Digital Clock:

A Digital clock is a type of clock that displays the time digitally. A Clock or watch in which the hours, minutes, and seconds are indicated by digits, rather than by hands on a dial compare to analogue clock.

Introduction :

- A Digital clock displays the current time.
- It display the DATE digitally (in numerals) in as DD:MM:YYYY formate.
- It display the time digitally (in numerals) in 12 hour formate as HH:MM:SS.
- Digital clocks are more accurate than analog clocks.
- Human can easily notify the time by using digital clock is better than Analogue clock.

PROGRAMME:

// Embedded C Programme for digital clock

```
#include <REGX51.H>
#include <stdio.h>
#define LCD_dat P2
sbit rs = P1^0;
sbit en = P1^2;
sbit one = P1^3;
sbit two = P1^4;
sbit three = P1^5;
sbit fin = P1^6;
sbit fin1 = P1^7;
void delay(unsigned int dly); // FUNCTION TO GENERATE DELAY
void lcd_cmd(unsigned char ch); // FUNCTION TO SEND COMMANDS TO LCD
void lcd_data(unsigned char ch); // FUNCTION TO SEND DATA TO LCD
void lcd_str(unsigned char *str); // FUNCTION TO SEND STRING TO LCD
void to_char(unsigned int value) // FUNCTION TO CONVERT INTEGER TO ASCII VALUE
                                 {char tens,units;
                                  tens=value/10;
```

```
                        lcd_data(tens+48);
                        units=value%10;
                       lcd_data(units+48);}
void main(void) { // MAIN FUNCTION
  unsigned int digit = 0;
  unsigned int j,k;
  unsigned int date = 0;
        unsigned int mon = 0;
        unsigned int year = 22;
        unsigned int hrs = 0;
        unsigned int min = 0;
        unsigned int sec = 0;
        unsigned char am1[2]="am";
        unsigned char pm1[2]="pm";
        unsigned int am = 1;
        lcd_cmd (0x38);
        lcd_cmd (0x0e);
        lcd_cmd (0x80);
        lcd_str("Date ");
        lcd_cmd(0xc0);
        lcd_cmd(0x0E);
        lcd_str("Time ");
        while(1) // INFINITE LOOP
              {++ sec;
                    if(sec>59){sec=0;
                              ++min;}
                    if(min>59){min=0;
                               ++hrs;}
                    if(hrs>12){hrs=1;
                              if(am==1){am=0;}
                              if(am==0){++date;
                                        am=1;}}
                    if(date>30){date=0;
                               ++mon;}
                    if(mon>12){mon=0;
                               ++year;}
        for(j=0;fin1==0;j++) // TO SET DATE/MONTH/YEAR
              {if (one == 0){if(date<=30){++date;}
                    else if(date>30) {date=1;}
        delay(500);}
               if (two == 0){if(mon<=11){++mon;}
                          else if(mon>11) {mon=1;}
                          delay(500); }
              if (three == 0){if(year<=98){++year;}
                           else if(year>98) {year=0;}
                           delay(500); }}
        for(j=0;fin==0;j++) // TO SET HOURS/MINUTES/(AM/PM)
        {if (one == 0){if(hrs<=11){++hrs;}
                    else if(hrs>11) {hrs=1;}
                    delay(500);}
        if (two == 0){if(min<=58){++min;}
                    else if(min>58) {min=0;}
                    delay(500); }
        if (three == 0){   am=0;
                    delay(500);
                   if (three == 0){am=1;} }}
        lcd_cmd(0x85); // TO PRINT ON DISPLAY
```

```
        to_char(date);
                        lcd_cmd(0x87);
                        lcd_data('/');
                        lcd_cmd(0x88);
                        to_char(mon);
                        lcd_cmd(0x8a);
                        lcd_data('/');
                        lcd_cmd(0x8b);
                        lcd_data('2');
                        lcd_cmd(0x8c);
                        lcd_data('0');
                        lcd_cmd(0x8d);
                        to_char(year);
                        lcd_cmd(0x8f);
                        lcd_data(' ');
                        lcd_cmd(0xc5);
                        to_char(hrs);
                        lcd_cmd(0xc7);
                        lcd_data(':');
                        lcd_cmd(0xc8);
                        to_char(min);
                        lcd_cmd(0xca);
                        lcd_data(':');
                        lcd_cmd(0xcb);
                        to_char(sec);
                        lcd_data(' ');
                        if(am==1){lcd_str(am1);}
                        if(am==0){lcd_str(pm1);}
                        delay(590);}  }
void lcd_str(unsigned char *str){ unsigned int loop = 0;
                                for(loop =0;str[loop]!='\0';loop++)
                                { lcd_data(str[loop]); }}
void lcd_data(unsigned char ch){LCD_dat = ch;
                                        rs =1;
                                        en = 1;
                                        delay(5);
                                        en = 0;}
void lcd_cmd(unsigned char ch){LCD_dat = ch;
                                        rs = 0;
                                        en = 1;
                                        delay(5);
                                        en = 0;}
void delay(unsigned int dly){unsigned int loop = 0;
                            unsigned int delay_gen = 0;
                for (loop=0; loop<dly; loop++){ for (delay_gen=0; delay_gen<115; delay_gen++);}}
```

