INDIAN INSTITUTE OF TECHNOLOGY PATNA

CS225 - CS226 ARDUINO PROJECT

DIGITAL CLOCK

ARDUINO UNO & 16x2 LCD DISPLAY

Description :

In this Project I designed a **Digital Clock** with **Arduino UNO** and **LCD**, which shows the **Time**, **Date**, **Day**, **Temperature** and **Humidity** in atmosphere. It also has the functions of **Alarm**, **Stopwatch** and **Timer** just like a normal Digital Clock. It has a special feature of **reminding** important and **special dates**. The Time can be set in both **12 or 24 hrs format**. The **temperature** can be displayed in various units such as **°C/°F/K**.

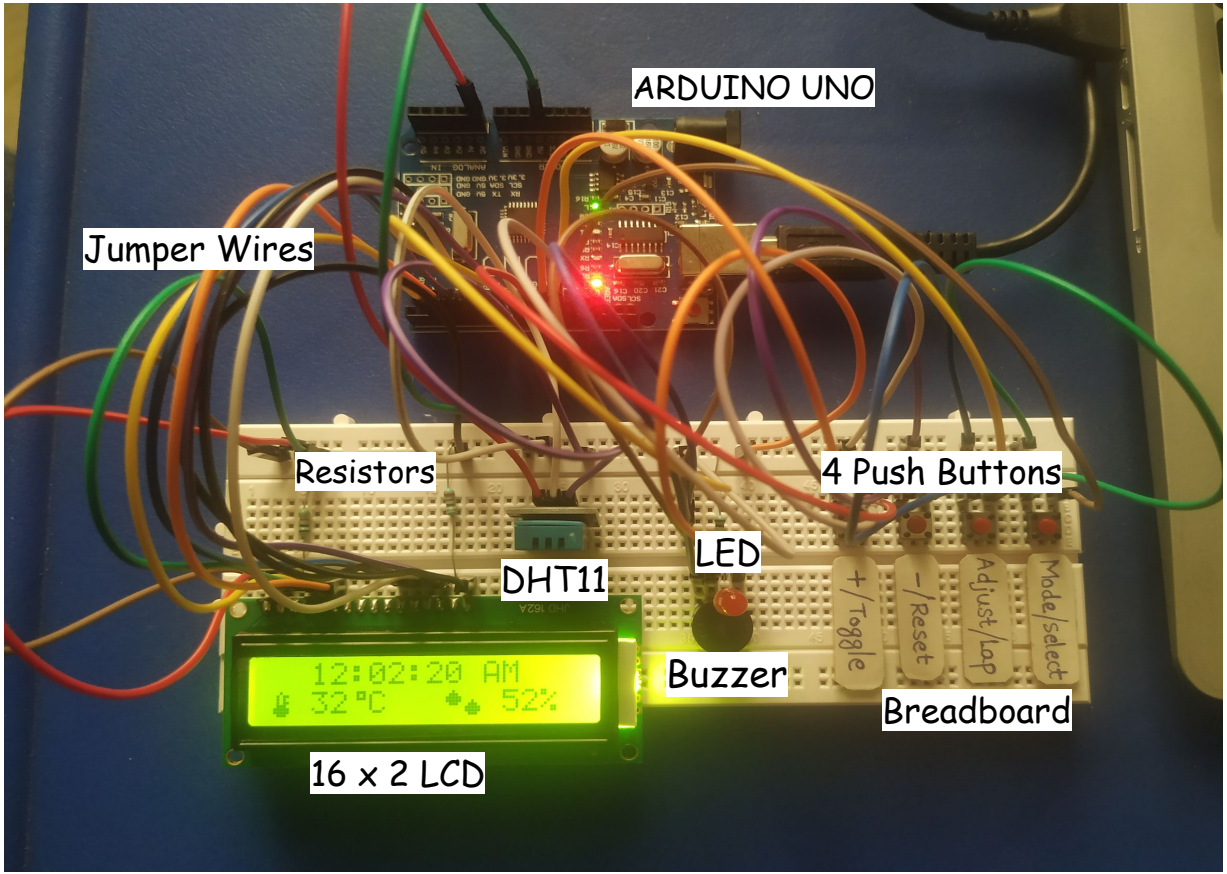
Inputs :

- **4 Push Buttons** - Controls & modifies the settings, like regular digital hand watches.
- **DHT11** - Temperature and Humidity Sensor in the atmosphere.

