DEPARTMENT OF INFORMATION & COMMUNICATION TECHNOLOGY

MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent College of Manipal University)

MANIPAL – 576104, KARNATAKA, INDIA

November 2015

EMBEDDED SYSTEMS

PROJECT REPORT

B.TECH (IT) FIFTH SEMESTER

8052 ELECTRONIC ORGAN

Submitted by

**Kashish Grover 130911410 (57)**

Under guidance of

Mr. Santhosh Kamath

Contents

[1. Synopsis 3](#_Toc435415441)

[2. Circuit 4](#_Toc435415442)

[3. Program Description 5](#_Toc435415443)

[4. Algorithm 6](#_Toc435415444)

[5. Program Listing 7](#_Toc435415445)

[6. Final Assembly 14](#_Toc435415446)

[7. Conclusion 15](#_Toc435415447)

[8. Cost Details 16](#_Toc435415448)

# Synopsis

This is a simple electronic organ project. The 8052-based microcontroller interfaced with a 4x4 keypad, an LCD and a speaker produces 16 different frequencies. The LCD shows the current note and the frequency. The objective of this project is to have a thorough understanding of interrupt based calculations and the interfacing of the microcontroller with LCD and keyboard.

Components used:

1. AT89S52 Project Board[[1]](#footnote-1)
2. MAX232 USB Programmer1
3. USB Cable1
4. 16x2 LCD Display[[2]](#footnote-2)
5. 4x4 Matrix Keypad
6. Breadboard
7. 22 AWG “Hook-up” wires
8. 10 Jumper Wires
9. Speaker from old headphones
10. Soldering Iron and Solder
11. 3 1kΩ Resistors
12. 1 10kΩ Resistor

Softwares used:

1. ProgISP (Ver 1.72)[[3]](#footnote-3) for burning the HEX files onto the microcontroller
2. MCU 8051 IDE[[4]](#footnote-4) for emulation and debugging of the program
3. Small Device C Compiler (SDCC)[[5]](#footnote-5) for C code compilation