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU - 572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Project Report on

"DIGITAL CLOCK"

BACHELOR OF ENGINEERING
in
ELECTRONICS & COMMUNICATION ENGINEERING

Submitted by

Suman H V	1SI20EC096
Nithesh Kumar V	1SI20EC122
Mithun J B	1SI20EC124
G Vivek	1SI20EC125
Rakshith	1SI20EC128

Submitted to

Dr. Seema B Hegde

Professor

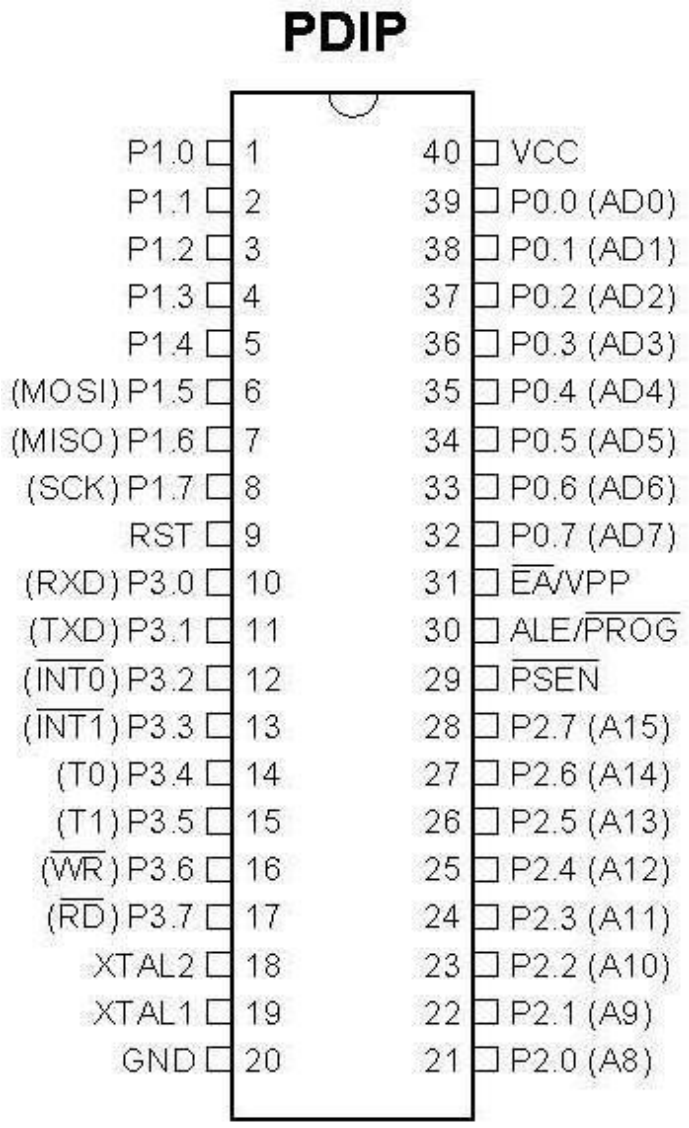
Department of E&C Engineering SIT,

Tumakuru – 572103

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING 2021-22

8051 (AT89S51 24PU):

