T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

PROJECT REPORT

CMP 3006 - EMBEDDED SYSTEM PROGRAMMING

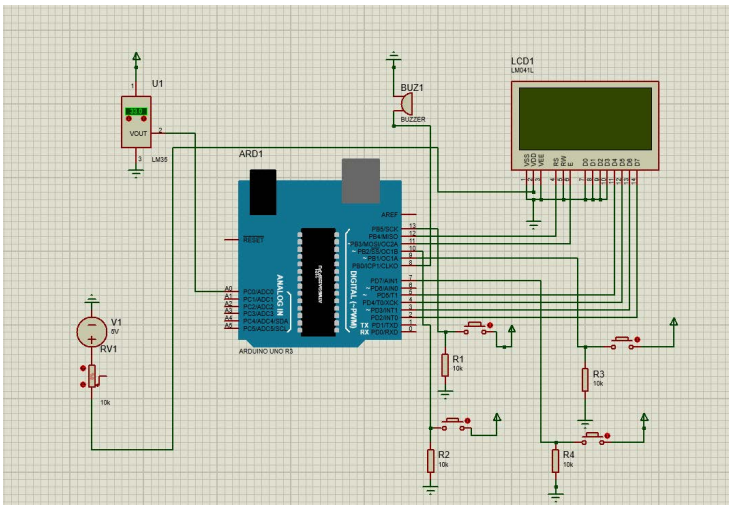
Anıl Şülekoğlu

1602414

İSTANBUL, June 06.06.2020

TABLE OF CONTENTS

Circiut on Proteus



Proteus Design

Components:

- -Arduino uno R3
- - 4x push buttons
- - lcd screen(16x4)
- - 4x 10k ohm Resistor
- - LM35 temprature sensor
- - Buzzer
- - POT-HG potentiometer
- - Power Supply

Arduino Code

Timer Part:

```
cli();
TCNT1= 0;
TCCR1A = 0;
TCCR1B = 0;
TCCR1B |= (1<<WGM12);
TCCR1B |= (1<<CS12)|(1<<CS10);
OCR1A = 15624; ;
TIMSK1 |= (1<<OCIE1A);

lcd.begin(16,4);
sei();
```

=>

Here is the i set the timer. Firstly i enabled interrupt with cli method, then i set up the ctc mode then i set prescaler value at 1024 and my ocr value is 15624, that's why after each 1 second interrupt service routine will work.

```
ISR(TIMER1_COMPA_vect){
 sec++;
 if(sec>=60){
  min++;
  sec=0;

 }
 if(min>=60){
  hour++;
  min=0;
 }
 if(hour>=24){
  hour=0;
 }
 if(al_min>=60){
  al_hour++;
  al_min=0;
 }
 if(al_hour>=24){
  al_hour=0;
 }
```

=>

Here is the Interrupt Service Routine methods, after each second second will be increase and minute and hour value will set to according to this. Also i added the alarm hour and minute, when we change the these value or lated the alarm this will be controlled each one second thanks to ISR.

lcd.print part:

```
lcd.setCursor(6,1);
 if(sec<10)lcd.print("0");

lcd.setCursor(7,1);
if(sec<10)lcd.print(sec);
lcd.setCursor(6,1);
if(sec>=10)lcd.print(sec);

lcd.setCursor(3,1);
 if(min<10)lcd.print("0");

lcd.setCursor(4,1);
if(min<10)lcd.print(min);
lcd.setCursor(3,1);
if(min>=10)lcd.print(min);

lcd.setCursor(5,1);
lcd.print(":");

lcd.setCursor(0,1);
 if(hour<10&&PM_control==0)lcd.print("0");
lcd.setCursor(1,1);
if(hour<10&&PM_control==0)lcd.print(hour);
lcd.setCursor(0,1);
if(hour>=10&&PM_control==0)lcd.print(hour);
```

=>

Here is the the clock to show lcd screen also i controlled PMAM mode here, if PM_control == 1 then PMAM mode is open else normal mode.