**Block Diagram**

16x2

LCD Display

Speaker

4x4 Keypad

A

T

8

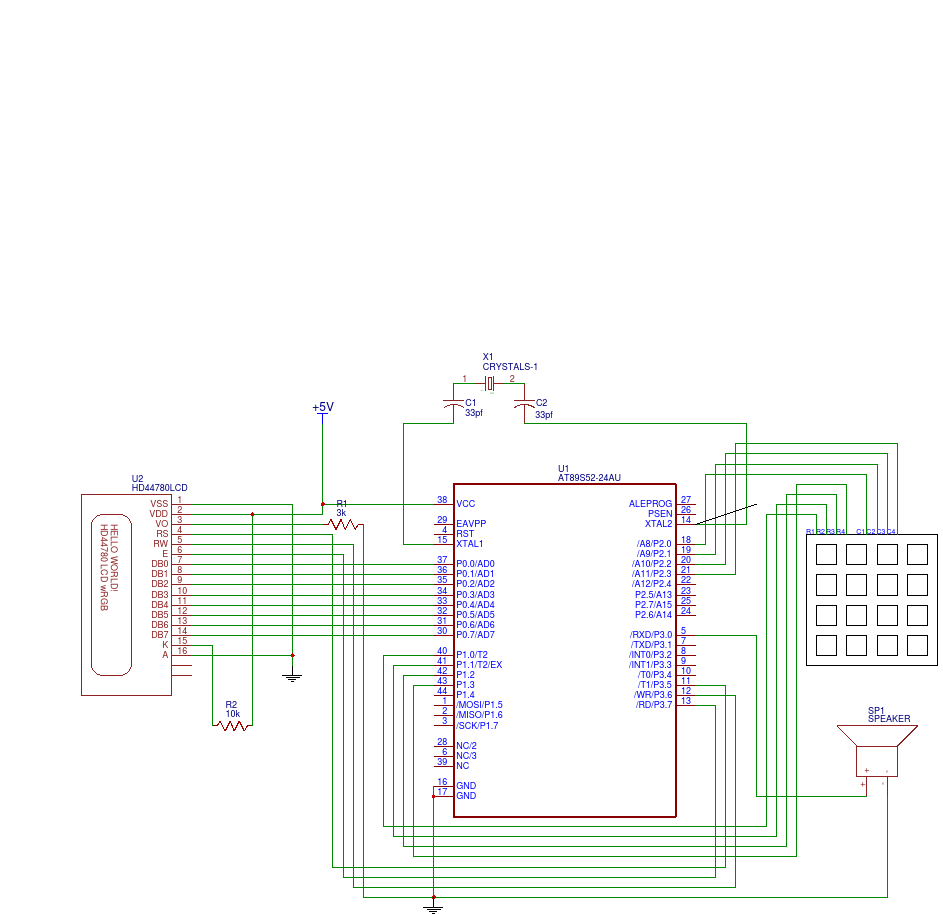
9

S

5

2

# Circuit



P3.0 of the microcontroller connects to the speaker through a transistor. The rows of the keypad connect from P1.0 to P1.3 and the columns from P2.0 to P2.3. The LCD works in 8-bit mode so D0 to D7 of the same connect to P0.0 to P0.7 respectively. RS, R/W and EN of the LCD connect to P3.5, P3.6 and P3.7 respectively. For a decent contrast in the LCD, V0 connects to ground through a 3kΩ resistor.

# Program Description

In this project, Timer-1 generates tones for the electric organ. The timer is in auto reload mode and is loaded to generate an interrupt every 50 µs (i.e. loaded with \_\_\_ for an 11.0592 MHz crystal). The keyboard is scanned in an endless for loop. When it detects a key press, the timer is loaded with the correct value to generate the required tone.

The counter is loaded so that it generates an interrupt every 50 µs. The number of counts (N) required for each note hence obtained by dividing the time-period by 50 µs.

The following table shows the frequencies considered:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Note** | **Octave** | **Frequency (f Hz)** | **Time Period**  **(T µs =106/f)** | **N**  **(=T/50)** |
| **1** | C | 4 | 523.35 | 1910.77 | 76.45 (~ 76) |
| **2** | C# | 4 | 277.1826 | 3607.73 | 72.15 (~ 72) |
| **3** | D | 4 | 293.6648 | 3405.24 | 68.10 (~ 68) |
| **4** | D# | 4 | 311.1270 | 3214.12 | 64.28 (~ 64) |
| **5** | E | 4 | 329.6276 | 3033.73 | 60.67 (~ 61) |
| **6** | F | 4 | 349.2282 | 2863.46 | 57.27 (~ 57) |
| **7** | F# | 4 | 369.9944 | 2702.74 | 54.05 (~ 54) |
| **8** | G | 4 | 391.9954 | 2551.05 | 51.02 (~ 51) |
| **9** | G# | 4 | 415.3074 | 2407.86 | 48.16 (~ 48) |
| **10** | A | 4 | 440 | 2272.73 | 45.45 (~ 45) |
| **11** | A# | 4 | 466.1638 | 2145.17 | 42.90 (~ 43) |
| **12** | B | 4 | 493.8833 | 2024.77 | 40.50 (~ 41) |
| **13** | C | 5 | 523.2511 | 1911.13 | 38.22 (~ 38) |
| **14** | C# | 5 | 554.3653 | 1803.86 | 36.08 (~ 36) |
| **15** | D | 5 | 587.3295 | 1702.62 | 34.05 (~ 34) |
| **16** | D# | 5 | 622.2540 | 1607.06 | 32.14 (~ 32) |

# Algorithm

The following algorithm describes the working of the program.