Outputs :

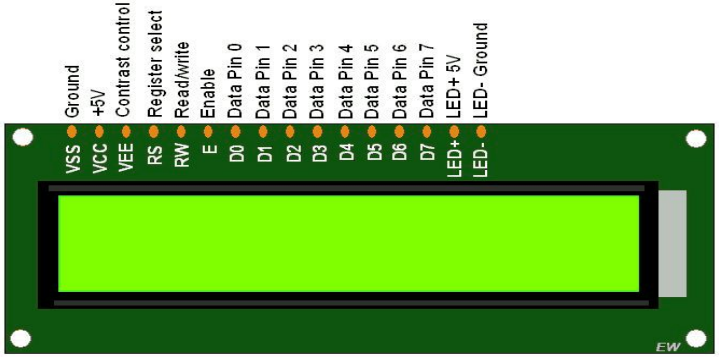
- **16x2 Liquid Crystal Display (LCD)** - Displays the Digital Time and its other modes.
- **Piezo Buzzer** - Makes beeping sounds to notify when a Alarm and Timer is set.
- **LED** - Blinks along with the Buzzer when a Alarm or Timer is set.

Circuit and Components :



Circuit Analysis :

- **LCD Configuration :**



→ Vss pin of LCD is connected to Ground.

→ Vcc pin of LCD is connected to 5V supply.

→ VEE pin is connected to to Ground across a 1k Ohm resistor.

This pin is used to adjust the contrast of the LCD display.

→ RS pin is connected to digital pin 2 of Arduino. This RS pin is used to select the register mode of LCD namely Command register and Data register.

→ R/W pin is connected to Ground, which is used for selecting between read and write modes.

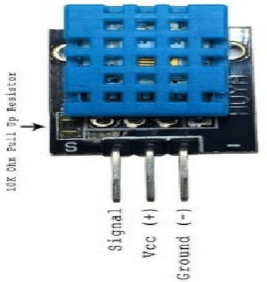
→ E pin is connected to digital pin 3 of Arduino which is meant for enabling the LCD module.

→ Pins DB4 to DB7 are connected to digital pins 4, 5, 6, 7 of Arduino respectively. These are data pins.

→ LED+ pin is connected to 5V supply across a 220 Ohm resistor, which is anode of the back light LED.

→ LED- pin is connected to Ground, which is cathode of the back light LED.

- **DHT11 Configuration :**



→ The GND pin of DHT11 is connected to the ground terminal.

→ The Vcc pin is connected to the 5V supply of the Arduino.

→ The Data pin is connected to Analog Pin A0, which outputs both temperature and

- **Piezo Buzzer Configuration:**

→ The Vcc pin is connected to the digital pin 8 of Arduino, and the GND pin is connecte to the Ground terminal from the Arduino.

- **Push Buttons Configuration:**

→ The 4 push buttons are connected to digital pins of 9, 10, 11, 12 of Arduino with pin modes as INPUT and the other terminals are connected to Ground.

Button - 1) Toggle/+ : Used to increment or toggle any setting in the Digital Clock.

Button - 2) Reset/- : Used to decrement or reset any setting to Start in the Clock.

Button - 3) Adjust/Lap : Used to adjust a particular setting or creating a Lap in Stopwatch.

Button - 4) Mode/Select : Used to select a mode and change the setting accordingly.

Working of the Digital Clock :

The Aim of this Project is to build a Digital Clock which includes the following features:

- **Mode - I** : Displays Time (in 12/24 hr format), Temperature (in °C/°F/K) and Relative Humidity (in %). Setting Time and Changing Formats/Units can be done using the Push Buttons.
- **Mode - II** : Displays Date and Day along with the Scrolling Text if that particular date indicates National Holiday. Setting Date can be done with the help of Push Buttons.
- **Mode - III** : Displays the Alarm Time and the ON/OFF status of the Alarm. A Bell Icon is displayed irrespective of the mode only if the Alarm is ON. The Buzzer for the Alarm rings for 1 - Minute from the moment the Clock Time matches the Alarm Time. Setting Alarm Time, ON/OFF and Stopping Alarm (if ringing) can be done by operating Push Buttons.
- **Mode - IV** : Displays the StopWatch. Start/Stop, Lap(time b/w two splits) and Reset can be done with the use of Push Buttons.
- **Mode - V** : Displays Timer. Set Timer, Start/Stop and Reset can be done by making use of Push Buttons