=>

```
void setPM_AM(){

if(debounceButton2(buttonState2) == HIGH && buttonState2 == LOW)
  {
   if(PM_control == 0){
      PM_control =1;
      }

      else{
        PM_control =0;
        }


    buttonState2 = HIGH;

  }
  else if(debounceButton2(buttonState2) == LOW && buttonState2 == HIGH)
  {
       buttonState2 = LOW;
    }

 }
```

```
lcd.setCursor(0,1);
if(hour-12<10 && PM_control==1&&hour-12>0)lcd.print("0");
lcd.setCursor(1,1);
if(hour-12<10 && PM_control==1&&hour-12>0)lcd.print(hour-12);
lcd.setCursor(0,1);
if(hour-12 >= 10 && PM_control==1 &&hour-12>0)lcd.print(hour-12);
```

Setting part

=> Here is the hold to button, if i push button more than 3 second F.E at here setAl_min and setAl_hour methods will work

=>

```
if(debounceButton2(buttonState2) == HIGH && buttonState2 == LOW)
  {
   al_min++;
   buttonState2 = HIGH;
  }
  else if(debounceButton2(buttonState2) == LOW && buttonState2 == HIGH)
  {
       buttonState2 = LOW;

  }
```

```
const long intervalButton = 3000;
const long intervalLed = 6000;

unsigned long currentMillis2 = millis();

  if(buttonState3 == HIGH && programState2 == 0){
    buttonMillis2= currentMillis2;
    programState2 = 1;
  }
  else if(buttonState3 == LOW && programState2==1){
   programState2 = 0; //if released early
  }

  if(currentMillis2-buttonMillis2>intervalButton && programState2==1){
    programState2 = 2; //successfully held for 3 seconds
    ledMillis2= currentMillis2;

  setAl_min();
  setAl_hour();
```

This code is set to alarm, when we push button2 al_min will increase. Same logic is included for alarm hour, hour and minute.

<=

```
if(debounceButton3(buttonState3) == HIGH && buttonState3 == LOW)
  {

   buttonState3 = HIGH;
   break;
  }
  else if(debounceButton3(buttonState3) == LOW && buttonState3 == HIGH)
  {
       buttonState3 = LOW;
    }
```

Alarm part

```
if(debounceButton3(buttonState3) == HIGH && buttonState3 == LOW)
  {
    if(al_control == 0){
      al_control =1;

      }

      else{
        al_control =0;
        }

    buttonState3 = HIGH;

  }
  else if(debounceButton3(buttonState3) == LOW && buttonState3 == HIGH)
  {
       buttonState3 = LOW;
    }
```

=>

This code is active to alarm. İf al_control == 1 then alarm is active. Else not active.

To ring buzzer.=>

Also if push button1 during the ringig al_min will increase 5 minute. That's why after 5 minutes alarm will ring again.

```
if(al_control == 1 && al_hour == hour && al_min == min){


 tone(buzzerPin, 262);
 delay(500);
 noTone(buzzerPin);
 delay(500);

 if(debounceButton(buttonState) == HIGH && buttonState == LOW)
{

 buttonState = HIGH;
 al_min = al_min+5;
}
else if(debounceButton(buttonState) == LOW && buttonState == HIGH)
{
     buttonState = LOW;
 }
   }
```

Temperature part

```
    if(temp_control == 0){

  lm35_okunan_deger= analogRead(lm_35);
  analog_sicaklik=(lm35_okunan_deger/1023)*5000;
  digital_sicaklik= analog_sicaklik/10;
     }

     else if (active == 1 &&temp_control == 1){
       digital_sicaklik = ((digital_sicaklik*1.8) + 32);
       active = 0;
       }
  }
```

=> Here i get analog value of sensor from A0 pin and i convert to digital value. Also F to C and C to F controlled here.

if temp_control==1 then we see lcd screen fahreneid value else we see Celcius value also i add active value because when i do not, else if mode work contionusly so i blocked this with active value.

<=

```
void set_temp(){

   buttonState = digitalRead(7);
if(debounceButton4(buttonState4) == HIGH && buttonState4 == LOW)
  {
    if(temp_control == 0){
      temp_control =1;

      active = 1;
      }

      else{
        temp_control = 0;

        active = 1;
        }


    buttonState4 = HIGH;

  }
  else if(debounceButton4(buttonState4) == LOW && buttonState4 == HIGH)
  {
       buttonState4 = LOW;
    }
```

Setting part(breaking While(true))

```
 void setHour(){

   while(true){

   buttonState2 = digitalRead(10);

 if(debounceButton2(buttonState2) == HIGH && buttonState2 == LOW)
   {
     hour++;
     buttonState = HIGH;
   }
   else if(debounceButton2(buttonState2) == LOW && buttonState2 == HIGH)
   {
        buttonState2 = LOW;

   }

     if(debounceButton(buttonState) == HIGH && buttonState == LOW)
   {

     buttonState = HIGH;
     break;
   }
   else if(debounceButton(buttonState) == LOW && buttonState == HIGH)
   {
        buttonState = LOW;
    }


 }
```

=>

When we hold button1 or button3 more than three second, one of the setHour, setMin, setAl_hour or setAl_min will work and we can set up only working value. So i put the this function in while(true) loop and until we push button1 it will work, when we push button1 loop will break.

Mark: Also i explanied some parts in the video because of page limit.