***Abstract:***

1. Introduction

At this time most peoples in the whole world use an automated digital clock in their everyday

use. Starting from the hand watch we were to those huge street clocks every one of us are

dependent on the display the make.

In 21th century time being more than money, regarding this change our hobbies of checking our time every minute is dramatically increasing. About 99% of today’s digital clocks are made using microcontrollers which make them more hand able from the rest, those we can set the time to start any minute or second we want and also set an alarm for reminder so that the system will store the value in a memory and then when the time reaches the alarm will be on. As the microcontroller consists almost all the logical devices external logic gates doesn’t exist.

In order to be used properly and for a long life usage digital clocks must cover a very small place as much as it could but the size of most of the digital clocks manufactured this time is

unexpectedly increasing as the use the give increases. This are the list of problems that exists in today’s digital clocks

****

**CONTENT:**

**1. INTRODUCTION**

**1.a Definition 4**

**1.b Objectives**

**1.c Significance of the project**

**1.d Scope of the project**

**2. MICROCONTROLLER AND ITS SPECS**

**2.a 8051 Microcontroller 5**

**2.b Pin Diagram of 8051 Microcontroller**

**2.c Specifications of 8051 Microcontroller 6**

**2.d Block Diagram of 8051 Microcontroller 7**

**3. CIRCUIT DIAGRAM**

**3.a Circuit Diagram using Proteus simulator 8**

**3.b Circuit Diagrams using Hardware Components 9**

**4. REQUIREMENTS**

**4.a Atmel 89s51 Microcontroller 10**

**4.b Development Board**

**4.c 16x2 LCD Display 11**

**4.d Jumper Wires 13**

**4.e 2 Pin Push buttons**

**4.f Power Adapter**

**5. FLOWCHART**

**5.a Flowchart of Digital Clock 14**

**6. EMBEDDED C PROGRAM FOR OPERATION**

**6.a Programme 15**

**7. RESULTS**

**7.a Results Using Proteus Simulator 19**

**7.b Results Using Hardware 21**

**8. REFERENCES 21**

**INTRODUCTION**

**1.a Definition of 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.

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

analogue clock.

**1.b Objectives:**

Our proposed system will be using a JK flip flop to make the synchronous counter that counts the hour, minute and second and also uses three push buttons to set those outputs.

**1.c Significance of the project:**

Digital clocks are being a very useful components of our lives. Regarding this change the need of accurate and simple materials also dramatically increasing. Our proposed project uses a very simple logic devices to build an accurate synchronous digital clock that is expected to satisfy the need of those materials.

**1.d Scope of the project:**

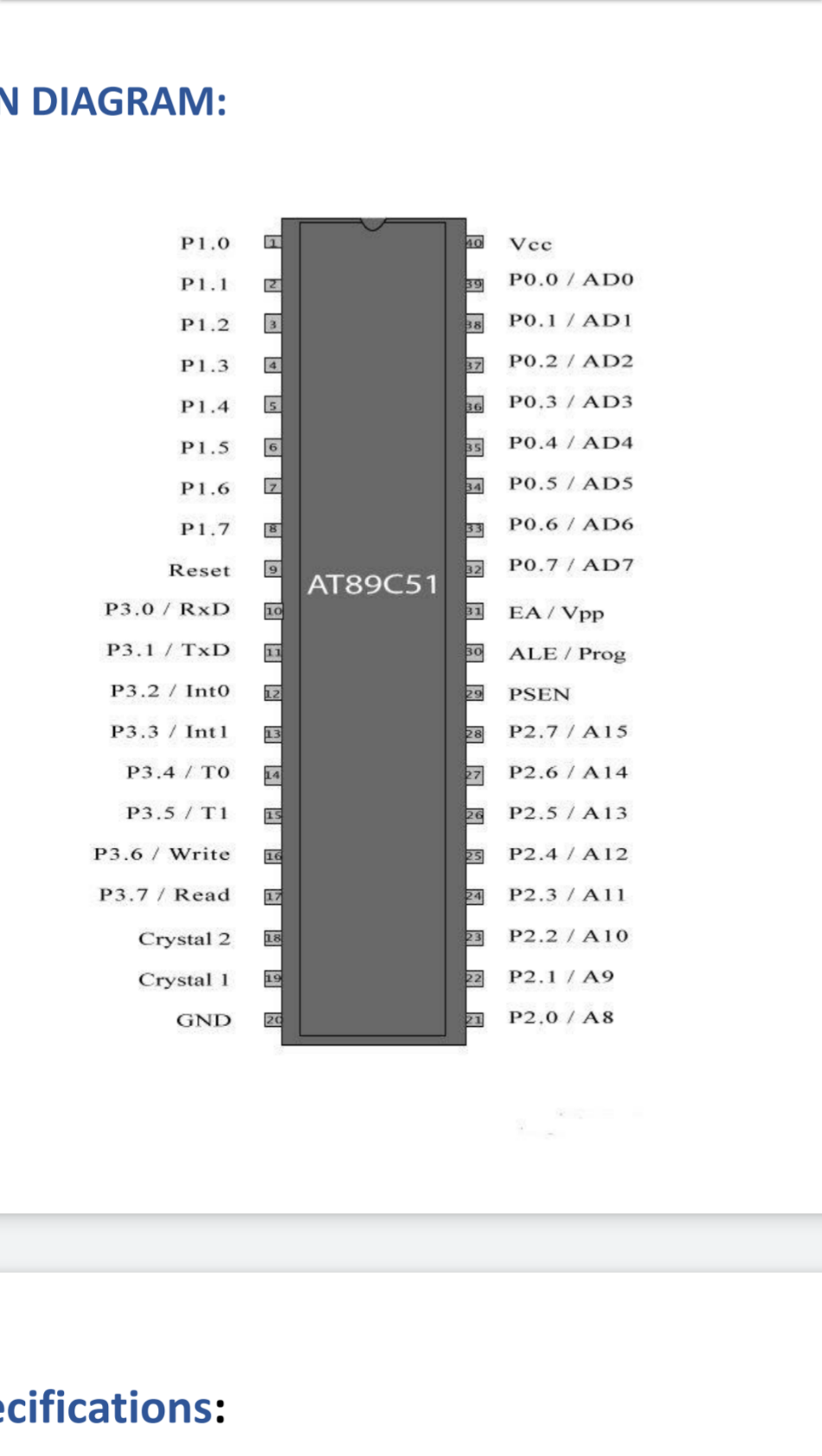
This project will extend its range till the far possible reach having a negligible delay, a setting buttons and a second display.

