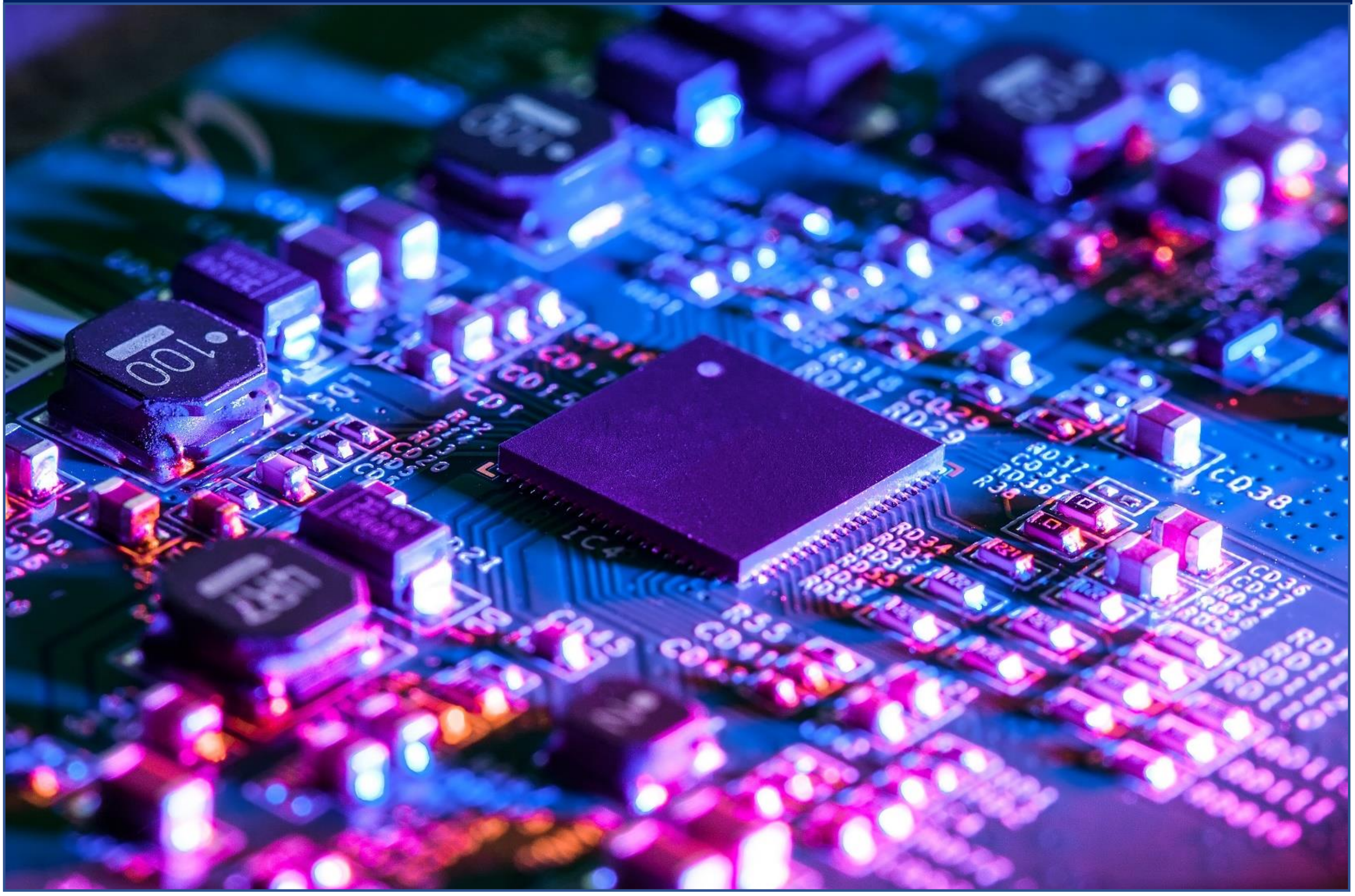
Real-Time Alarm Clock and Calendar based on Arduino UNO