**START**

Turn off Speaker

Initialize the LCD

Initialize timer1

Initialize *count*

**DO FOREVER**

**IF** a key is pressed **THEN**

Load the value of the key in variable *tone*

Display the values on the LCD

**ENDIF**

**ENDDO**

**END**

*Timer 1 interrupt service routine*

**START**

Increment *count*

**IF** *count* == *tone* **THEN**

*count* = 0

Complement speaker output

**ENDIF**

**END**

# Program Listing

*The following program was written in MCU 8051 IDE with SDCC.*

*Date of final build – 15/11/2015*

#include<at89x52.h>

#define COL P2

#define ROW P1

\_\_sbit \_\_at **(**0xB0**)** SPEAKER**;** // P3.0

\_\_sbit \_\_at **(**0xB5**)** RS**;** // P3.5

\_\_sbit \_\_at **(**0xB6**)** RW**;** // P3.6

\_\_sbit \_\_at **(**0xB7**)** EN**;** // P3.7

\_\_sbit \_\_at **(**0x80**)** D0**;** // P0\_0

\_\_sbit \_\_at **(**0x81**)** D1**;** // P0\_1

\_\_sbit \_\_at **(**0x82**)** D2**;** // P0\_2

\_\_sbit \_\_at **(**0x83**)** D3**;** // P0\_3

\_\_sbit \_\_at **(**0x84**)** D4**;** // P0\_4

\_\_sbit \_\_at **(**0x85**)** D5**;** // P0\_5

\_\_sbit \_\_at **(**0x86**)** D6**;** // P0\_6

\_\_sbit \_\_at **(**0x87**)** D7**;** // P0\_7

int count**,**tone**;**

unsigned char colloc**;**

int col**,**row**,**flag**=**0**;**

int keypad**[**4**][**4**]** **=** **{{**76**,**72**,**68**,**64**},**

**{**61**,**57**,**54**,**51**},**

**{**48**,**45**,**43**,**41**},**

**{**38**,**36**,**34**,**32**}};**

const char note**[**5**][**5**][**7**]** **=** **{{**"C4 "**,**"C#4"**,**"D4 "**,**"D4 "**,**"D#4"**},**