**2. MICROCONTROLLER AND ITS SPECS**

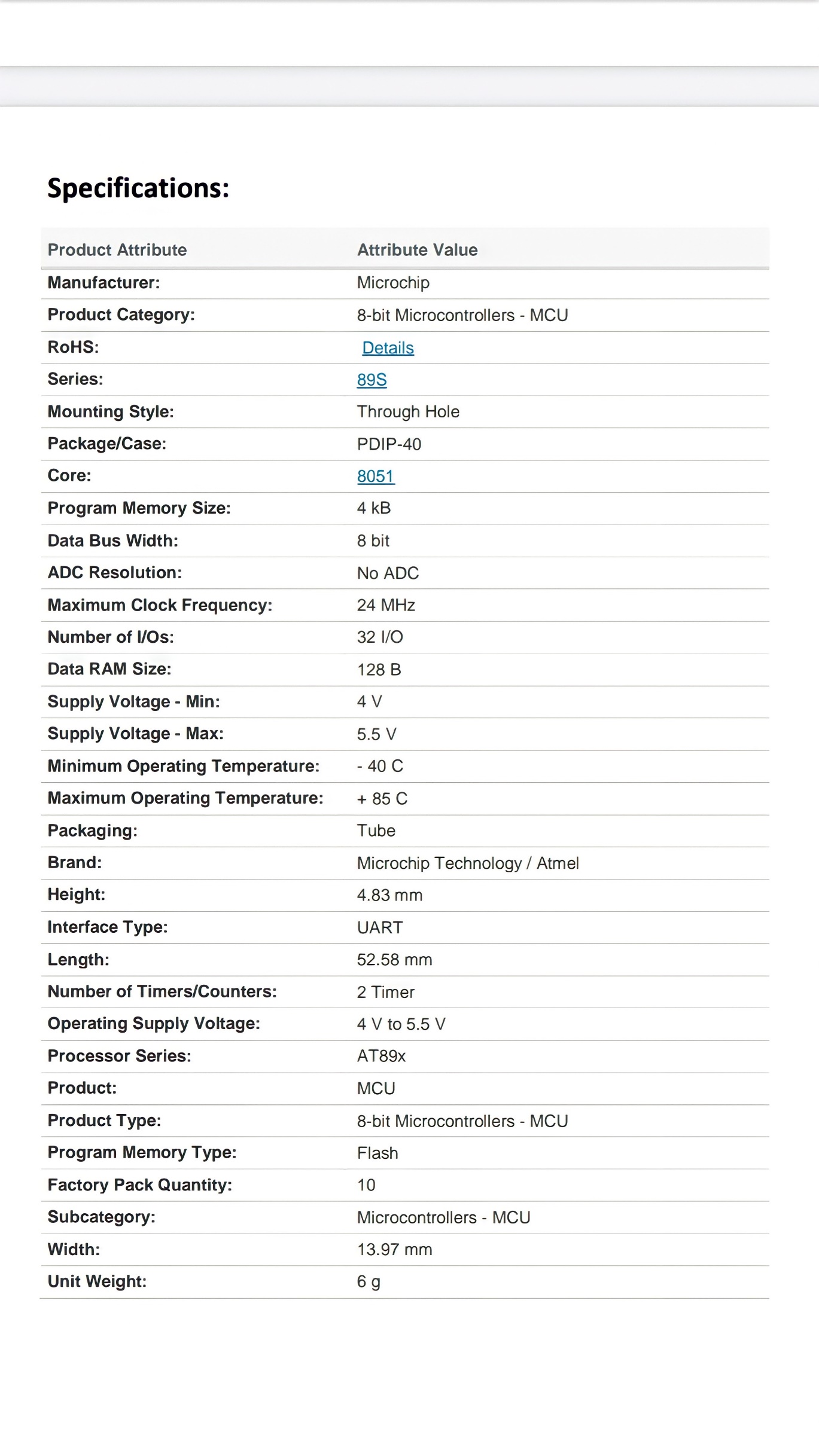
**2.a 8051 MICROCONTROLLER:**



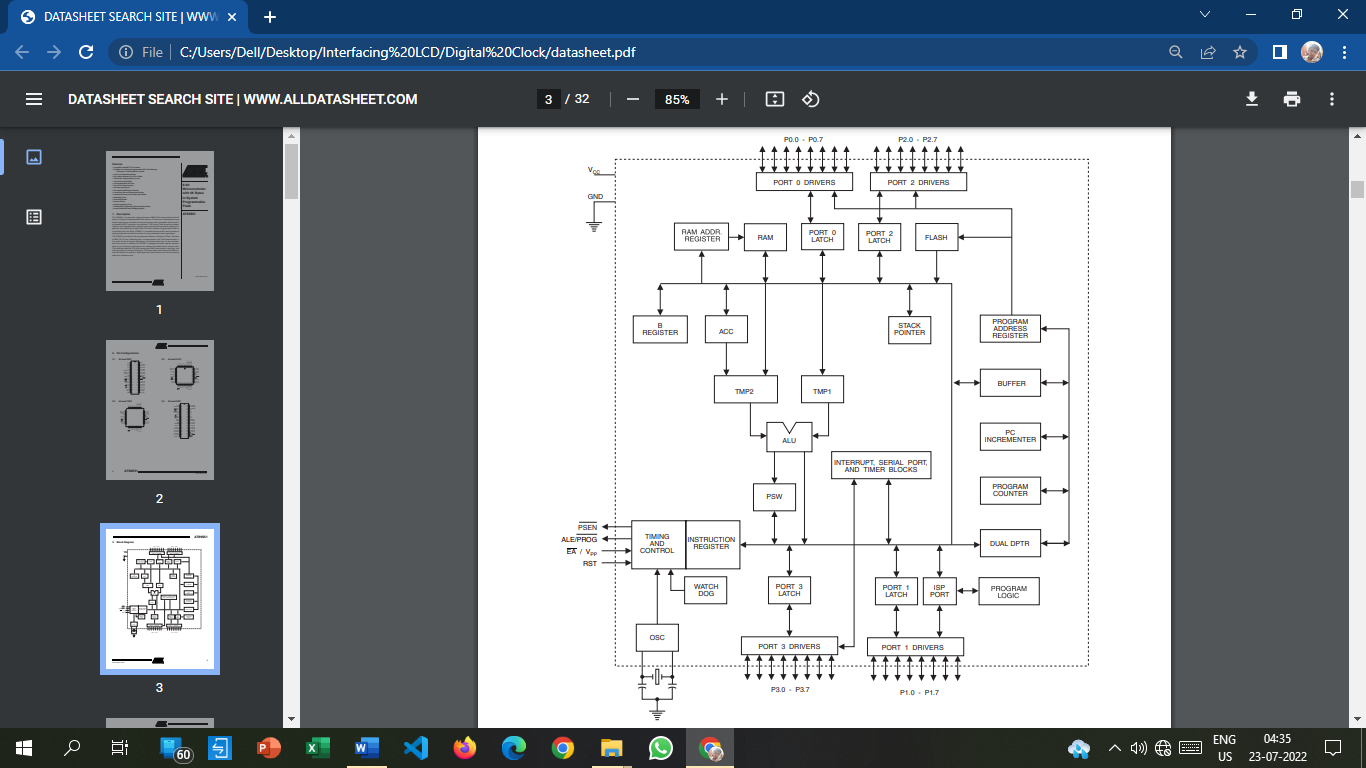
**2.b PIN DIAGRAM 8051 MICROCONTROLLER:**

