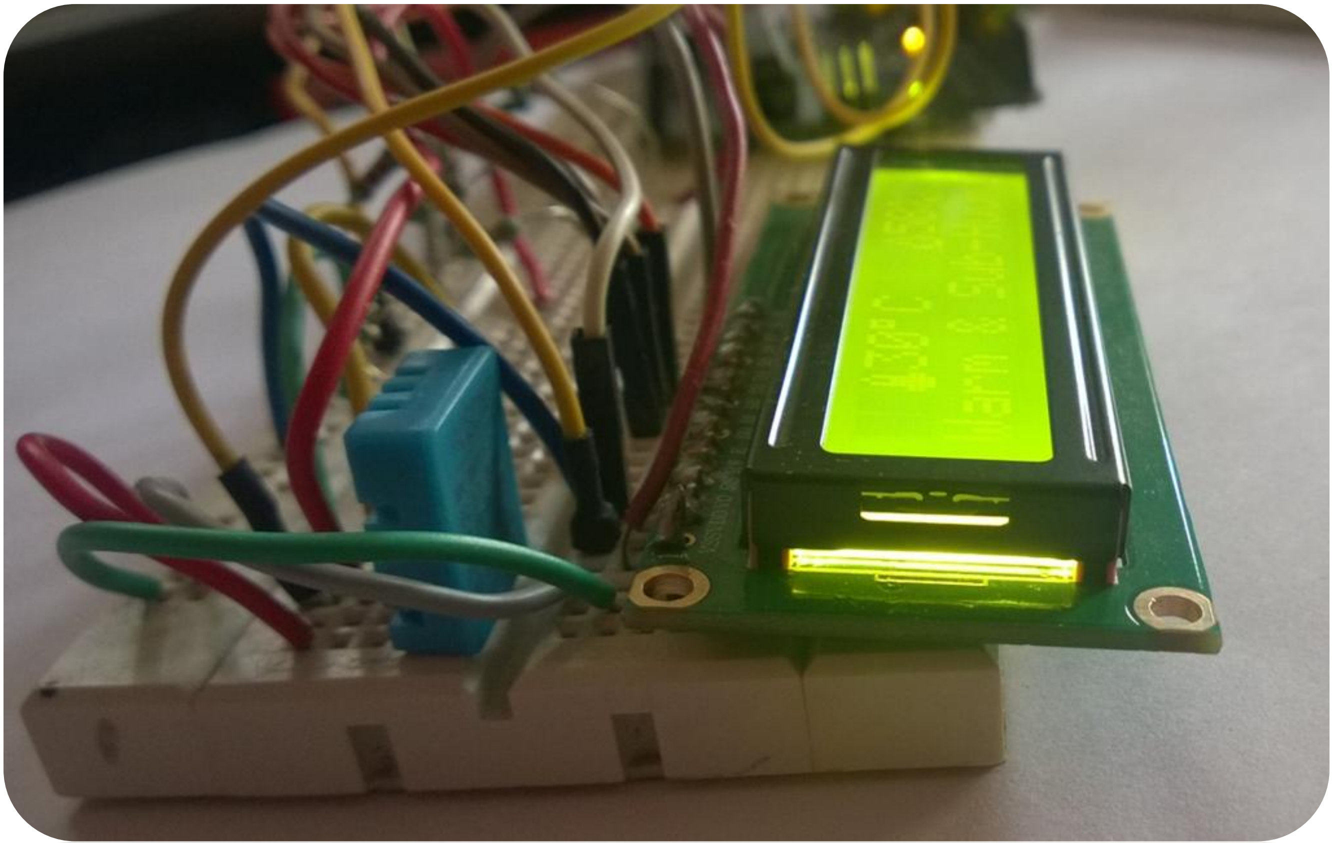
Capstone Project Report

Weather Classifier



Group9

Parmar Shubham A.-14EE30016
Pawa Amey A.-14EE30017

Under the guidance of

Prof. Amit Patra

Contents

1. List of Components……………………………………………………………….....................

2. Introduction……………………………………………………………………………………….

3. Hardware Components………………………………………………………………………….

3.1 Arduino Uno……………………………………………………………………………………….

3.2 Atmega32……………………………………………………………………………………………

3.3 DHT11. ………………………………………………………………………………………………

3.4 LCD 16x2 display………………………………………………………………………………….

4. Design Steps……………………………………………………………………………………………

5. Schematic Diagram……………………………………………………………………………….

5.1 Using Arduino Uno…………………………………………………………………………….

5.2 Using Atmega32…………………………………………………………………………………

6. Arduino Code……………………………………………………………………………………….

7. Atmega Assembly language Code…………………………………………………………….

8. Results……………………………………………………………………..........................

1. List of Components:

- Arduino Uno
- Atmega32
- Dht11 Temperature and Humidity Sensor
- LCD 16X2 Display
- Resistor 10k ohm
- Jumper Wires
- Breadboard