Project Report on Real-Time Alarm Clock and Calendar based on Arduino UNO

Course Title: Operating Systems Design

Course Code: CSE323

Section: 06

Semester: Fall 2022

Submitted to

Dr. Saeed Mahmud Ullah

Professor

Department of Electrical and Electronic Engineering

University of Dhaka

Submitted by

Sl. No	Name	ID
1	Rania Noor	2011492642
2	Zarin Tasnim Pushpita	2011280642
3	Mahirul Alam Chowdhury	2013445042
4	Md. Tanvir Ann Noor meem	2021516642
5	Iftekher Mahbub Rafi	2021463642

Date of Submission: 17/12/2022

Table of Contents

OBJECTIVES 3

COMPONENTS 3

SOFTWARE 3

WORKING DIAGRAM 4

CODE 6

DISCUSSION 26

REFERENCE 27

OBJECTIVES

- Understand how hardware connects to the software through an Operating System
- Learn the use of RTC modules for timekeeping in micro-controllers
- Learn about the various types of micro-controllers i.e., Arduino UNO, Arduino Nano etc.
- Learn the workings of various type of hardware components
- Understand, implement a prototype clock using Arduino UNO

COMPONENTS

- Arduino UNO
- Real Time Clock (RTC) DS1307
- Display LCD 20×04 with I2C module
- Pushbutton Switch
- LED
- Resistor 220 ohm
- Buzzer
- Jumper wires