Software/ Libraries Used:

- C++ Programming Language
- Arduino IDE
- Liquid Crystal Library
- DHT Library

Code Analysis :

```
#include <LiquidCrystal.h>
#include <dht.h>

#define DHT11_apin A0

dht DHT11;

const int rs = 2;
const int en = 3;
const int d4 = 4;
const int d5 = 5;
const int d6 = 6;
const int d7 = 7;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

- **Initially I am importing the Liquid Crystal and DHT libraries**
- **I am creating an Object from the DHT11 class and lcd object from the Liquid Crystal class by declaring the digital pins used to connect to the LCD as described in LCD configuration.**

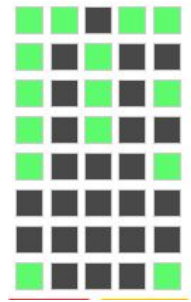
Custom Characters in LCD :

```
byte Temperature_Symbol[8] = {
  B00100,
  B01011,
  B01010,
  B01011,
  B01110,
  B11111,
  B11111,
  B01110
};
```

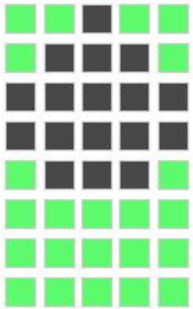
```
byte Dropl[8] = {
  B00100,
  B01110,
  B11111,
  B11111,
  B01110,
  B00000,
  B00000,
  B00000
};
```

```
byte AlarmOn[8] = {
  B00100,
  B01110,
  B01110,
  B01110,
  B11111,
  B00000,
  B00100,
  B00000
};
```

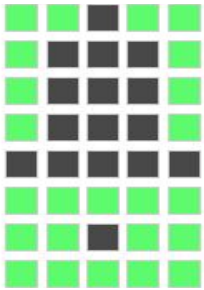
Generated Characters :



Thermometer



Water Drop



Alarm Bell

These Characters are used to display temperature, Humidity and Alarm ON status in LCD.

Set Up function :

```
void setup()
{
  lcd.begin(16, 2);

  lcd.createChar(1,Temperature_Symbol);
  lcd.createChar(2,Degrees);
  lcd.createChar(3,Drop1);
  lcd.createChar(4,Drop2);
  lcd.createChar(5,AlarmOn);

  pinMode(9, INPUT_PULLUP);
  digitalWrite(9, HIGH);
  pinMode(10, INPUT_PULLUP);
  digitalWrite(10, HIGH);
  pinMode(11, INPUT_PULLUP);
  digitalWrite(11, HIGH);
  pinMode(12, INPUT_PULLUP);
  digitalWrite(12, HIGH);

  pinMode(8, OUTPUT);
}
```

- With Setup function, I am initialising the variables and pin modes.
- Using lcd.begin(), I am beginning the lcd screen to display the output.
- Creating the custom generated LCD characters.
- Specifying the Pin modes as input mode and output mode.
- Here I am setting up the pins 9, 10, 11, 12 to input mode, which are connected to 4 push buttons.
- Pin 8 is specified as output mode to connect to the Buzzer.

Mode - 1 : Displays Time (12 and 24 hrs format) and Temperature and Humidity.

```
if(settime == 0)
{
  lcd.setCursor(hrs_pos,0);

  lcd.print("  ");

  delay(500);

  if(digitalRead(9) == LOW)
  {
      hrs++;
      flag++;
  }

  if(digitalRead(10) == LOW)
  {
      hrs--;
      flag--;
  }
}
```

Code on the left side handles the hours setting. If settime==0, then we can adjust the hours value of the Digital Clock Time, either by incrementing or decrementing the Hours by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the hours are in range 1-12/0-23 for respective 12 or 24 hrs format.