**{**"E4 "**,**"F4 "**,**"F#4"**,**"F#4"**,**"G4 "**},**

**{**"G#4"**,**"A4 "**,**"A#4"**,**"A#4"**,**"B4 "**},**

**{**"C5 "**,**"C#5"**,**"D5 "**,**"D5 "**,**"D#5"**}};**

void Lcd\_Delay**(**int**);**

void Lcd8\_Port**(**char**);**

void Lcd8\_Cmd**(**char**);**

void Lcd8\_Clear**();**

void Lcd8\_Set\_Cursor**(**char a**,** char b**);**

void Lcd8\_Init**();**

void Lcd8\_Write\_Char**(**char a**);**

void Lcd8\_Write\_String**(**char **\***a**);**

void MSDelay**(**int value**);**

//Timer1 initialization routine

void init\_timer**()**

**{**

ET1**=**1**;**

TMOD**=**0x20**;**

TH1**=**210**;**

TR1**=**1**;**

EA**=**1**;**

**}**

void timer1**()** \_\_interrupt 3

**{**

count**++;**

**if(**count**==**tone**)**

**{**

count**=**0**;**

SPEAKER**^=**1**;**

**}**

**if(**count**==**150**)**

count**=**0**;**

**}**

void main**()**

**{**

count**=**0**;**

SPEAKER**=**1**;**

init\_timer**();**

Lcd8\_Init**();**

Lcd8\_Clear**();**

Lcd8\_Set\_Cursor**(**1**,**0**);**

Lcd8\_Write\_String**(**" KASHISH GROVER "**);**

Lcd\_Delay**(**5000**);**

Lcd8\_Set\_Cursor**(**2**,**0**);**

Lcd8\_Write\_String**(**" 130911410 IT-A "**);**

Lcd\_Delay**(**5000**);**

Lcd8\_Clear**();**

Lcd8\_Set\_Cursor**(**1**,**4**);**

Lcd8\_Write\_String**(**"Note: "**);**

COL**=**0xFF**;**

**while(**1**)**

**{**

**while(**1**)**

**{**

**if(**flag**==**1**)**

**{**

Lcd8\_Set\_Cursor**(**1**,**10**);**

Lcd8\_Write\_String**(**note**[**row**][**col**]);**

Lcd\_Delay**(**200**);**

**}**

ROW**=**0XFE**;**

colloc**=**COL**;**

colloc**&=**0x0F**;**

**if(**colloc**!=**0x0F**)**

**{**

TR1**=**1**;**

row**=**0**;**

**break;**

**}**

ROW**=**0XFD**;**