SOFTWARE

- Arduino IDE

WORKING DIAGRAM

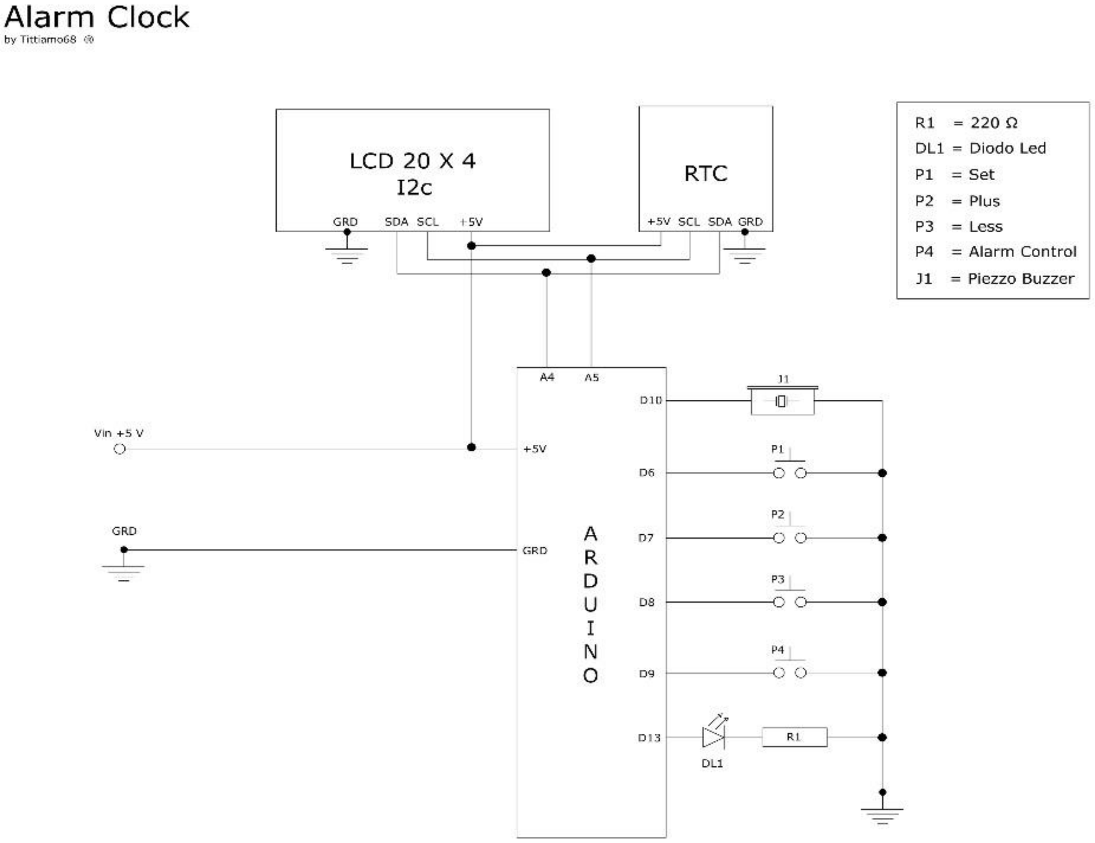


Figure 1 Circuit Diagram

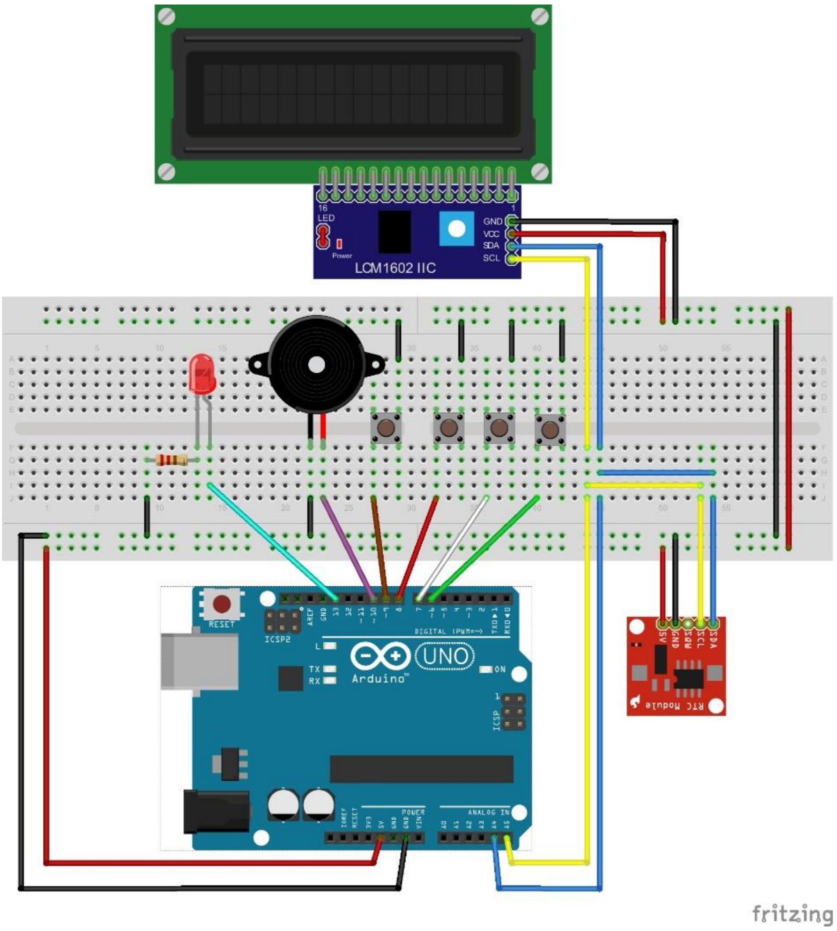


Figure 2 Circuit Diagram | Fritz Diagram

CODE

```
/*

:Project:Alarm_Clock

*/


//************libraries**************//

#include <Wire.h>

#include <RTClib.h>

#include <LiquidCrystal_I2C.h>


//************************************//

LiquidCrystal_I2C lcd(0x27,20,4); // Display I2C 20 x 4

RTC_DS1307 RTC.


//************Button*****************//

int P1=6; // Button SET MENU'

int P2=7; // Button +

int P3=8; // Button -

int P4=9; // SWITCH Alarm


//**************Alarm***************//

#define LED 13

#define buzzer 10
```

```
//************Variables**************//

int hourupg;

int minupg;

int yearupg;

int monthupg;

int dayupg;

int menu =0;

int setAll =0;


uint8_t alarmHours = 0, alarmMinutes = 0;  // Holds the current alarm time




void setup()

{


  lcd.begin();

  lcd.backlight();

  lcd.clear();


  pinMode(P1,INPUT_PULLUP); // https://www.arduino.cc/en/Tutorial/InputPullupSerial

  pinMode(P2,INPUT_PULLUP);

  pinMode(P3,INPUT_PULLUP);

  pinMode(P4,INPUT_PULLUP);

  pinMode(LED,OUTPUT);
  pinMode(buzzer, OUTPUT); // Set buzzer as an output
  printAllOff();
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // Set the date and time at compile time
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  // RTC.adjust(DateTime(__DATE__, __TIME__)); //removing "//" to adjust the time
    // The default display shows the date and time
  int menu=0;
}

void loop()
{

// check if you press the SET button and increase the menu index
  if(digitalRead(P1)== LOW)
  {
   menu=menu+1;
  }
  if((digitalRead(P2)== LOW)&&(digitalRead(P3)== LOW))
  {

    DisplaySetHourAll();
    DisplaySetMinuteAll();
    lcd.clear();
    lcd.setCursor(5,0);
    lcd.print("ALARM");
    lcd.setCursor(5,1);
    lcd.print(alarmHours, DEC);
    lcd.print(":");
    lcd.print(alarmMinutes, DEC);
    delay(1000);
    lcd.clear();
  }
// in which subroutine should we go?
  if (menu==0)
    {
     DisplayDateTime(); // void DisplayDateTime
     Alarm(); // Alarm control
          }
  if (menu==1)
    {
    DisplaySetHour();
    }
  if (menu==2)
    {
    DisplaySetMinute();
    }
  if (menu==3)
    {
    DisplaySetYear();
    }
  if (menu==4)
    {
    DisplaySetMonth();
    }
  if (menu==5)
    {
    DisplaySetDay();
    }
  if (menu==6)
    {
    StoreAgg();
    delay(500);
    menu=0;
    }
    delay(100);
```

```
}


void DisplayDateTime ()
{
// We show the current date and time
  DateTime now = RTC.now();


  lcd.setCursor(0, 2);
  lcd.print("Hour : ");


  if (now.hour()<=9)
  {
    lcd.print("0");
