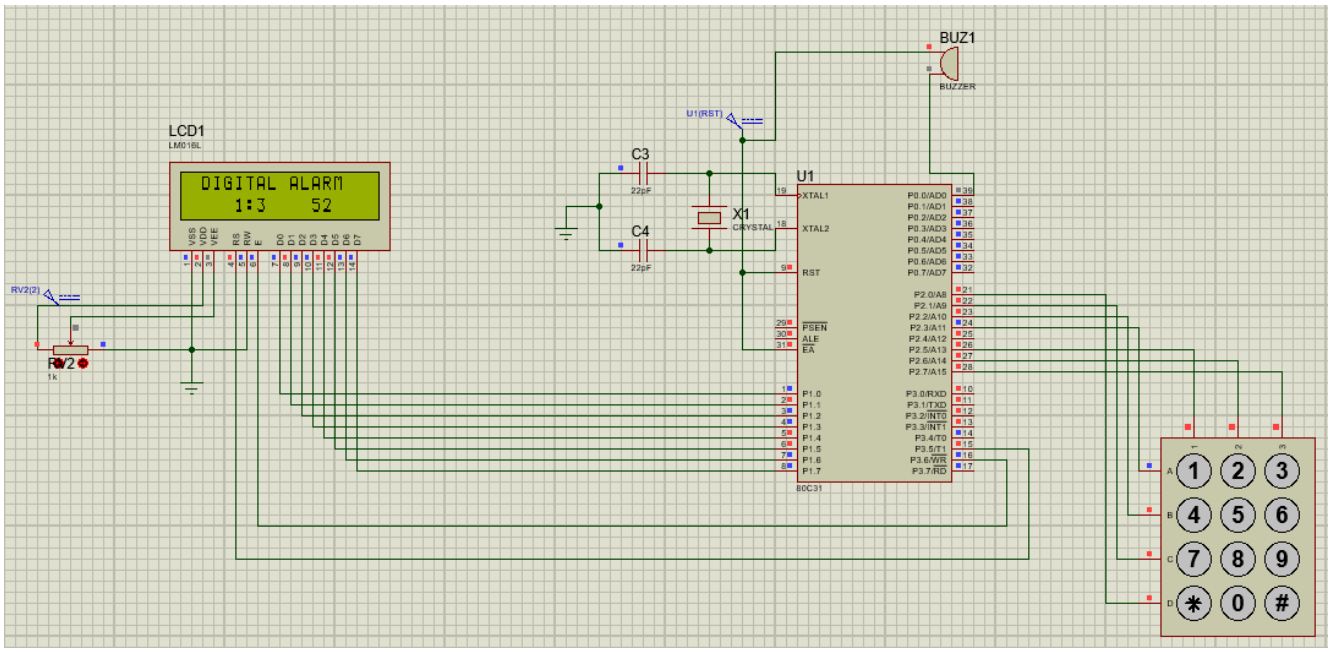
Digital Clock using 8051 Microcontroller

Circuit Diagram:



Code Overview:

1. Hardware Connections:
   - Uses the AT89C52 (8051) microcontroller.
   - P0, P1, P2, P3 ports control the LCD, keypad, and buzzer.
   - Keypad scanning is performed using row-column logic on port P2.
   - LCD control pins (rs, rw, en) are assigned to P3.
   - Buzzer (P0^0) activates when the alarm time matches.
2. Main Functionalities:
   - lcdcmd() and lcddata() → Send commands and data to the LCD.
   - keypad() and scankey() → Detect user input from a 4x3 keypad.
   - settime() → User inputs the current time using the keypad.
   - checkconditions() → Compares real-time clock with the alarm time.
   - clockdelay() → Uses Timer 0 to create 1-second delays.
   - start() → Displays the real-time clock on the LCD.
   - tochar() → Converts numerical values to characters for LCD display.
   - main() → Initializes hardware, sets the time, and runs the clock loop.