****

**2.c SPECIFICATIONS:**

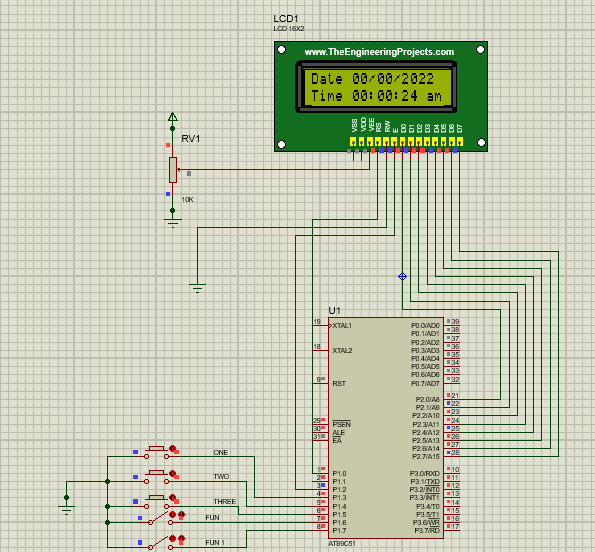
****

**2.d Block Diagram of Atmel 89s51:**

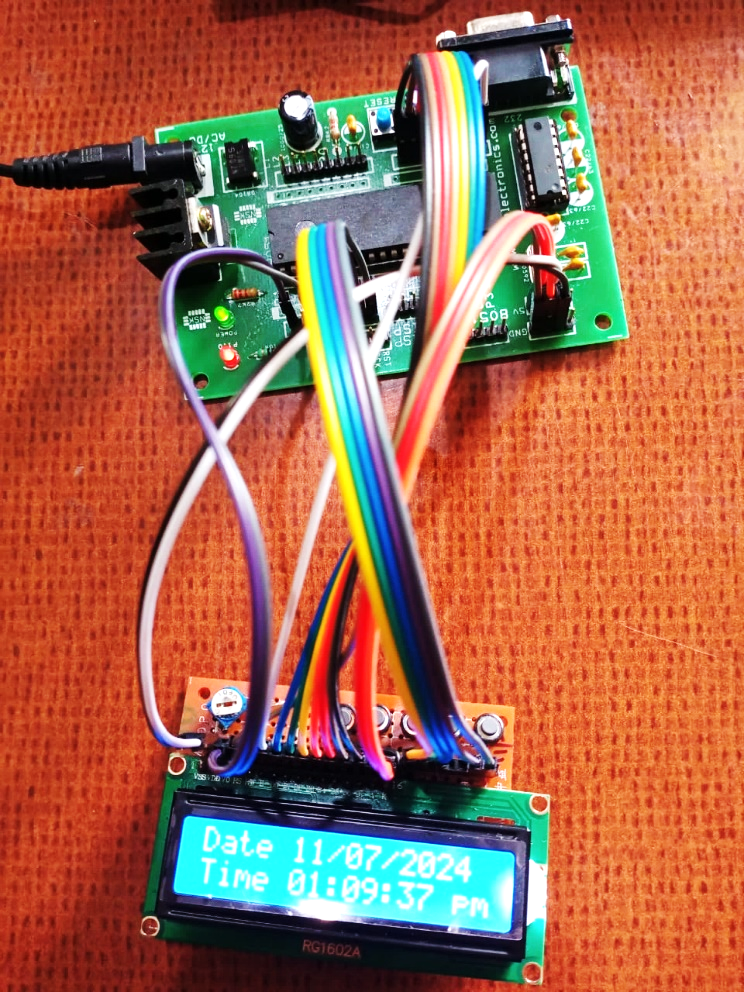
****

**CIRCUIT DIAGRAMS**

**3.a Circuit Diagram of Digital Clock Using Proteus Simulator:**

****

**3.b Circuit Diagram of Digital Clock Using Hardware components:**

****

**Components required:**

* 8051 Microcontroller
* 16x2 LCD Display
* 8051 Development Board
* Jumper Wires
* 2 Pin Push Buttons
* 12V-2A Power Adapter
* PCB Board

**4. REQUIREMENTS:**

**4.a 8051 Microcontroller [Atmel 89s51]:**

****

A **microcontroller** (**MCU** for *microcontroller unit*) is a small [computer](https://en.wikipedia.org/wiki/Computer) on a single [VLSI](https://en.wikipedia.org/wiki/VLSI) [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit) (IC) chip. A microcontroller contains one or more [CPUs](https://en.wikipedia.org/wiki/Central_processing_unit) ([processor cores](https://en.wikipedia.org/wiki/Processor_core)) along with [memory](https://en.wikipedia.org/wiki/Computer_memory) and programmable [input/output](https://en.wikipedia.org/wiki/Input/output) peripherals. Program memory in the form of [ferroelectric RAM](https://en.wikipedia.org/wiki/Ferroelectric_RAM), [NOR flash](https://en.wikipedia.org/wiki/Flash_memory#NOR_flash) or [OTP ROM](https://en.wikipedia.org/wiki/Programmable_read-only_memory) is also often included on chip, as well as a small amount of [RAM](https://en.wikipedia.org/wiki/Random-access_memory). Microcontrollers are designed for [embedded](https://en.wikipedia.org/wiki/Embedded_system) applications, in contrast to the [microprocessors](https://en.wikipedia.org/wiki/Microprocessor) used in [personal computers](https://en.wikipedia.org/wiki/Personal_computer) or other general purpose applications consisting of various discrete chips.

**4.b 8051 Development Board:**

****

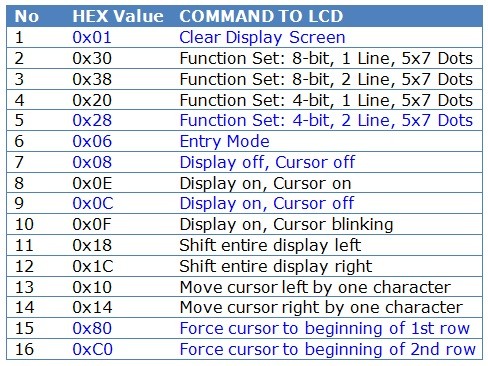
**8051 Development Board** is proposed to smooth the progress of developing and debugging of various designs encompassing Microcontrollers from Atmel, NXP and Dallas. It’s designed to facilitate (8051 DIP / PLCC package) On-board Programmer for NXP and Dallas Microcontroller through ISP on serial ports.

**4.c 16x2 LCD display:**

****

An electronic device that is used to display data and the message is known as LCD 16×2. As the name suggests, it includes 16 Columns & 2 Rows so it can display 32 characters (16×2=32) in total & every character will be made with 5×8 (40) Pixel Dots.

**Some LCD Commands**

****

|  |  |  |
| --- | --- | --- |
|  | **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. |

**4.d Jumper Wires:**

****

Jumper wires are simply **wires that have connector pins at each end**, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with [breadboards](https://www.bing.com/ck/a?!&&p=ac3880b2efb50710JmltdHM9MTY1ODUwODczMiZpZ3VpZD05MWUwMGRiMC00ZTQxLTQxMjMtYmE1MS00YTBhMmFhOGRlZmQmaW5zaWQ9NTQ2NQ&ptn=3&hsh=3&fclid=a1f0ec82-09de-11ed-a0dd-ee8db0ef302c&u=a1L3NlYXJjaD9xPUJyZWFkYm9hcmQmZmlsdGVycz1zaWQlM2FkNDgxZGMyMC01YWM4LWZhZWMtNzUzNy01NTA5ZDVjMjA0ODYmZm9ybT1FTlRMTks&ntb=1) and other prototyping tools in order to make it easy to change a circuit as needed.