  }
  lcd.print(now.hour(), DEC);
  hourupg=now.hour();
  lcd.print(":");
  if (now.minute()<=9)
  {
    lcd.print("0");
  }
  lcd.print(now.minute(), DEC);
  minupg=now.minute();
  lcd.print(":");
```

```
if (now.second()<=9)

{

  lcd.print("0");

}

lcd.print(now.second(), DEC);



lcd.setCursor(0, 1);

lcd.print("Date : ");

if (now.day()<=9)

{

  lcd.print("0");

}

lcd.print(now.day(), DEC);

dayupg=now.day();

lcd.print("/");

if (now.month()<=9)

{

  lcd.print("0");

}

lcd.print(now.month(), DEC);

monthupg=now.month();

lcd.print("/");

lcd.print(now.year(), DEC);
```

```
  yearupg=now.year();


  char DOW[][10]={"Sunday   ","Monday   ","Tuesday  ","Wednesday","Thursday ","Friday   ","Saturday "};

  lcd.setCursor(0, 0);

  lcd.print("Day  : ");

  lcd.print(DOW[now.dayOfTheWeek()]); // if it appears error in the code, enter the code given below

  //lcd.print(DOW[now.dayOfWeek()]);

}


void DisplaySetHour()

{

// time setting

  lcd.clear();

  DateTime now = RTC.now();

  if(digitalRead(P2)==LOW)

  {

    if(hourupg==23)

    {

      hourupg=0;

    }

    else

    {

      hourupg=hourupg+1;
```

```
     }
   }
    if(digitalRead(P3)==LOW)
   {
     if(hourupg==0)
     {
       hourupg=23;
     }
     else
     {
       hourupg=hourupg-1;
     }
   }
   lcd.setCursor(0,0);
   lcd.print("Set time:");
   lcd.setCursor(0,1);
   lcd.print(hourupg,DEC);
   delay(200);
 }

void DisplaySetMinute()
{
// Setting the minutes
  lcd.clear();
```

```
if(digitalRead(P2)==LOW)
{
  if (minupg==59)
  {
    minupg=0;
  }
  else
  {
    minupg=minupg+1;
  }
}
 if(digitalRead(P3)==LOW)
{
  if (minupg==0)
  {
    minupg=59;
  }
  else
  {
    minupg=minupg-1;
  }
}
lcd.setCursor(0,0);
lcd.print("Set Minutes:");
```

```
  lcd.setCursor(0,1);
  lcd.print(minupg,DEC);
  delay(200);
}

void DisplaySetYear()
{
// setting the year
  lcd.clear();
  if(digitalRead(P2)==LOW)
  {
    yearupg=yearupg+1;
  }
   if(digitalRead(P3)==LOW)
  {
    yearupg=yearupg-1;
  }
  lcd.setCursor(0,0);
  lcd.print("Set Year:");
  lcd.setCursor(0,1);
  lcd.print(yearupg,DEC);
  delay(200);
}
```

```
void DisplaySetMonth()
{
// Setting the month
  lcd.clear();
  if(digitalRead(P2)==LOW)
  {
    if (monthupg==12)
    {
      monthupg=1;
    }
    else
    {
      monthupg=monthupg+1;
    }
  }
   if(digitalRead(P3)==LOW)
  {
    if (monthupg==1)
    {
      monthupg=12;
    }
    else
    {
      monthupg=monthupg-1;
    }
  }
  lcd.setCursor(0,0);
  lcd.print("Set Month:");
  lcd.setCursor(0,1);
  lcd.print(monthupg,DEC);
  delay(200);
}

void DisplaySetDay()
{
// Setting the day
  lcd.clear();
  if(digitalRead(P2)==LOW)
  {
    if (dayupg==31)
    {