Source Code:

```
#include<reg52.h>
int keypad();
void tochar(unsigned int value);
sbit alm=P3^0;
sbit buzzer=P0^0;
sbit r0=P2^0;
sbit r1=P2^1;
sbit r2=P2^2;
sbit r3=P2^3;
sbit c0=P2^5;
sbit c1=P2^6;
sbit c2=P2^7;
sbit en=P3^6;
sbit rs=P3^5;
sbit rw=P3^7;
char hour[]="HOURS-> 01 - 12",mint[]="MINTS->
01 - 60";
char alarm[]=" DIGITAL ALARM";
int
count=0,H,M,S=1,hour1[1],mint1[1],H1,M1,hour2[
1],mint2[1];


void delay(unsigned int no)
{
unsigned int i,j;
for(j=0;j<=no;j++)
for(i=0;i<=2;i++);
}
void lcdcmd(unsigned int  command){
P1=command;
rw=0;
rs=0;
en=0;
delay(300);
en=1;
delay(300);
en=0;
}
void lcddata(char data1)
{
P1=data1;
rw=0;
rs=1;
en=0;
delay(300);
en=1;
delay(300);
en=0;
}
 int scankey(){
 r0=1;r1=1;r2=1;r3=0;
if(c0==0){return 2;}
 }
void checkconditions(){
  unsigned int i;
  i=scankey();
while(i==2)
        {
                count=0;
lcdcmd(0x01);
lcdcmd(0x80);
        while(hour[count]!='\0')
                {

        lcddata(hour[count]);count++;
                }
```

```
count=0;
lcdcmd(0xC8);
        while(count!=2)
                {
        hour2[count]=keypad();count++;
                }
H1=(hour2[0]*10)+hour2[1];
count=0;
lcdcmd(0x01);
lcdcmd(0x80);
        while(mint[count]!='\0')
                {
        lcddata(mint[count]);count++;
                }
count=0;
lcdcmd(0xC8);
        while(count!=2)
                {
        mint2[count]=keypad();count++;
                }
M1=(mint2[0]*10)+mint2[1];
count=0;
delay(1000);
lcdcmd(0x01);
lcdcmd(0x80);
count=0;
        while(alarm[count]!='\0')
                {
        lcddata(alarm[count]);count++;
                }
count=0;
lcdcmd(0xC4);tochar(H);lcddata(':');tochar(M);lcdcmd(0xCB);tochar(S);
i=1;
                }
if(S>=60)
{
M++;
}
if(M>=60)
{
H++;
}
if((H==13) && (M>=60) && (S>=60))
{
S=1;
H=1;
M=1;
lcdcmd(0x01);
count=0;
        while(alarm[count]!='\0')
                {
        lcddata(alarm[count]);count++;
                }
count=0;
lcdcmd(0xC4);tochar(H);lcddata('
');lcddata(':');tochar(M);lcddata('
');lcdcmd(0xCB);tochar(S);lcddata(' ');
}

if(S>=60)
{
S=1;
```

```
lcdcmd(0xC4);tochar(H);lcddata(':');tochar(M);lcdcmd(0xCB);tochar(S);lcddata(' ');
}
if(M>=60)
{
M=1;
lcdcmd(0xC4);tochar(H);lcddata(':');tochar(M);lcddata(' ');lcdcmd(0xCB);tochar(S);lcddata(' ');
}
}
void clockdelay()
{
int count=0,sec;
for(sec=0;sec<60;sec++){
count=0;
while(count!=495)
{
TMOD=0x01;  //16-bit timer0 selected
TH0=0xF8;   // Loading high byte in TH
TL0=0xCC;   // Loaded low byte in TL
TR0=1;      // Running the timer
while(!TF0);   //Checking the timer flag register if it is not equal to 1
TR0 = 0;      // If TF0=1 stop the timer
TF0 = 0;      // Clear the Timer Flag bit for next calculation
count++;
}
S++;lcdcmd(0xCB);tochar(S);
checkconditions();
if(H==H1)
{
if(M==M1)
buzzer=0;
}
}
}
void lcdint()
{
lcdcmd(0x30);  delay(3000);  lcdcmd(0x30);
        delay(3000);  lcdcmd(0x30);
        delay(3000);
lcdcmd(0x30);  delay(3000);  lcdcmd(0x30);
        delay(3000);  lcdcmd(0x38);
        delay(3000);
lcdcmd(0x01);  delay(3000);  lcdcmd(0x0C);
        delay(3000);  lcdcmd(0x80);
        delay(3000);
}
int keypad()
{
char c='a';
while(c!='s'){
r0=0;r1=1;r2=1;r3=1;
if(c0==0){lcddata('1');delay(20000);c='s';return 1;}
 if(c1==0){lcddata('2');delay(20000);c='s';return 2;}