**4.e 2-Pin Push Buttons:**

****

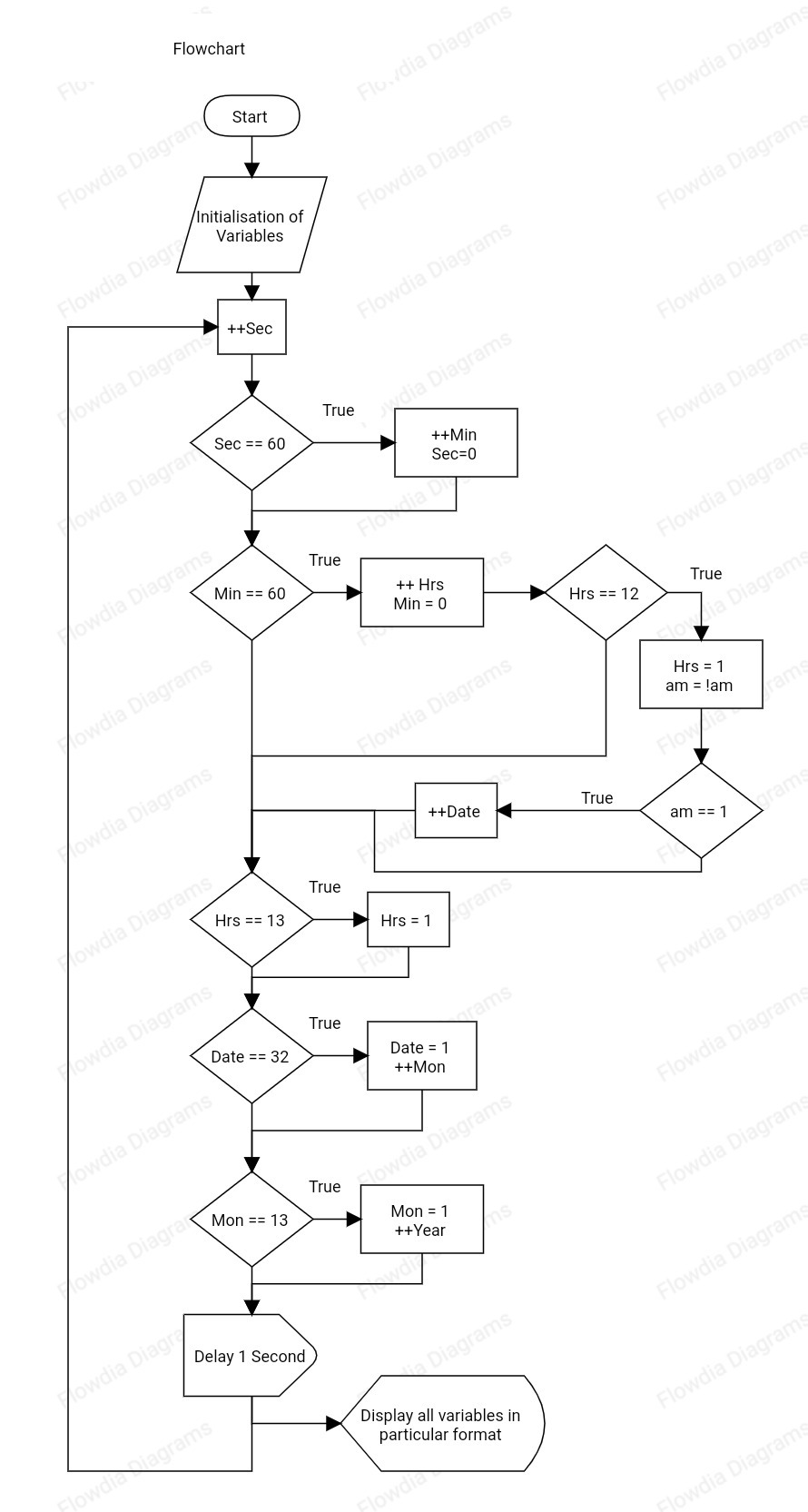
Push buttons can be explained as simple power controlling switches of a machine or appliance. These are generally metal or thermoplastic switches that are intended to grant easy access to the user.

**4.f Power Adapter:**

****

An AC power adapter is also familiar as **AC/DC adapter, AC converter, or charger.** For electronic devices, it supplies power. Such an external power supply runs on batteries. Electronic devices get this so that they don’t need to rely on other power sources. Getting plugged into a wall outlet power adapters convert AC to a single DC voltage.

**5. FLOWCHART:**

****

**6. EMBEDDED C CODE FOR Digital clock**

#include <REGX51.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){if(am==1){am=0;}

// TO CHANGE AM AND PM

else{++date;