      dayupg=1;
    }
    else
    {
      dayupg=dayupg+1;
    }
  }
```

```
   if(digitalRead(P3)==LOW)
  {
    if (dayupg==1)
    {
      dayupg=31;
    }
    else
    {
      dayupg=dayupg-1;
    }
  }
  lcd.setCursor(0,0);
  lcd.print("Set Day:");
  lcd.setCursor(0,1);
  lcd.print(dayupg,DEC);
  delay(200);
}

void StoreAgg()
{
// Variable saving
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SAVING IN");
  lcd.setCursor(0,1);
  lcd.print("PROGRESS");
  RTC.adjust(DateTime(yearupg,monthupg,dayupg,hourupg,minupg,0));
  delay(200);
}
void DisplaySetHourAll()// Setting the alarm minutes
{
  while(digitalRead(P1)==HIGH){

  lcd.clear();

  if(digitalRead(P2)==LOW)
  {
    if(alarmHours==23)
    {
      alarmHours=0;
    }
    else
    {
      alarmHours=alarmHours+1;
    }
  }
   if(digitalRead(P3)==LOW)
  {
    if(alarmHours==0)
    {
      alarmHours=23;
    }
    else
    {
      alarmHours=alarmHours-1;
    }
  }
  lcd.setCursor(0,0);
  lcd.print("Set HOUR Alarm:");
  lcd.setCursor(0,1);
  lcd.print(alarmHours,DEC);
  delay(200);
 }
 delay(200);
}

void DisplaySetMinuteAll()// Setting the alarm minutes
 {
  while(digitalRead(P1)==HIGH){

  lcd.clear();
  if(digitalRead(P2)==LOW)
```

```
{
  if (alarmMinutes==59)
  {
    alarmMinutes=0;
  }
  else
  {
    alarmMinutes=alarmMinutes+1;
  }
}
 if(digitalRead(P3)==LOW)
{
  if (alarmMinutes==0)
  {
    alarmMinutes=59;
  }
  else
  {
    alarmMinutes=alarmMinutes-1;
  }
}
lcd.setCursor(0,0);
lcd.print("Set MIN. Alarm:");
lcd.setCursor(0,1);
```

```
  lcd.print(alarmMinutes,DEC);
  delay(200);
 }
 delay(200);
}
void printAllOn(){
   lcd.setCursor(0,3);
  lcd.print("Alarm: ");




  if (alarmHours <= 9)
  {
    lcd.print("0");
  }
  lcd.print(alarmHours, DEC);

  lcd.print(":");
  if (alarmMinutes <= 9)
  {
    lcd.print("0");
  }
  lcd.print(alarmMinutes, DEC);
```

```
}
void printAllOff() {
  lcd.setCursor(0, 3);
  lcd.print("Alarm: Off  ");
}
void Alarm(){
   if(digitalRead(P4)== LOW)
  {
   setAll=setAll+1;
  }
  if (setAll==0)
    {
     printAllOff();
     noTone (buzzer);
     digitalWrite(LED,LOW);
     }
  if (setAll==1)
    {


     printAllOn();


     DateTime now = RTC.now();
     if ( now.hour() == alarmHours && now.minute() == alarmMinutes )
        {
```

```
       lcd.noBacklight();

       DateTime now = RTC.now();

       digitalWrite(LED,HIGH);

       tone(buzzer,880); //play the note "A5" (LA5)

       delay (300);

       tone(buzzer,698); //play the note "F6" (FA5)

       lcd.backlight();

      }

    else{

       noTone (buzzer);

       digitalWrite(LED,LOW);

      }


    }

  if (setAll==2)

    {

     setAll=0;

    }

    delay(200);

}
```

