Name: Hager Ahmed Zaki **ID:** 7030
Name: Haguar Tarek Dessouky **ID:** 6878

Embedded Final Project

Microcontroller based overheat detector using temperature sensor with Buzzer indication

Project overview:

The purpose of the project is to implement an overheat detection system that alerts a rise in temperature via sounding a buzzer. The buzzer is then stopped momentarily using a serial terminal and the temperature is monitored until it's back in normal range.

Components used:

- Dht11 temperature sensor
- LCD 16x2
- Microcontroller AT89S52
- Potentiometer 10k ohm
- Buzzer
- Transistor 2N3906
- Resistor 330 ohm
- 11.0592 MHz Crystal

Code functions:

1-Temperature sensor functions:

Detecting the temperature with DHT11 is a very specific process that needs an accurate time delay of 20ms and 30us created using the timers in the following manner:

DHT11 is a naturally digital sensor which has no problem being interfaced with the microcontroller as the 8051 have no built in ADC if analogue sensors were to be used instead. In addition, the accuracy ranges in comparison with the LM35 are better and more reliable. Attached below is the manufacturer's datasheet for the error range.

- **Temperature Measurement Range:** 0 ~50
- **Measurement Error:** +~2 degrees
- **Working Voltage:** 3.3 V~5 V
- Output form digital output

```
void timer_delay20ms()  /* Timer0 delay function */
{
 TMOD = 0x01;
 TH0 = 0xB8;  /* Load higher 8-bit in TH0 */
 TL0 = 0x0C;  /* Load lower 8-bit in TL0 */
 TR0 = 1;  /* Start timer0 */
 while(TF0 == 0); /* Wait until timer0 flag set */
 TR0 = 0;  /* Stop timer0 */
 TF0 = 0;  /* Clear timer0 flag */
}
```

```
void timer_delay30us()  /* Timer0 delay function */
{
 TMOD = 0x01;  /* Timer0 model (16-bit timer mode) */
 TH0 = 0xFF;  /* Load higher 8-bit in TH0 */
 TL0 = 0xF1;  /* Load lower 8-bit in TL0 */
 TR0 = 1;  /* Start timer0 */
 while(TF0 == 0); /* Wait until timer0 flag set */
 TR0 = 0;  /* Stop timer0 */
 TF0 = 0;  /* Clear timer0 flag */
}
```

Then, the microcontroller send the request/pulses to detect the temperature using **Request ( )** and wait to receive response from the DHT11 Sensor using **Response( ).**

```
void Request()   /* Microcontroller send request */
{
 DHT11 = 0;  /* set to low pin */
 timer_delay20ms(); /* wait for 20ms */
 DHT11 = 1;  /* set to high pin */
}
```

```
void Response()   /* Receive response from DHT11 */
{
 while(DHT11==1);
 while(DHT11==0);
 while(DHT11==1);
}
```

The last function **Receive_data()** is used to receive the temperature in celsius to be printed later on the lcd for the user.

```
int Receive_data()  /* Receive data */
{
 int q,c=0;
 for (q=0; q<8; q++)
 {
  while(DHT11==0);/* check received bit 0 or 1 */
  timer_delay30us();
  if(DHT11 == 1) /* If high pulse is greater than 30ms */
   c = (c<<1)|(0x01);/* Then its logic HIGH */
  else  /* otherwise its logic LOW */
    c = (c<<1);
  while(DHT11==1);
 }
 return c;
}
```