am=1;}}}

if(hrs>12){hrs=1;}

if(date>31){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;}

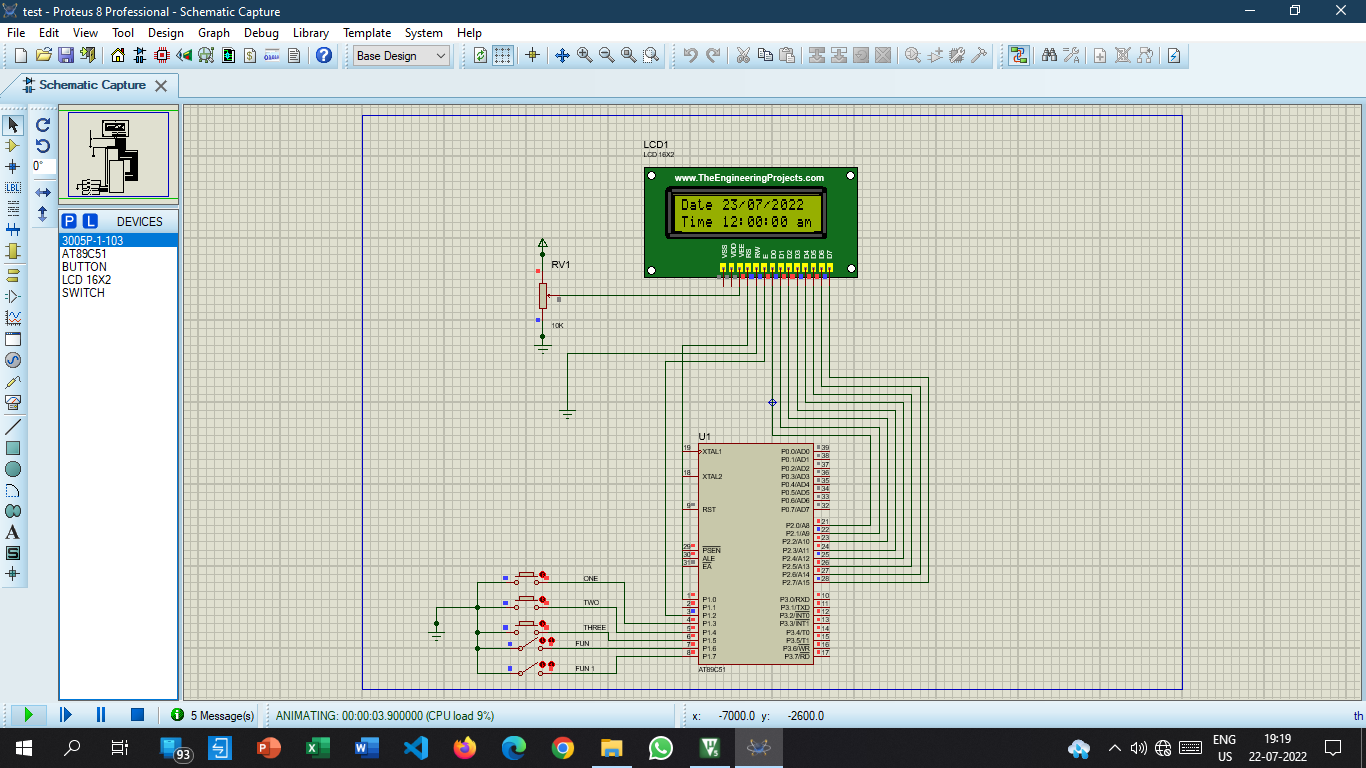
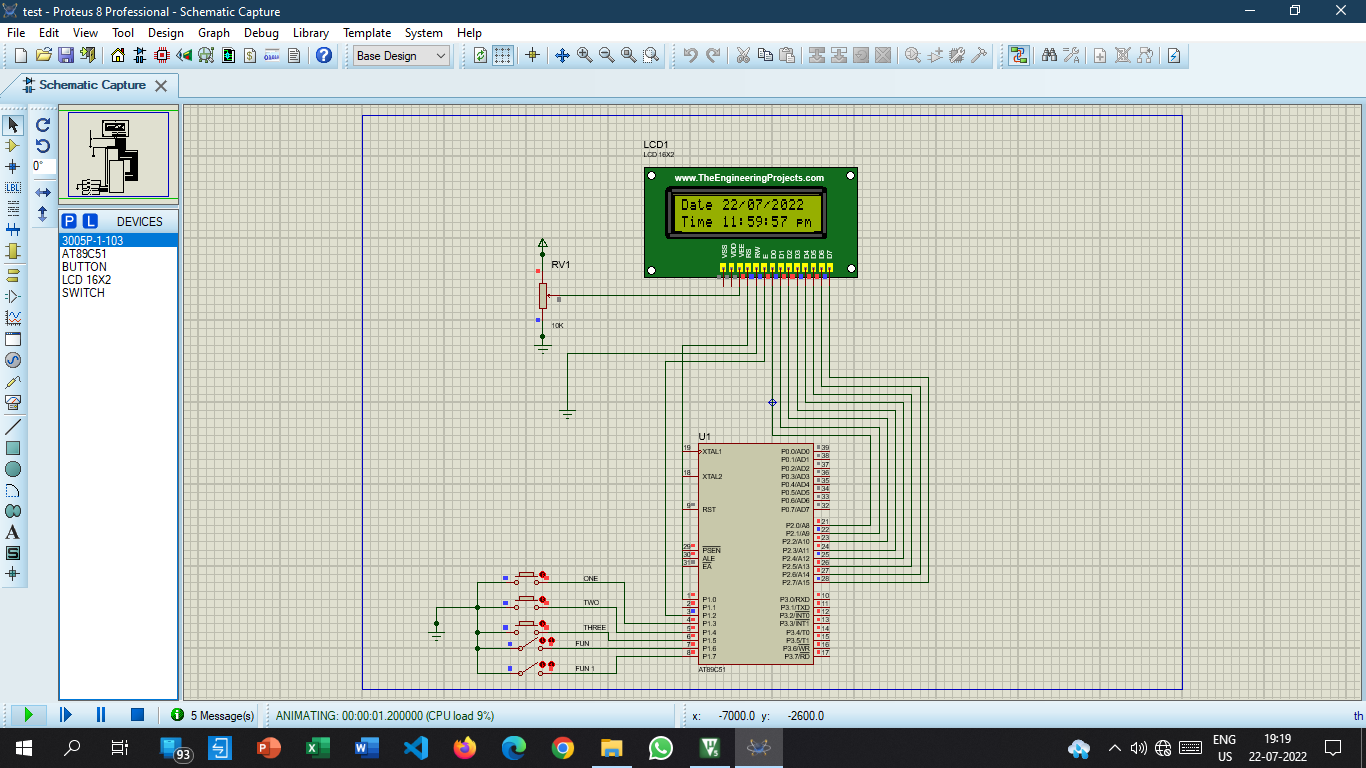
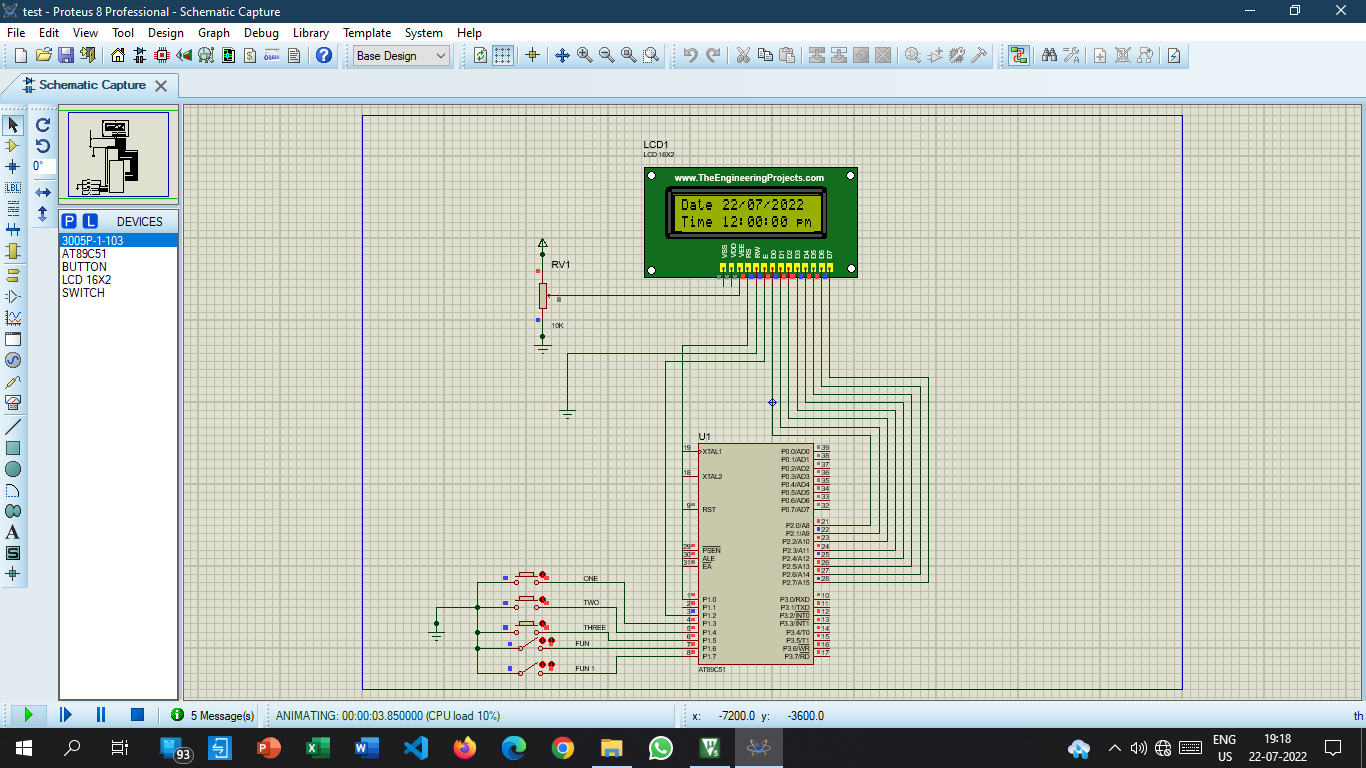
