Contents

1. **Project Title** **Page No: 1**
2. **Components** **Page No: 1**
3. **Literature Survey** **Page No: 1**
4. **Circuit Diagram** **Page No: 2**
5. **Flow Chart** **Page No: 2**
6. **Working** **Page No: 3**
7. **Advantages and Future scope** **Page No: 3**
8. **Code** **Page No: 4 - 9**
9. **References** **Page No: 10**

Project Title: Digital Clock using 8051 Microcontroller.

Components:

- **AT89C51 Microcontroller-**

1. 0-24 MHz frequency range.
2. 128x8 bit internal RAM.
3. 2x16-bit counters.

- **DS1307 I2C real-time clock chip (RTC)-**

1. AT24C32 32K I2C EEPROM memory.
2. 56 Bytes of Non-volatile memory available to user.
3. Supply voltage 4.5-5.2V.
4. Battery voltage 2.0-3.5V.
5. Two wire I2C interface.
6. 1Hz output pin.

- **LM032L**

1. 20 Character x 2 lines, built-in control LSI HD44780 type.
2. +5V single power supply.

- **Resistor-**

1. Material: Carbon Film.
2. Resistors Dimensions: 3 x 1 x 1 cm sq.
3. Weight: 5 grams.

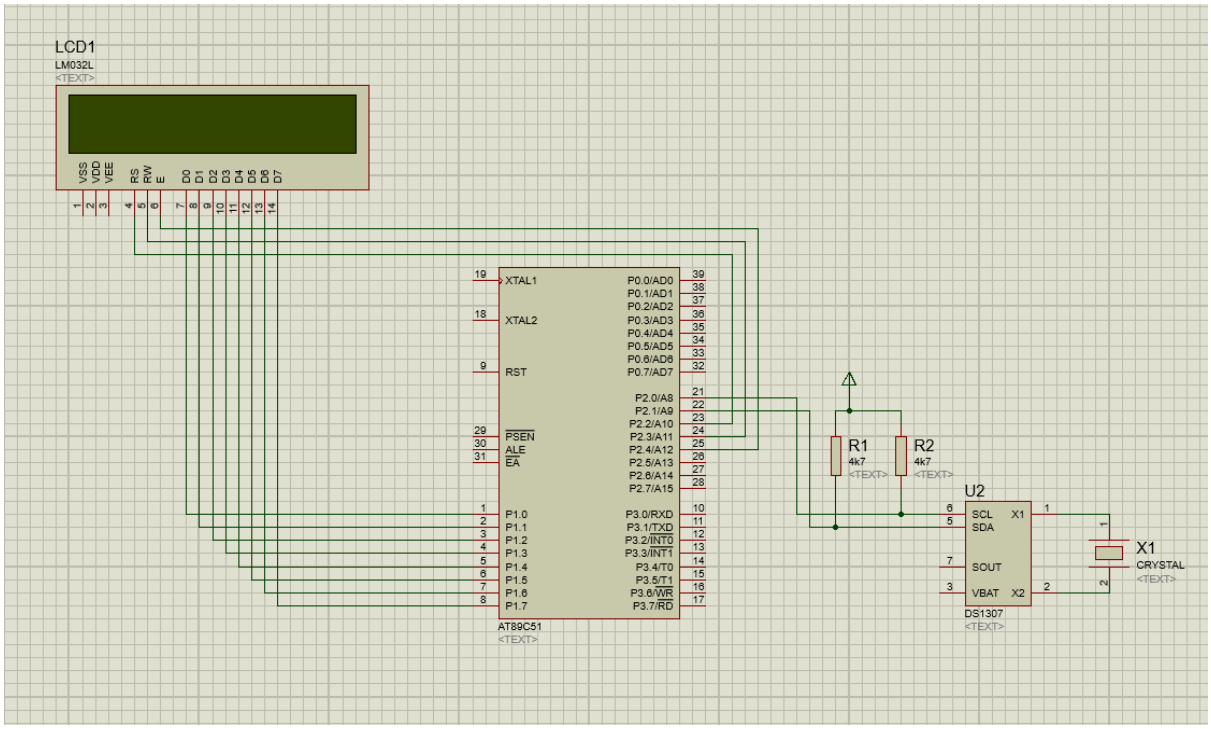
Literature Survey:

Digital clocks have evolved from early nixie tube displays to precise timekeepers with quartz oscillators. They are widely used in consumer electronics, industrial settings, and public spaces. Digital clocks are powered by microcontrollers, synchronized via radio signals or the internet, and aim to be energy-efficient.