2. Introduction

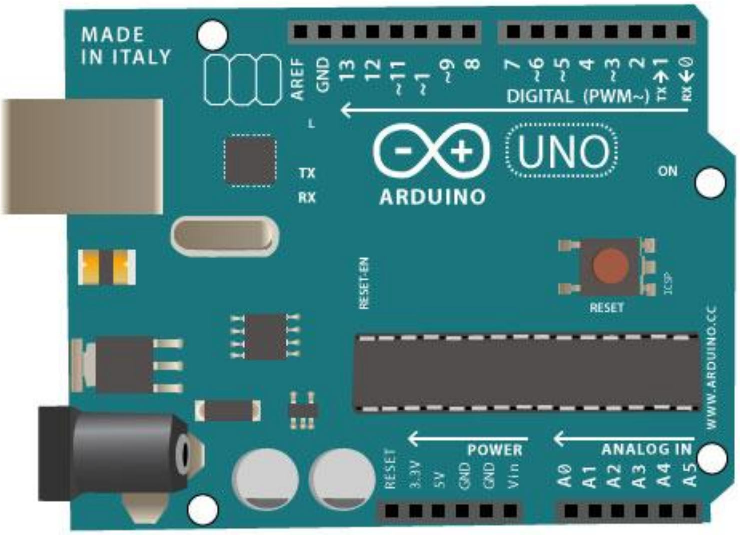
Now a day's temperature is changing rapidly due to global warming, so it is necessary to keep track of it. This will help to know the current weather.

Keeping track of Temperature and Humidity and classify weather is the aim of this project, so that the user can get idea of current weather condition and may predict future weather condition to plan his day.

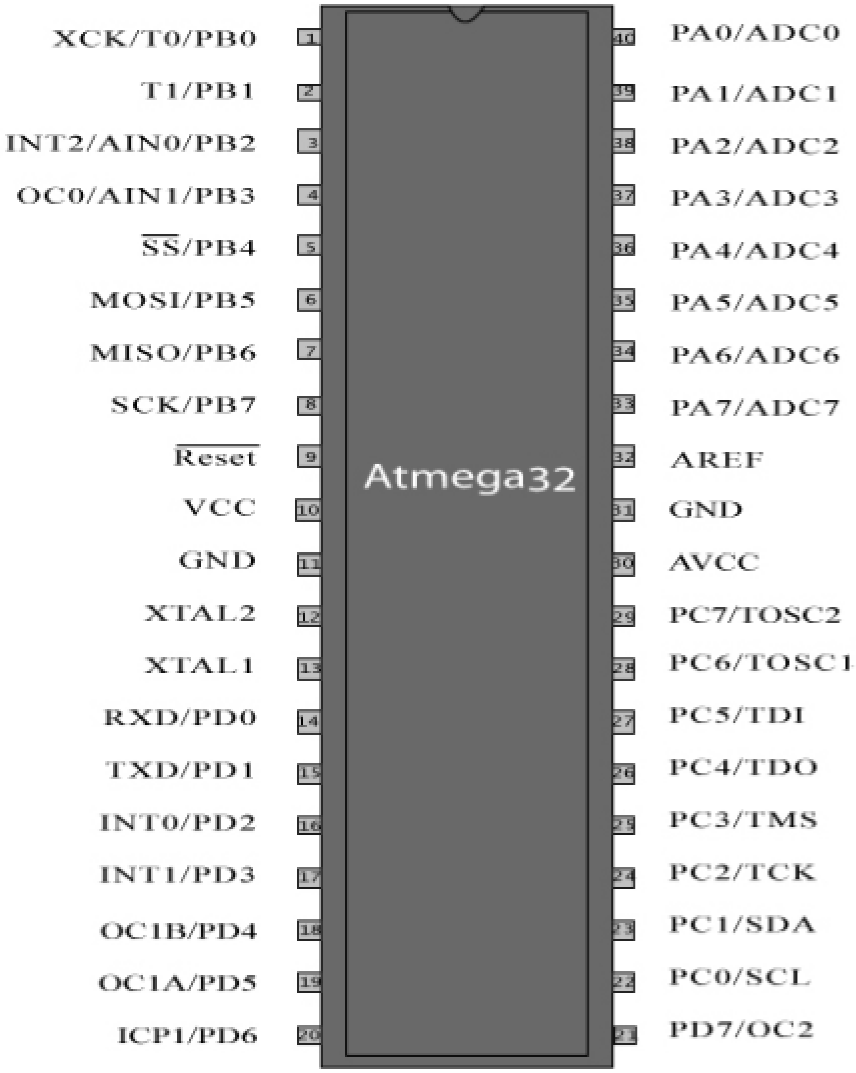
In this project an Arduino based weather classifier has been made where in temperature is classified as hot, warm and cold, and humidity is classified as humid, sub-humid and dry. The temperature and humidity are sensed using DHT11- Temperature and Humidity sensor, which is shown on LCD display.