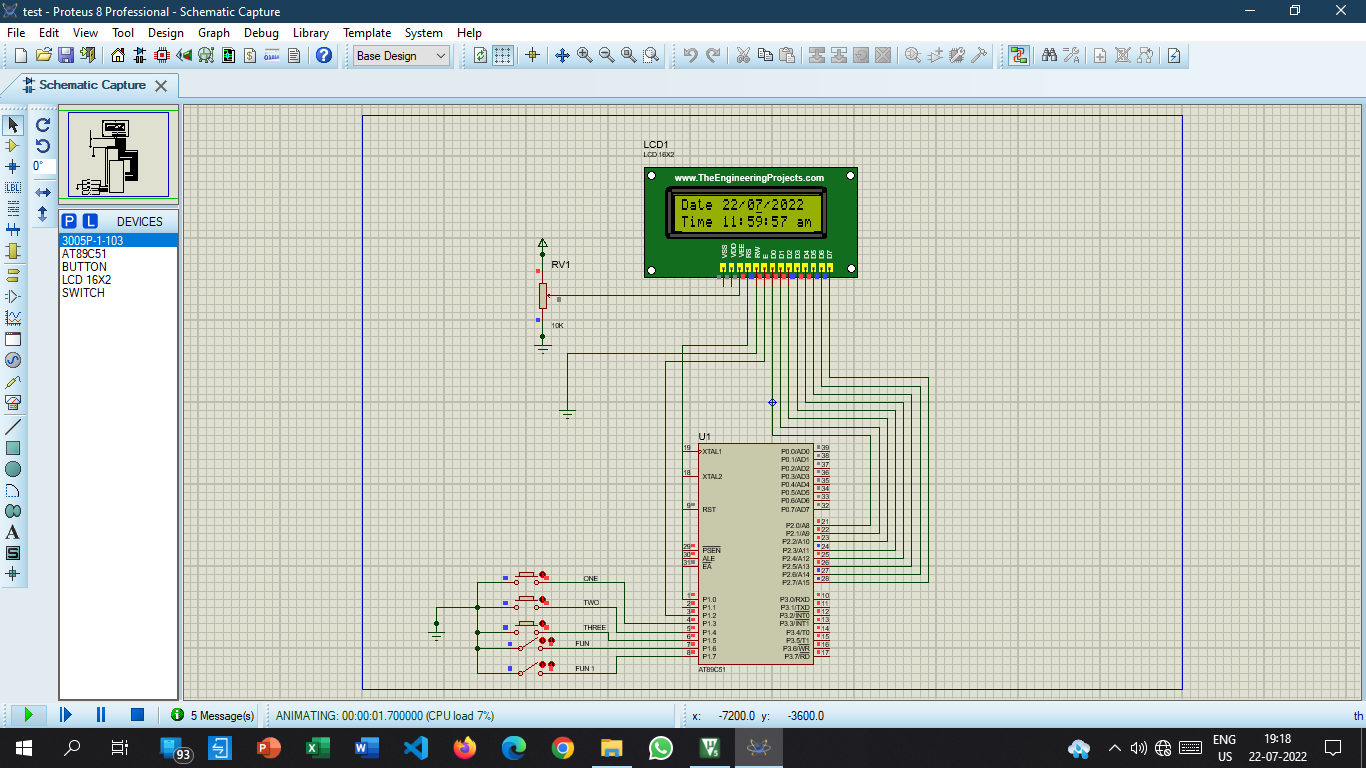
void delay(unsigned int dly){unsigned int loop = 0;