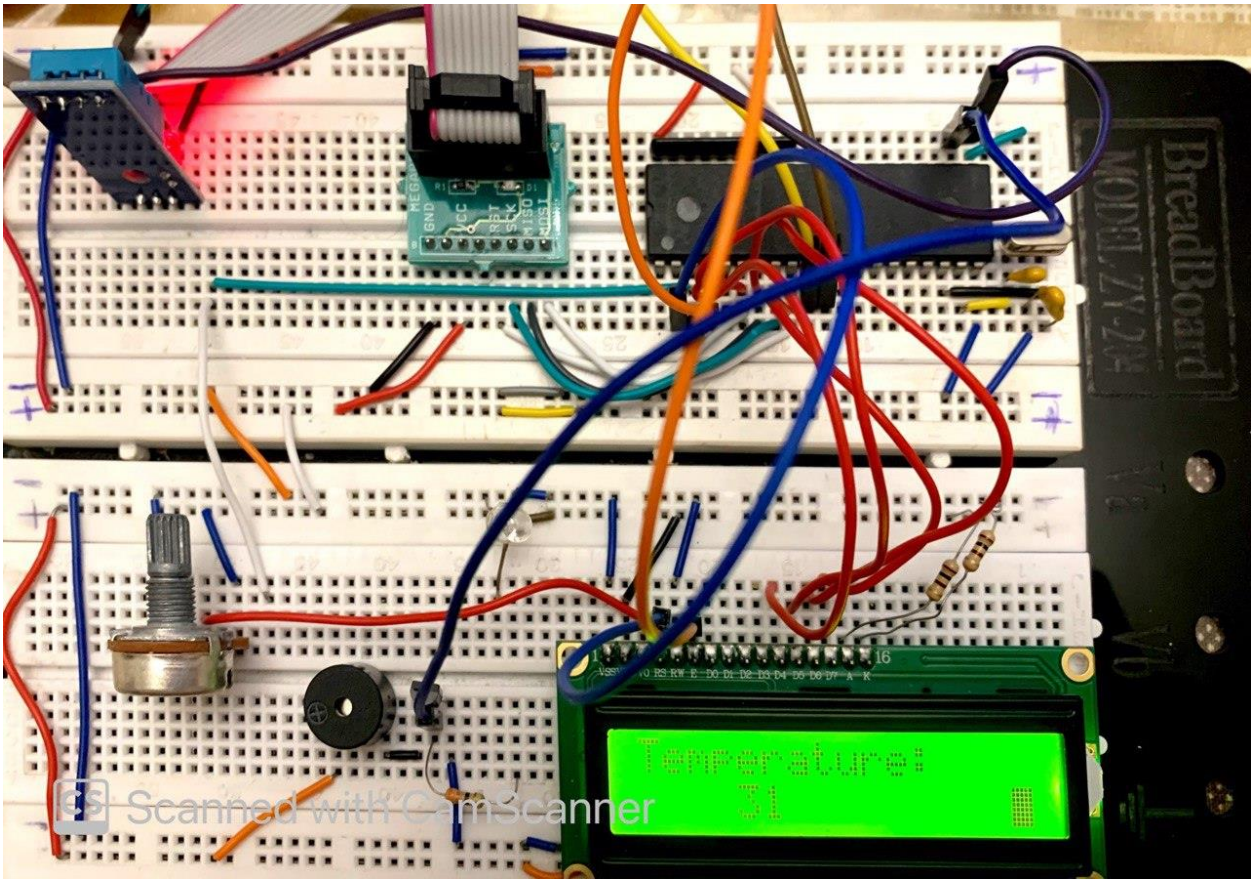
2-Main Code:

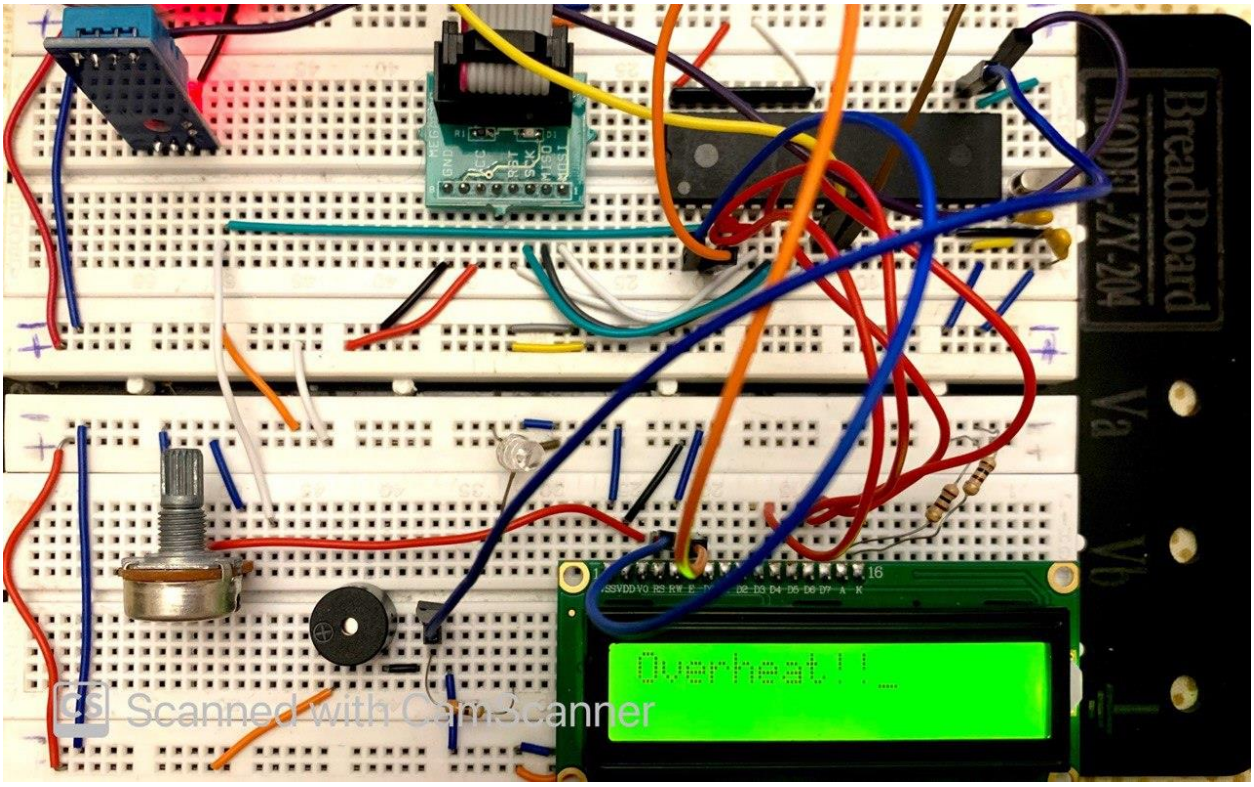
After initializing the LCD, the code starts to execute the main while(1) starting by calling temperature functions to receive the exact temperature of the room in Celsius, displaying then the value on the LCD.

