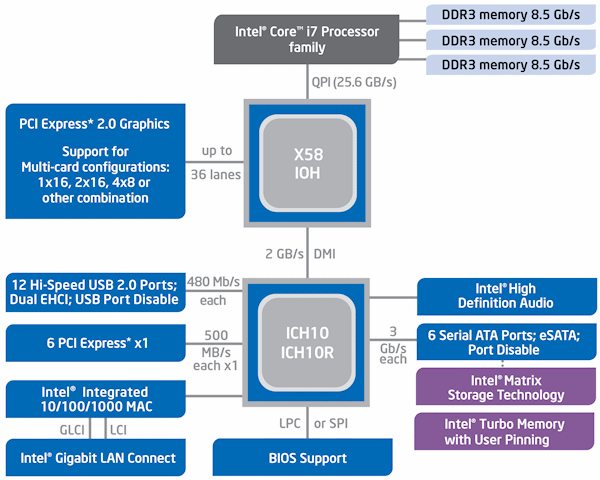
 



Intel I5



Intel I7



Register organization of 8051