```
if(settime == 1)
{
  lcd.setCursor(mins_pos,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW)
  {
      mins++;
  }

  if(digitalRead(10) == LOW)
  {
      mins--;
  }
}
```

Code on the left side handles the minutes setting. If settime==1, then we can adjust the minutes value of the Digital Clock Time, either by incrementing or decrementing the minutes by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the minutes are in the range 0-59.

Mode - 1 (contd..)

```
if(settime == 2)
{
  lcd.setCursor(secs_pos,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW || digitalRead(10) == LOW)
  {
      secs = -1;
  }
}
if(settime == 3)
{
  lcd.setCursor(11,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW || digitalRead(10) == LOW)
  {
      flag = (flag+12)%24;
  }
}
```

Code on the left side handles the seconds setting. If settime==2, then we can adjust the seconds value of the Digital Clock Time to 00 seconds either by pressing the Push buttons connected to the pins 9 and 10.

Code on the left side handles the AM/PM setting in 12 hrs format time. If settime==3, then we can toggle the AM/PM value of the Digital Clock Time, by pressing the Push buttons connected to the pins 9 or 10.

Mode - 1 (Contd..) :

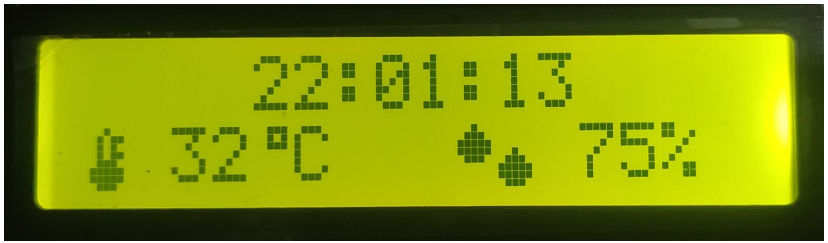
```
void Disp_Temp_RH()
{
    DHT11.read11(DHT11_apin);

    int T_C = DHT11.temperature;
    int T_F = Celcius_to_Fahrenheit(T_C);
    int T_K = Celcius_to_Kelvin(T_C);
    int RH = DHT11.humidity;
    lcd.setCursor(0,1);
    lcd.write(1);
    lcd.print(" ");
    if(T_Unit == 0)
    {
      lcd.print(T_C);
      lcd.write(2);
      lcd.print("C ");
    }
    if(T_Unit == 1)
    {
      lcd.print(T_F);
      lcd.write(2);
      lcd.print("F ");
    }
    if(T_Unit == 2)
    {
      lcd.print(T_K);
      lcd.print("K ");
    }
    lcd.setCursor(9,1);
    lcd.write(3);
    lcd.write(4);
    lcd.print(" ");
    lcd.print(RH);
    lcd.print("%");
}
```

The function Disp_Temp_RH( ) reads the output from the DHT11- Temperature and Humidity sensor through analog pin A0 as an object. This object contains values of both Temperature and Humidity. This function will print the values on the LCD screen with the previously generated custom characters. This function also has a feature of printing the Temperature in desired unit such as Celsius, Fahrenheit or Kelvin scale and it prints the relative humidity in % units.

Time, Temperature & Humidity Display

Changing the Time

Time in 24 hr Format



Changing Temperature Scale °C/°F/K

Mode - 2 : Displays Date(DD/MM/YYYY) and Day and reminds us any special dates.

```
if(setdate==0)
{
  lcd.setCursor(3,0);

  lcd.print("  ");

  delay(500);

  if(digitalRead(9) == LOW)
  {
      dd++;
      if(dd > daysOfMonth[mm])
      {
          dd = 1;
      }
  }

  if(digitalRead(10) == LOW)
  {
      dd--;
      if(dd == 0)
      {
          dd = daysOfMonth[mm];
      }
  }
```

Code on the left side handles the Day(DD) setting. If setdate==0, then we can adjust the day of the Digital Clock Date, either by incrementing or decrementing the Days by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the days are in correspondence with the given month.

```
if(setdate==1)
{
  lcd.setCursor(6,0);
  lcd.print("  ");

  delay(500);

  if(digitalRead(9) == LOW)
  {
      mm++;
      if(mm == 13)
    {
      mm = 1;
    }
  }

  if(digitalRead(10) == LOW)
  {
      mm--;
      if(mm == 0)
      {
          mm = 12;
      }
  }
}
```

Code on the left side handles the month(MM) setting. If setdate==1, then we can adjust the month of the Digital Clock Date, either by incrementing or decrementing the month by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the months are in the range 1-12(Jan-Dec).

Mode - 2 (Contd..)