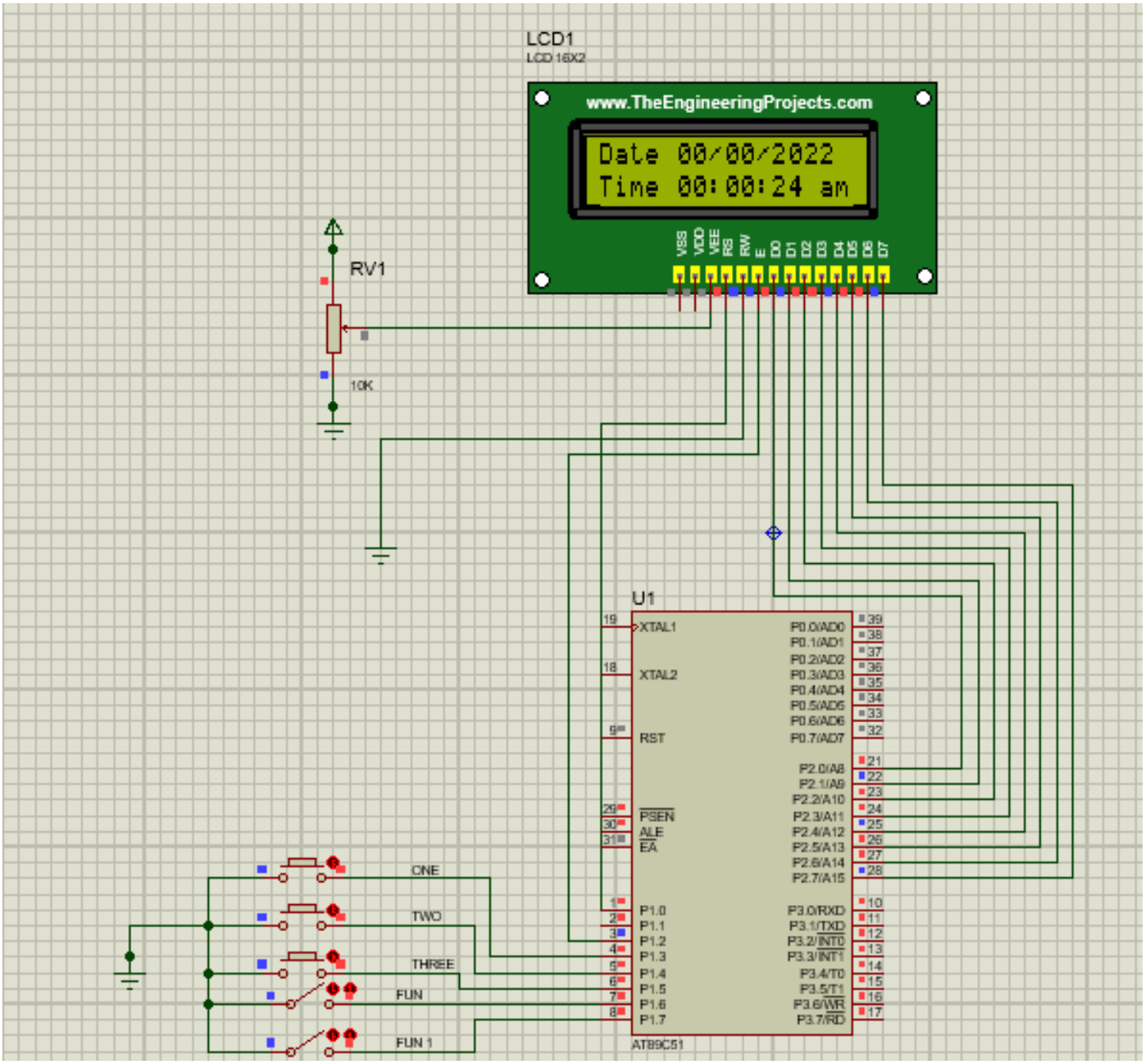
PIN DIAGRAM:

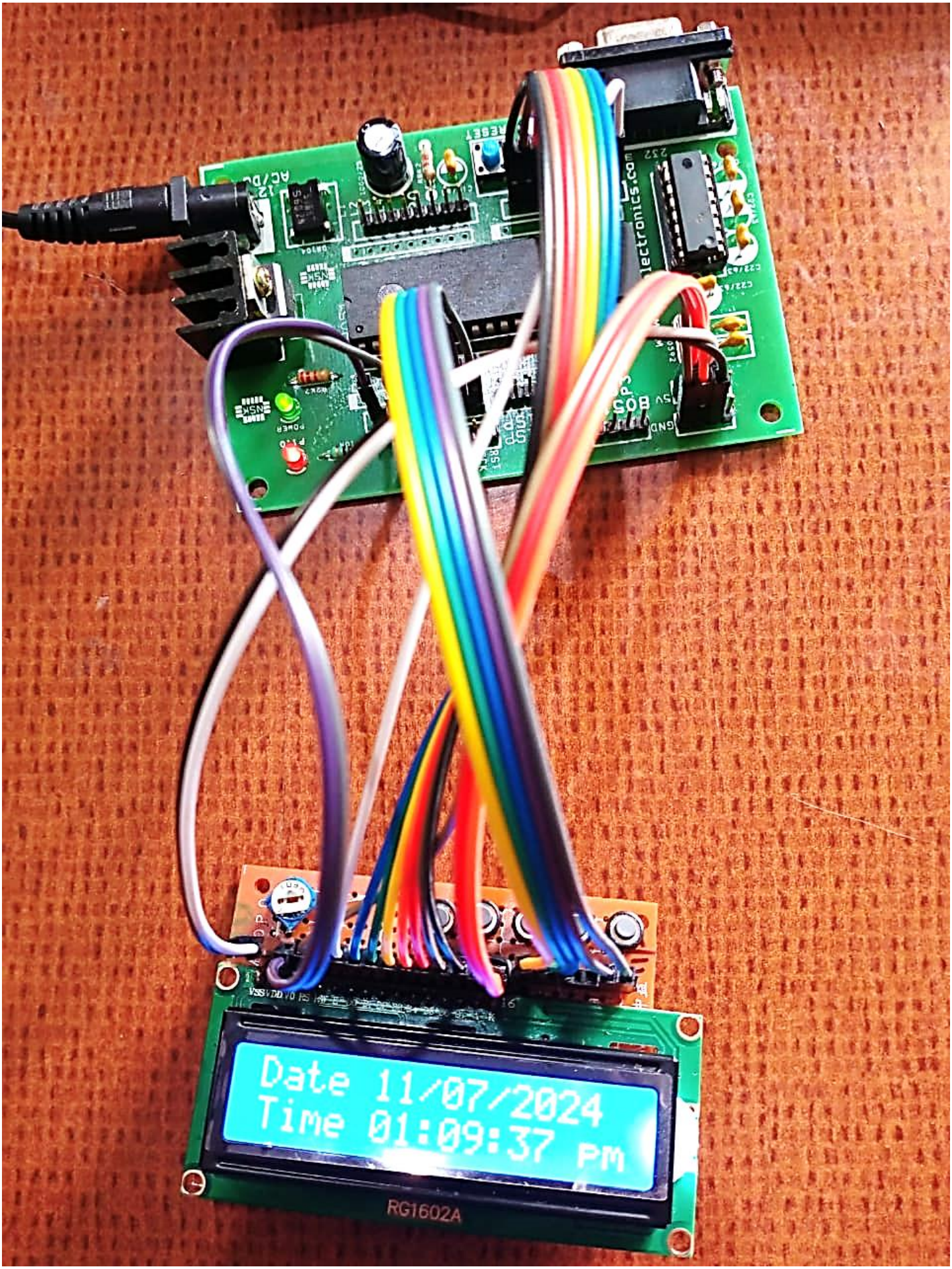


Specifications:

Product Attribute	Attribute Value
Manufacturer:	Microchip
Product Category:	8-bit Microcontrollers - MCU
RoHS:	Details
Series:	89S
Mounting Style:	Through Hole
Package/Case:	PDIP-40
Core:	8051
Program Memory Size:	4 kB
Data Bus Width:	8 bit
ADC Resolution:	No ADC
Maximum Clock Frequency:	24 MHz
Number of I/Os:	32 I/O
Data RAM Size:	128 B
Supply Voltage - Min:	4 V
Supply Voltage - Max:	5.5 V
Minimum Operating Temperature:	- 40 C
Maximum Operating Temperature:	+ 85 C
Packaging:	Tube
Brand:	Microchip Technology / Atmel
Height:	4.83 mm
Interface Type:	UART
Length:	52.58 mm
Number of Timers/Counters:	2 Timer
Operating Supply Voltage:	4 V to 5.5 V
Processor Series:	AT89x
Product:	MCU
Product Type:	8-bit Microcontrollers - MCU
Program Memory Type:	Flash
Factory Pack Quantity:	10
Subcategory:	Microcontrollers - MCU
Width:	13.97 mm
Unit Weight:	6 g