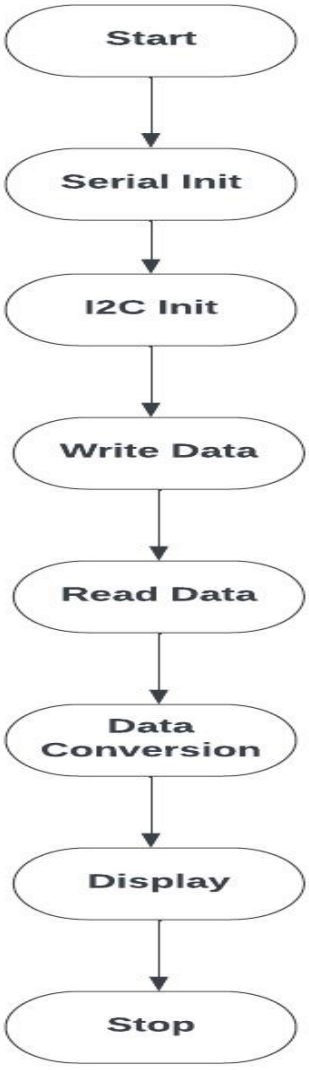
Challenges include improving energy efficiency and integrating with smart devices, while customization and aesthetics remain important design considerations. Additionally, digital clocks are crucial in scientific laboratories, data logging equipment, and various custom applications, providing accurate time references.

As technology advances, the quest for further energy efficiency and seamless integration with the Internet of Things (IoT) continues to drive innovation in the field of digital clocks.

Circuit Diagram:



Flow Chart:



Working:

This code is for interfacing a Real-Time Clock (RTC) with an 8051 micro-controller using I2C communication. It reads time and date from the RTC, displays it on an LCD, and transmits it serially. The program initializes the RTC, reads the time continuously, and displays it on an LCD. The start, stop, write, and read functions manage I2C communication. The time and date are then formatted and displayed on an LCD, and the entire process repeats. The code also includes functions for serial communication and LCD control, making it a comprehensive system for real-time clock interfacing and display.

Advantages:

- **Cost-Effective Solution**: The 8051 microcontroller is a popular and cost-effective choice for embedded systems. Its affordability makes it suitable for applications with budget constraints.
- **Ease of Programming**: The 8051 microcontroller is well-documented, and there is a wealth of resources, libraries, and development tools available for programming. This makes it easier for developers, hobbyists, and students to work with.
- **Versatility**: The 8051 microcontroller is versatile and can be used in a wide range of applications beyond digital clocks. Its flexibility makes it suitable for various projects, including embedded systems, robotics, and automation.

Future Scope:

- **Multiple Time Zones**: Incorporate the ability to display and switch between different time zones. This feature could be valuable for users in different geographical locations or for applications in international settings.
- **Alarm Functionality**: Add an alarm feature that allows users to set alarms at specific times. This could involve additional user input and interface components, such as buttons for setting the alarm time.
- **Battery Backup**: Implement a battery backup system to maintain timekeeping during power outages. This could involve adding a rechargeable battery and implementing power management functionality.