```
if(setdate==2)
{
  lcd.setCursor(9,0);
  lcd.print("    ");

  delay(500);

  if(digitalRead(9) == LOW)
  {
      yy++;
  }

  if(digitalRead(10) == LOW)
  {
      yy--;
  }

  if(LeapYear(yy))
  {
    daysOfMonth[2]=29;
  }
  else
  {
    daysOfMonth[2]=28;
  }
}
```

Code on the left side handles the Year(YYYY) setting. If setdate==2, then we can adjust the Year of the Digital Clock Date, either by incrementing or decrementing the
Year by pressing the Push buttons connected to the pins 9 and 10 respectively. We will also consider the case of Leap year and adjust the Year accordingly.

Mode - 2 (Contd..) :

```
bool LeapYear(int year)
{
  if((year%4 == 0 && year%100 != 0) || year%400 == 0)
  {
    return true;
  }
  else
  {
    return false;
  }
}

int dayFinder(int d, int m, int y)
{
  static int t[12] = { 0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4 };
  y -= m < 3;
  return ( y + y / 4 - y / 100 + y / 400 + t[m - 1] + d) % 7;
}
```

The function Leap Year( ) tells us whether a given year is Leap year (feb-29 days) or not (feb-28 days). So that we can decide the number of days in that year accordingly.

The function dayFinder( ) helps us to find the Day for a given particular Date. This functions maintains the dependency of Day on a given Date.

Date and Day Display

Special Day Display

Mode - 3 : We can set the Alarm for a desired Time and buzzer beeps.

```
if(alarm == 0)
{
  lcd.setCursor(ahrs_pos,0);

  lcd.print("  ");

  delay(500);

  if(digitalRead(9) == LOW)
  {
      ahrs++;
      aflag++;
  }

  if(digitalRead(10) == LOW)
  {
      ahrs--;
      aflag--;
  }
}
```

Code on the left side handles the Alarm Hours setting. If alarm==0, then we can adjust the alarm hours of the Digital Clock,either by incrementing or decrementing the Alarm Hours in 12/24 hrs format by pressing the Push buttons connected to the pins 9 and 10 respectively.

```
if(alarm == 1)
{
  lcd.setCursor(amins_pos,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW)
  {
      amins++;
  }

  if(digitalRead(10) == LOW)
  {
      amins--;
  }
}
```

Code on the left side handles the Alarm Minutes setting. If alarm==1, then we can adjust the alarm minutes value of the Digital Clock, either by incrementing or decrementing the alarm minutes by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the alarm minutes are in the range 0-59.

Mode - 3 (Contd..) :

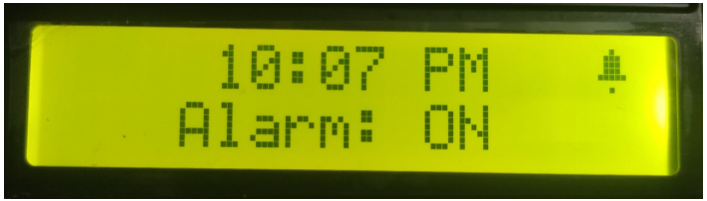
```
if(alarm == 2)
{
  lcd.setCursor(10,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW || digitalRead(10) == LOW)
  {
      aflag = (aflag+12)%24;
  }

if(buzzer)
{
  if(buz_time)
  {
    tone(8, 500, 500);
    buz_time--;
    if((!buz_time) || digitalRead(9) == LOW || digitalRead(10) == LOW || digitalRead(11) == LOW || digitalRead(12) == LOW)
    {
        buzzer = 0;
        buz_time = 60;
    }
  }