unsigned int delay\_gen = 0;

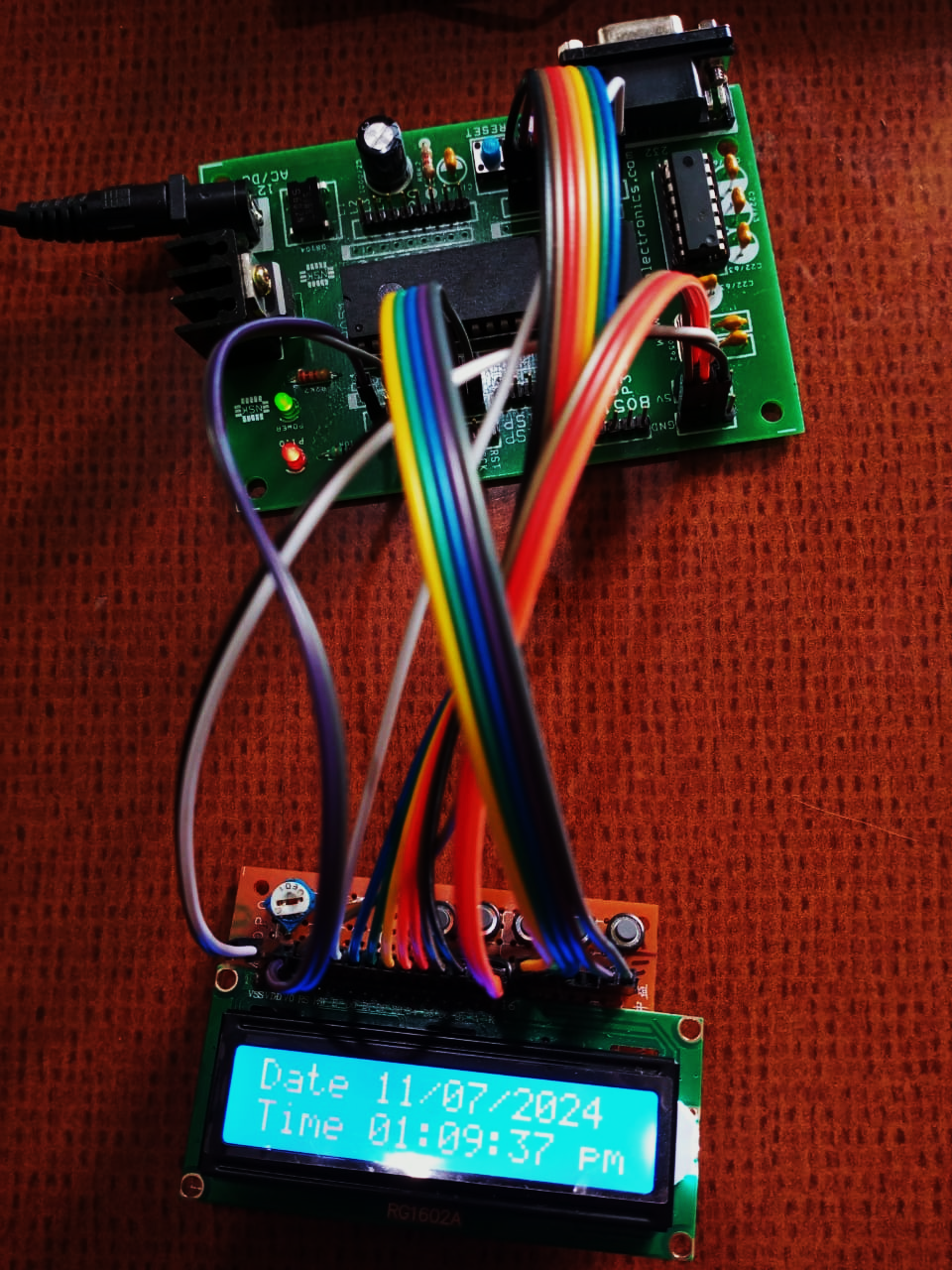
for (loop=0; loop<dly; loop++){ for (delay\_gen=0; delay\_gen<115; delay\_gen++);}}

**7.Results:**

**Results Using Proteus Simulator:**

****

**Results Using Hardware:**

****

**8. References:**

* **Some basics on C programming**
* **Internet**
* **Lcd data sheet**
* **Microcontroller datasheet**