 if(c2==0){lcddata('3');delay(20000);c='s';return 3;}
r0=1;r1=0;r2=1;r3=1;
if(c0==0){lcddata('4');delay(20000);c='s';return 4;}
 if(c1==0){lcddata('5');delay(20000);c='s';return 5;}
 if(c2==0){lcddata('6');delay(20000);c='s';return 6;}
r0=1;r1=1;r2=0;r3=1;
if(c0==0){lcddata('7');delay(20000);c='s';return 7;}
 if(c1==0){lcddata('8');delay(20000);c='s';return 8;}
 if(c2==0){lcddata('9');delay(20000);c='s';return 9;}
r0=1;r1=1;r2=1;r3=0;
//if(c0==0){lcddata('*');P0=0xF0;delay(20000);c='s';
}
 if(c1==0){lcddata('0');delay(20000);c='s';return 0;}
```

```
//
if(c2==0){lcddata('#');P0=0xF0;delay(20000);c='s';}
  c='a';
}
}
void settime()      //Setting time
{
lcdcmd(0x01);
lcdcmd(0x80);
            while(hour[count]!='\0')
                        {

            lcddata(hour[count]);count++;
                        }
count=0;
lcdcmd(0xC8);
            while(count!=2)
                        {

            hour1[count]=keypad();count++;
                        }
H=(hour1[0]*10)+hour1[1];
count=0;
lcdcmd(0x01);
lcdcmd(0x80);
            while(mint[count]!='\0')
                        {

            lcddata(mint[count]);count++;
                        }
count=0;
lcdcmd(0xC8);
            while(count!=2)
                        {
            mint1[count]=keypad();count++;
                        }
M=(mint1[0]*10)+mint1[1];
count=0;
delay(1000);
lcdcmd(0x01);
lcdcmd(0x80);
}
void start(){
count=0;
lcdcmd(0x01);
lcdcmd(0x80);
            while(alarm[count]!='\0')
                        {

            lcddata(alarm[count]);count++;
                        }
count=0;
lcdcmd(0xC4);
tochar(H);
lcddata(':');
tochar(M);
lcdcmd(0xCB);
tochar(S);
}
void tochar(unsigned int value)
{
            char tendigit,unitdigit;
            tendigit=value/10;
if((tendigit+48)>='1')
            lcddata(tendigit+48);
            unitdigit=value%10;
```

```
        lcddata(unitdigit+48);
}
void main()
{
unsigned int i=0;
P0=0x00;
P1=0x00;
P2=0xF0;
P3=0x0F;
lcdint();
buzzer=1;
settime();
 start();
while(1)
        {
                clockdelay();
                buzzer=1;
        }
}
```

Conclusion:

This project implements a digital alarm clock using the 8051 microcontrollers with an LCD display, keypad input, and a buzzer for alarm notifications. The system allows users to set the current time and configure an alarm, which is triggered when the real-time clock matches the alarm time.
The code utilizes 8051 timers (Timer0) to generate a one-second delay, ensuring accurate timekeeping. It also employs keypad scanning logic for user input and LCD interfacing functions to display time and alarm settings.
However, there are some limitations, such as incomplete keypad scanning, potential timer inaccuracies due to polling-based delay, and minor logical errors in time resetting and alarm activation. These can be improved using interrupt-driven timers, enhanced debouncing for keypad input, and optimized alarm logic for precise operation.
Overall, this program provides a functional base for a digital alarm clock, demonstrating essential embedded system concepts, including GPIO interfacing, LCD communication, timer usage, and keypad handling. Further refinements can enhance its accuracy, reliability, and usability.