DISCUSSION

During this project we had to merge the implementation of operating system with some hardware components by using microcontroller. We decided to build an Alarm Clock that can exhibit time, date, day, year using RTC1307 module and i2c module with the vision to make the compilation easier. For this project we chose ***Arduino Uno*** as our microcontroller since it is inexpensive and does have a simple configuration compared to the other existing microcontrollers available. While working on this real-time alarm and calendar project we faced a couple of major difficulties. To begin with we faced some problems due to our ignorance about jumper wires, as initially we did not the difference between F-M wires and M-M wires which is why we purchased only Male-Male jumper wires at first, but during the working procedures we got to know about their distinct functionalities. Sequentially, while working we also discovered that display, we bought initially was 16x2 lcd which is smaller than our preference, for that we bought bigger display 20x4. For the connection between LCD with Arduino and RTC we also used an i2c module. After the competition of assembling our hardware part, when we successfully compiled our whole project, we faced another problem which we could not overcome. This problem was we could not coincide the second with real time, consequently per second on the display was counted equal to 14 seconds. We opted for many solutions to solve this problem such as: using new RTC module, debugging code, rechecking the circuit connection etc. Later, after much research we could conclude that this problem occurred due to incompatible CPU clock speed and clock time by the Arduino board.

Moreover, throughout the project progression we could associate our theoretical knowledge and practical knowledge. Consequently, we gained enticement to work on versatile hardware projects in the near future.

REFERENCE

- Tittiamo. (2017, Feb 6). *Alarm Clock with Arduino.* Project Hub. https://create.arduino.cc/projecthub/tittiamo68/alarm-clock-f61bad
- Microcontrol 2560. (2020, Jul 12). *Arduino Alarm Clock!! Set your own Alarm!!*. YouTube. https://youtu.be/YGnHV29uOCM
- ENGINEER THIS ELECTRONICS. (2018, Jun 5). *How to make Arduino alarm clock ||[HINDI]*. YouTube. https://youtu.be/YNmdTctBK3U