Circuit Diagram:




Date 11/07/2024
Time 01:09:37 PM
RG1602A

LCD Display Module 16X2

	Pin Name	Description
1	Vss (Ground)	VSS pin connected to microcontroller ground
2	Vdd (+5 Volt)	VDD pin connected to microcontroller + 5V power supply
3	VE (Contrast V)	Adjusts the contrast of the LCD display. It is Connected to a variable POT that can provide 0-5V power supply. Connect it to the ground to get maximum contrast.
4	RS (Register Select)	Toggles between Command/Data Register. Connect a microcontroller data pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
5	RW (Read/Write)	Used to read or write data. Normally grounded to write data to LCD
6	E (Enable)	This pin should be held high to execute the Read/Write

		process, and it is connected to the microcontroller data pin & constantly held high.
7	D0 (Data Pin 0)	These 8 Pins are used to sending commands or data to the LCD. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller data pin 0 to 3. And in 8-wire mode, 8-pins are connected to microcontroller data pin 0 to 7.
8	D1 (Data Pin 1)	
9	D2 (Data Pin 2)	
10	D3 (Data Pin 3)	
11	D4 (Data Pin 4)	
12	D5 (Data Pin 5)	
13	D6 (Data Pin 6)	
14	D7 (Data Pin 7)	
15	LED + (+5V)	This is the positive terminal of the backlight LED of the display. It’s connected to +5V to turn on the backlight LED.
16	LED – (Ground)	This is the negative terminal of the backlight LED of the display. It’s connected to the ground to turn on the backlight LED.

Some LCD Commands

No	HEX Value	COMMAND TO LCD
1	0x01	Clear Display Screen
2	0x30	Function Set: 8-bit, 1 Line, 5x7 Dots
3	0x38	Function Set: 8-bit, 2 Line, 5x7 Dots
4	0x20	Function Set: 4-bit, 1 Line, 5x7 Dots
5	0x28	Function Set: 4-bit, 2 Line, 5x7 Dots
6	0x06	Entry Mode
7	0x08	Display off, Cursor off
8	0x0E	Display on, Cursor on
9	0x0C	Display on, Cursor off
10	0x0F	Display on, Cursor blinking
11	0x18	Shift entire display left
12	0x1C	Shift entire display right
13	0x10	Move cursor left by one character
14	0x14	Move cursor right by one character
15	0x80	Force cursor to beginning of 1st row
16	0xC0	Force cursor to beginning of 2nd row

ASCII Table

```
cook@pop-os:~$ ascii -d
   0 NUL     16 DLE     32        48 0     64 @     80 P     96 `    112 p
   1 SOH     17 DC1     33 !      49 1     65 A     81 Q     97 a    113 q
   2 STX     18 DC2     34 "      50 2     66 B     82 R     98 b    114 r
   3 ETX     19 DC3     35 #      51 3     67 C     83 S     99 c    115 s
   4 EOT     20 DC4     36 $      52 4     68 D     84 T    100 d    116 t
   5 ENQ     21 NAK     37 %      53 5     69 E     85 U    101 e    117 u
   6 ACK     22 SYN     38 &      54 6     70 F     86 V    102 f    118 v
   7 BEL     23 ETB     39 '      55 7     71 G     87 W    103 g    119 w
   8 BS      24 CAN     40 (      56 8     72 H     88 X    104 h    120 x
   9 HT      25 EM      41 )      57 9     73 I     89 Y    105 i    121 y
  10 LF      26 SUB     42 *      58 :     74 J     90 Z    106 j    122 z
  11 VT      27 ESC     43 +      59 ;     75 K     91 [    107 k    123 {
  12 FF      28 FS      44 ,      60 <     76 L     92 \    108 l    124 |
  13 CR      29 GS      45 -      61 =     77 M     93 ]    109 m    125 }
  14 SO      30 RS      46 .      62 >     78 N     94 ^    110 n    126 ~
  15 SI      31 US      47 /      63 ?     79 O     95 _    111 o    127 DEL
```

Logic to Convert Integer to Character:

```
void to_char(unsigned int value)
 // FUNCTION TO CONVERT INTEGER TO ASCII VALUE
                {char tens,units;
                 tens=value/10;
                lcd_data(tens+48);
                units=value%10;
                        lcd_data(units+48);}
```