Code:

```
#include<reg51.h>
#include<intrins.h>

sbit scl=P2^0;
sbit sda=P2^1;
sbit rs=P2^2;
sbit rw=P2^3;
sbit en=P2^4;

void start();
void delay();
void check();
void lcd_display();
void lcd_init();
void lcddisp(unsigned char *);
void lcdcmd(unsigned char);
void lcddat(unsigned char);
void write(unsigned char);
void delay3();

unsigned char read();

void enter();
void ack();
void stop();
void display();
void delay2();
void serial_init();
void ser_msg(unsigned char *);
void conversion(unsigned char);

unsigned char sec, min, hr, ch;
unsigned char day, date, mon, yr;
bit c=0;

void main(){
   serial_init();
   start();
   write (0xd0);
   write (0x00);
   write (0x00);
   write (0x00);
   write (0x12);
   write (0x06);
   write (0x02);
   write (0x12);
   write (0x23);
   stop();

   while (1){
      serial_init();
      start();
```

```
write (0xd0);
write (0x00);
start();
write (0xd1);
read();
sec=ch;
sec=sec&0x7f;
stop();
start();
write (0xd0);
write (0x01);
start ();
write (0xd1);
read ();
min=ch;
min=min&0x7f;
stop();
start();
write (0xd0);
write (0x02);
start();
write (0xd1);
read ();
hr=ch;
hr=hr&0x3f;
stop();
start();
```

```
write (0xd0);
write (0x03);
start();
write (0xd1);
read();
day=ch;
day=day&0x0f;
stop();
start();
write (0xd0);
write (0x04);
start();
write (0xd1);
read();
date=ch;
date=date&0x03f;
stop();
start();
write (0xd0);
write (0x05);
start();
write (0xd1);
read();
mon=ch;
mon=mon&0x0f;
stop();
start();
```

```
        write (0xd0);
        write (0x06);
        start();
        write (0xd1);
        read();
        yr=ch;
        yr=yr&0x3f;
        stop();
        display();
        lcd_display();
        //while (1);
    }
}

void display(){
    conversion (hr);
    conversion (min);
    conversion (sec);
    enter();
    conversion (day);
    enter ();
    conversion (date);
    conversion (mon);
    conversion (yr);
    enter();
}

void conversion (unsigned char res){
    unsigned char v4, v5;

    v4=res&0x0f;
    v5=res&0xf0;
    v5=v5>>4;
    v4=v4|0x30;
    v5=v5|0x30;

    SBUF=v5; while (TI==0); TI=0;
    SBUF=v4; while (TI==0); TI=0;
    SBUF=':'; while (TI==0); TI=0;
}

void lcd_display(){
    unsigned char v9, v10;

    lcd_init();
    lcddisp("TIME:");
    lcdcmd (0xc0);
    lcddisp("DATE:");

    v9=hr&0x0f;
    v10=(hr&0xf0)>>4;

    lcdcmd (0x86);
    lcddat (v10|0x30);
```

```
lcddat (v9|0x30);
lcddat (':');

v9=min&0x0f;
v10=(min&0xf0)>>4;

lcddat (v10|0x30);
lcddat (v9|0x30);
lcddat (':');

v9=sec&0x0f;
v10=(sec&0xf0) >>4;

lcddat (v10|0x30);
lcddat (v9|0x30);
lcdcmd (0xc6);

v9=date&0x0f;
v10=(date&0xf0)>>4;

lcddat (v10|0x30);
lcddat (v9|0x30);
lcddat (':');

v9=mon&0x0f;
v10=(mon&0xf0)>>4;

   lcddat (v10|0x30);
   lcddat (v9|0x30);
   lcddat (':');

   v9=yr&0x0f;
   v10=(yr&0xf0)>>4;

   lcddat (v10|0x30);
   lcddat (v9|0x30);
   delay3();
}

void delay3(){
   unsigned int v11;
   for (v11=0; v11<32000; v11++);
}

void lcd_init(){
   lcdcmd (0x38);
   lcdcmd (0x01);
   lcdcmd (0x10);
   lcdcmd (0x0c);
   lcdcmd (0x80);
}

void enter (){
   SBUF=0X0D;
```

```
    while (TI==0);
    TI=0;
}

void write (unsigned char val){
    unsigned char v2=0x80, v3;
    unsigned char v5=val;

    for (v3=0; v3<8; v3++){
        sda=v5&v2;
        scl=1;
        delay();
        scl=0;
        v5=v5<<1;
    }

    c=sda;
    scl=1;
    delay();
    scl=0;

    if (c==1){
        stop();
        while(1);
    }
}

void stop (){
    scl=1;
    sda=0;
    delay();

    sda=1;
    scl=0;
    delay();
}

void delay(){
    nop();
    nop();
}

unsigned char read(){
    unsigned char v16;
    bit m=0;
    ch = 0;

    for(v16=0; v16<=7; v16++){
        scl=1;
        delay();
        m=sda;
        scl=0;

        if(m==1){
```

```
            ch|=0x01;
        }

        if(v16<=6)
        ch=ch<<1;
        delay();
    }
    return ch;
}

void start(){
    scl=1;
    sda=1;
    delay();
    sda=0;
    delay();
    scl=0;
}

void serial_init(){
    SCON=0x50;
    TMOD=0x20;
    TH1=-3;
    TR1=1;
}

void lcddisp(unsigned char *m){
    unsigned char p;
    for(p=0;p[m]!='\0';p++)
    lcddat(m[p]);
}

void lcddat(unsigned char val){
    P1 = val;
    rs = 1;
    rw = 0;
    en = 1;
    delay2();
    en=0;
}

void lcdcmd(unsigned char val){
    P1=val;
    rs=0;
    rw=0;
    en=1;
    delay2();
    en=0;
}

void delay2(){
    unsigned int v1;
    for(v1=0; v1<2000; v1++);
}
```

References:

https://www.youtube.com/watch?v=wC1uN71kDkE

https://gmostofabd.github.io/8051-RTC/