colloc**=**COL**;**

colloc**&=**0x0F**;**

**if(**colloc**!=**0x0F**)**

**{**

TR1**=**1**;**

row**=**1**;**

**break;**

**}**

ROW**=**0XFB**;**

colloc**=**COL**;**

colloc**&=**0x0F**;**

**if(**colloc**!=**0x0F**)**

**{**

TR1**=**1**;**

row**=**2**;**

**break;**

**}**

ROW**=**0XF7**;**

colloc**=**COL**;**

colloc**&=**0x0F**;**

**if(**colloc**!=**0x0F**)**

**{**

TR1**=**1**;**

row**=**3**;**

**break;**

**}** **}**

**if(**colloc**==**0x0E**)**

**{**

col**=**0**;**

tone **=** keypad**[**row**][**0**];**

flag**=**1**;**

**}**

**else** **if(**colloc**==**0x0D**)**

**{**

col**=**1**;**

tone **=** keypad**[**row**][**1**];**

flag**=**1**;**

**}**

**else** **if(**colloc**==**0x0B**)**

**{**

col**=**3**;**

tone **=** keypad**[**row**][**2**];**

flag**=**1**;**

**}**

**else** **if(**colloc**==**0x07**)**

**{**

col**=**4**;**

tone **=** keypad**[**row**][**3**];**

flag**=**1**;**

**}**

**else** //No key is pressed

**{**

SPEAKER**=**0**;**

count**=**0**;**

TR1**=**0**;**

**}**

**}**

**}**

void Lcd\_Delay**(**int a**)**

**{**

int j**;**

int i**;**

**for(**i**=**0**;**i**<**a**;**i**++)**

**{**

**for(**j**=**0**;**j**<**10**;**j**++)**

**{**

**}**

**}**

**}**

//LCD 8 Bit Interfacing Functions

void Lcd8\_Port**(**char a**)**

**{**

**if(**a **&** 1**)**

D0 **=** 1**;**

**else**

D0 **=** 0**;**

**if(**a **&** 2**)**

D1 **=** 1**;**

**else**

D1 **=** 0**;**

**if(**a **&** 4**)**

D2 **=** 1**;**

**else**

D2 **=** 0**;**

**if(**a **&** 8**)**

D3 **=** 1**;**

**else**

D3 **=** 0**;**

**if(**a **&** 16**)**

D4 **=** 1**;**

**else**

D4 **=** 0**;**