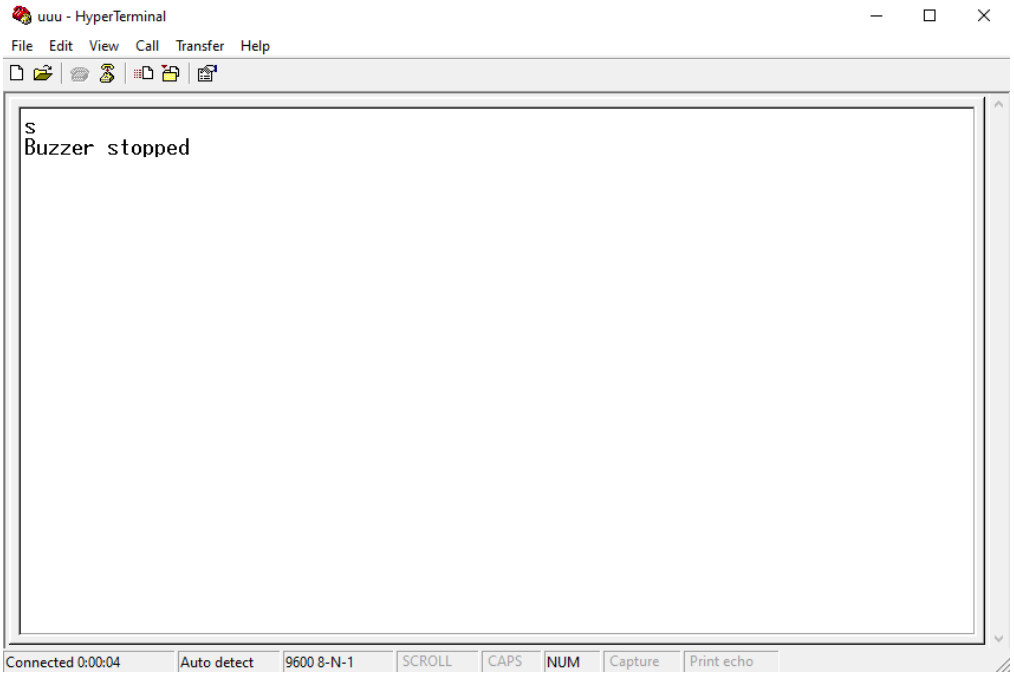
```
void main()
{
 lcd_reset();
 lcd_init();
 lcd_gotoxy_str(0, 0 , "Temperature: ");
 Buzz=1;
 while(1)
 {
  Request(); /* send start pulse */
  Response(); /* receive response */
  I_RH=Receive_data(); /* store first eight bit in I_RH */
  D_RH=Receive_data(); /* store next eight bit in D_RH */
  I_Temp=Receive_data(); /* store next eight bit in I_Temp */
  D_Temp=Receive_data(); /* store next eight bit in D_Temp */
  disp_val(1, 4,I_Temp);
   delay(1000);
```

And then the **if condition** detect overheat, turning on the buzzer if it exceeds 30 degrees Celsius and the code is connected to the HyperTerminal if the user wants to turn off the buzzer.

```
    if(I_Temp>= 32){
   Buzz=0;
      delay(1000);
       lcd_reset();
       lcd_init();
       lcd_gotoxy_str(0,0,"Overheat!!");
   //waiting for command from hyperterminal to stop buzzer
       setup_serial();
   key = __getchar();
     __putchar(key);
      switch(key)
      {
     case 'S':
     case 's':
       Buzz=1;
      lcd_reset();
  lcd_init();
         printString("\r\nBuzzer stopped\r\n");
       lcd_gotoxy_str(0,0,"Buzzer Stopped");
     delay(1000);
     lcd_reset(); lcd_init();
      lcd_gotoxy_str(0, 0 , "Temperature: ");
         break;
       }
 }
}
```


BreadBoard
MODEL ZY-204
Temperature
31
Scanned with CamScanner


BreadBoard
MODEL ZY-204
Overheat!!
Scanned with CamScanner



The buzzer, as mentioned above, is stopped momentarily under the command of 'S' on HyperTerminal, and the temperature is monitored until it falls back to room temperature. While cooling off, the buzzer will still be sounded to indicate the temperature is still high and it is up to the user to command it to stop or not via HyperTerminal.