  delay(1000);
}
```

Code on the left side handles the AM/PM setting in 12 hrs format time of Alarm. If alarm==3, then we can toggle the AM/PM value of the Digital Clock Alarm, by pressing the Push buttons connected to the pins 9 or 10.

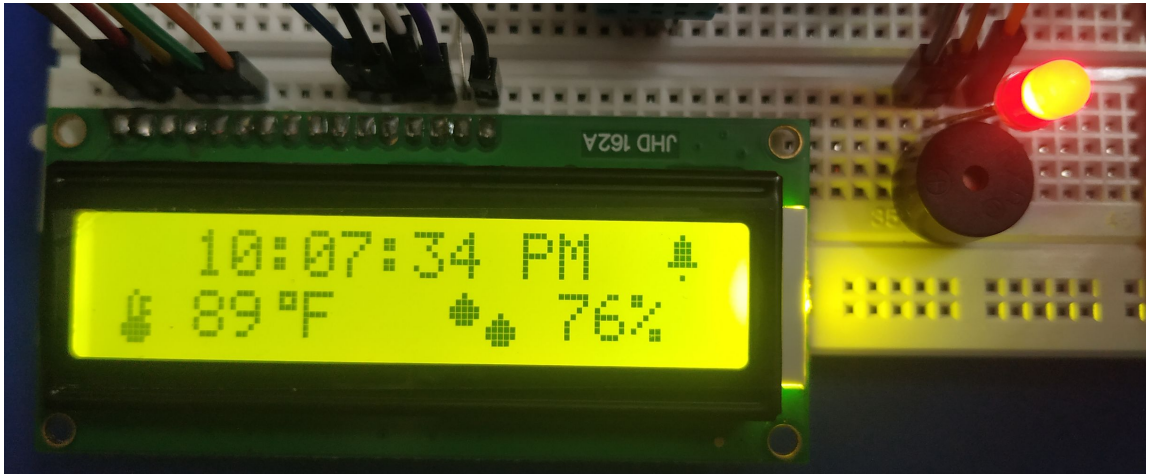
When the Alarm time that we set, matches with the Real Time, the Buzzer beeps with a period of 0.5 seconds, along with LED blinking until we Turn off the Alarm or else for a minute.

Alarm OFF Status

Alarm ON Status (Bell Icon)



Alarm Ringing & LED Blinking



Mode - 4 : Stopwatch with start, stop and reset options along with Lap Time calculator.

```
{
  DispSTW();

  if(digitalRead(9) == LOW)
  {
    sstart = !sstart;
  }
  if(!sstart)
  {
    if(digitalRead(10) == LOW)
    {
       shrs = smins = ssecs = 0;
       lhrs = lmins = lsecs = 0;
       lcd.clear();
    }
  }

  delay(1000);
}
```

Code on the left enables us to start/stop the Stopwatch using the Push button connected to Pin 9 of the Arduino. When we stopped the stopwatch, we can again continue the stopwatch from the same time by pressing start. When the Stopwatch is in the state of 'stop', we can reset the Stopwatch using the push button connected to the pin 10. When we reset the Stopwatch then hours, minutes and seconds all are set to zeros.

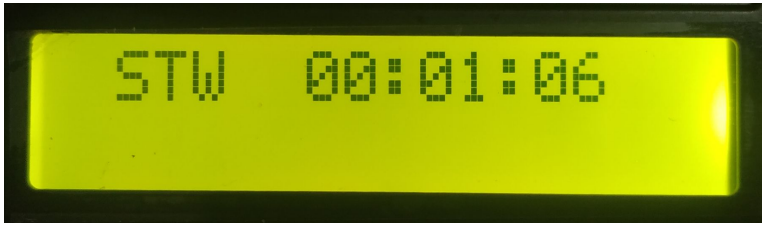
Mode - 4 (Contd..) :

```
ssecs++;
if(ssecs == 60)
{
  ssecs = 0;
  smins++;
}
if(smins == 60)
{
  smins = 0;
  shrs++;
}