**if(**a **&** 32**)**

D5 **=** 1**;**

**else**

D5 **=** 0**;**

**if(**a **&** 64**)**

D6 **=** 1**;**

**else**

D6 **=** 0**;**

**if(**a **&** 128**)**

D7 **=** 1**;**

**else**

D7 **=** 0**;**

**}**

void Lcd8\_Cmd**(**char a**)**

**{**

RS **=** 0**;** // => RS = 0

Lcd8\_Port**(**a**);** //Data transfer

EN **=** 1**;** // => E = 1

Lcd\_Delay**(**5**);**

EN **=** 0**;** // => E = 0

**}**

void Lcd8\_Clear**()**

**{**

Lcd8\_Cmd**(**1**);**

**}**

void Lcd8\_Set\_Cursor**(**char a**,** char b**)**

**{**

**if(**a **==** 1**)**

Lcd8\_Cmd**(**0x80 **+** b**);**

**else** **if(**a **==** 2**)**

Lcd8\_Cmd**(**0xC0 **+** b**);**

**}**

void Lcd8\_Init**()**

**{**

Lcd8\_Port**(**0x00**);**

RS **=** 0**;**

Lcd\_Delay**(**200**);**

///////////// Reset process from datasheet /////////

Lcd8\_Cmd**(**0x30**);**

Lcd\_Delay**(**50**);**

Lcd8\_Cmd**(**0x30**);**

Lcd\_Delay**(**110**);**

Lcd8\_Cmd**(**0x30**);**

/////////////////////////////////////////////////////

Lcd8\_Cmd**(**0x38**);** //function set

Lcd\_Delay**(**50**);**

Lcd8\_Cmd**(**0x0C**);** //display on,cursor off,blink off

Lcd\_Delay**(**50**);**

Lcd8\_Cmd**(**0x01**);** //clear display

Lcd\_Delay**(**1000**);**

Lcd8\_Cmd**(**0x06**);** //entry mode, set increment

Lcd\_Delay**(**50**);**

**}**

void Lcd8\_Write\_Char**(**char a**)**

**{**

RS **=** 1**;** // => RS = 1

Lcd8\_Port**(**a**);** //Data transfer

EN **=** 1**;** // => E = 1

Lcd\_Delay**(**5**);**

EN **=** 0**;** // => E = 04

**}**

void Lcd8\_Write\_String**(**char **\***a**)**

**{**

int i**;**

**for(**i**=**0**;**a**[**i**]!=**'\0'**;**i**++)**

Lcd8\_Write\_Char**(**a**[**i**]);**

**}**

void MSDelay**(**int value**)**

**{**

int x**;**

int y**;**

**for(**x**=**0**;**x**<**1275**;**x**++)**

**{**

**for(**y**=**0**;**y**<**value**;**y**++)**

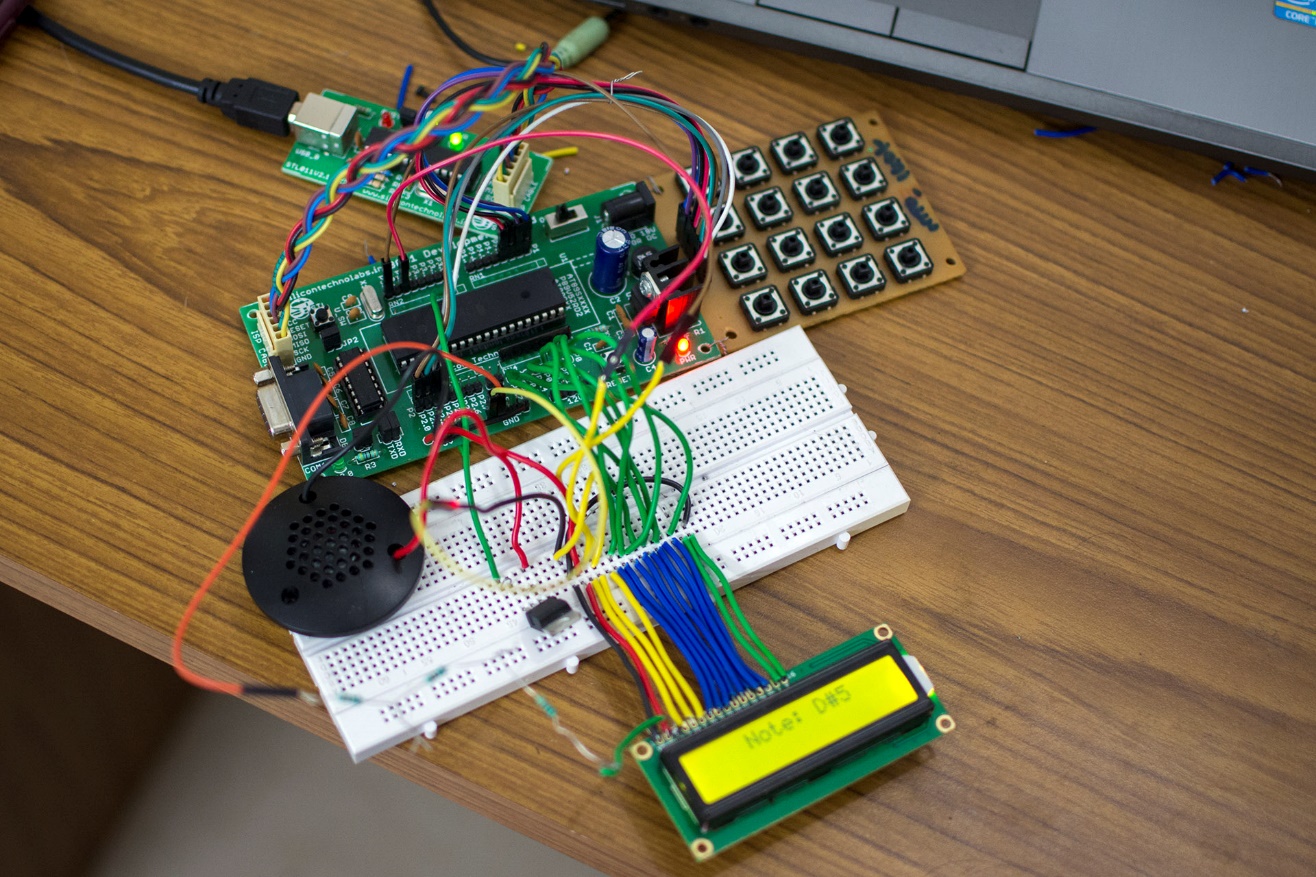
**{**

**}**

**}**

**}**

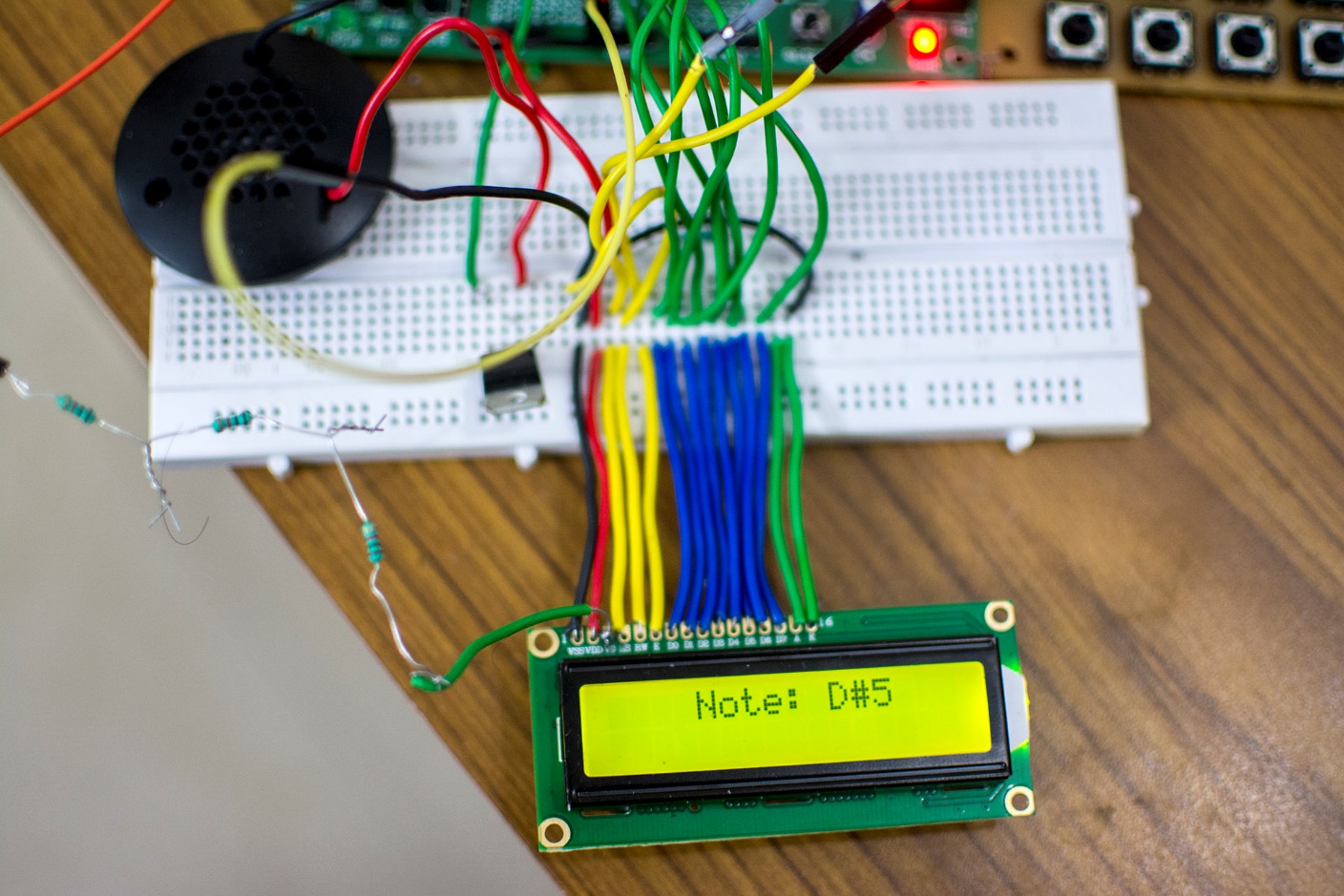
# Final Assembly



# Conclusion

This project was successfully completed and it helped me understand many concepts related to programming of 8051 family of microcontrollers, the HD44780 LCD, 4x4 Keypad and timer based interrupt calculations.

All 16 frequencies were generated with great accuracy. This was confirmed using smartphone-based tuners. The LCD also worked perfectly with the correct note being displayed for each key press.



# Cost Details

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Component** | **Price** |
| **1** | AT89S52 Project Board, MAX232 USB Programmer, USB Cable | ₹ 650/- |
| **2** | 16x2 LCD Display | ₹ 200/- |
| **3** | 4x4 Matrix Keypad | ₹ 40/- |
| **4** | Breadboard | ₹ 80/- |
| **5** | 22 AWG “Hook-up” Wires | ₹ 30/- |
| **6** | 10 Jumper Wires | ₹ 40/- |
| **7** | Speaker | Old headphones |
| **8** | Soldering Iron and Solder | Borrowed |
| **9** | Resistors | ₹ 5/- |
| **10** | Transistor | ₹ 20/- |
| **TOTAL** | | **₹ 1065/-** |

1. <http://www.ebay.in/itm/201416474873?ssPageName=STRK:MEWNX:IT&_trksid=p3984.m1439.l2649> [↑](#footnote-ref-1)
2. <http://www.ebay.in/itm/231643566013?ssPageName=STRK:MEWNX:IT&_trksid=p3984.m1439.l2649> [↑](#footnote-ref-2)
3. <http://www.electrodragon.com/w/ProgISP> [↑](#footnote-ref-3)
4. <http://www.moravia-microsystems.com/mcu-8051-ide/> [↑](#footnote-ref-4)
5. <http://sdcc.sourceforge.net/> [↑](#footnote-ref-5)