lsecs++;
if(lsecs == 60)
{
  lsecs = 0;
  lmins++;
}
if(lmins == 60)
{
  lmins = 0;
  lhrs++;
}
```

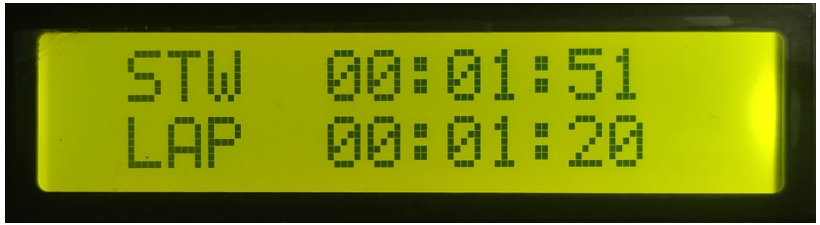
Code on the left increments the time both for the Stopwatch and the Lap. When the seconds reaches 60, it increments the minutes by one, and sets the seconds to zero. Similarly when the minutes reaches 60, it increments the hours by one and sets the minutes to zero. This happens for both Stopwatch time and Lap time.

Stopwatch (Initially)

Stopwatch (running)



Stopwatch & Lap Time



Stopwatch (Reset)

Mode - 5 : Timer which notifies us with buzzer after a desired time period.

```
if(timer == 0)
{
  lcd.setCursor(6,0);

  lcd.print("  ");

  delay(500);

  if(digitalRead(9) == LOW)
  {
      thrs++;
  }

  if(digitalRead(10) == LOW)
  {
      thrs--;
  }
}
```

Code on the left side handles the Timer hours setting. If timer==0, then we can adjust the hours value of the Timer time, either by incrementing or decrementing the Hours by pressing the Push buttons connected to the pins 9 and 10 respectively.

```
if(timer == 1)
{
  lcd.setCursor(9,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW)
  {
      tmins++;
  }

  if(digitalRead(10) == LOW)
  {
      tmins--;
  }
}
```

Code on the left side handles the Timer minutes setting. If timer==1, then we can adjust the minutes value of the Timer Time, either by incrementing or decrementing the minutes by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the Timer minutes are in the range of 0-59.

Mode - 5 (Contd..) :

```
if(timer == 2)
{
  lcd.setCursor(12,0);

  lcd.print("  ");
  delay(500);

  if(digitalRead(9) == LOW)
  {
      tsecs++;
  }

  if(digitalRead(10) == LOW)
  {
      tsecs--;
  }
}
```

Code on the left side handles the Timer Seconds setting. If timer==2, then we can adjust the Timer seconds, either by incrementing or decrementing the seconds by pressing the Push buttons connected to the pins 9 and 10 respectively. We will always ensure that the Timer seconds are in the range of 0-59.

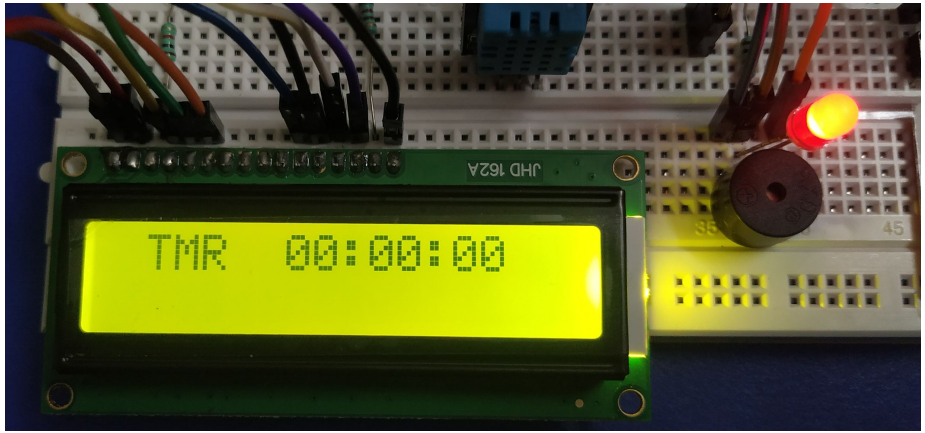
```
if(tstart)
{
  if(thrs == 0 && tmins == 0 && tsecs == 0)
  {
    tone(8, 500, 1000);
    tstart = 0;
  }
  else
  {
    tsecs--;
    if(tsecs == -1)
    {
      tsecs = 59;
      tmins--;
    }
    if(tmins == -1)
    {
      tmins = 59;
      thrs--;
    }
  }
```

When we set the Timer state to 'Start', Code on the left decrements the Timer time and when the Timer time reaches zero, the buzzer beeps for a second and the LED glows notifying us and sets the Timer state to 'OFF'.

Timer (Initially)

Setting Up the Timer

Buzzer Beeps & LED Glows after Timer reaches all 0's



Thank You