3. Hardware Components

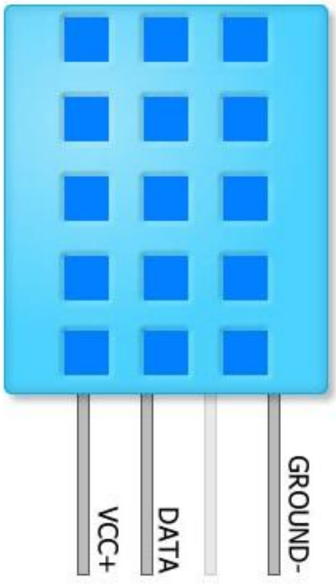
3.1 Arduino Uno



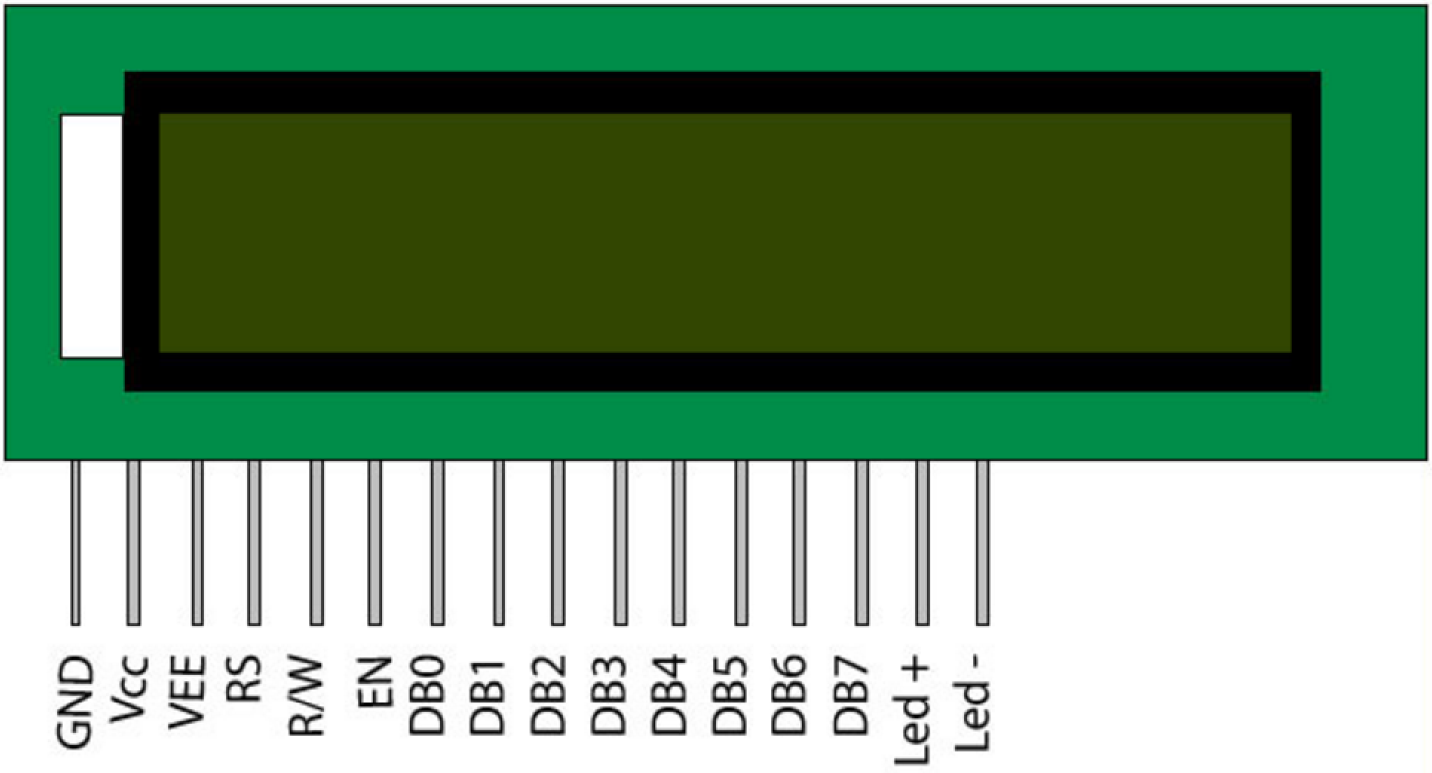
3.2 Atmega32



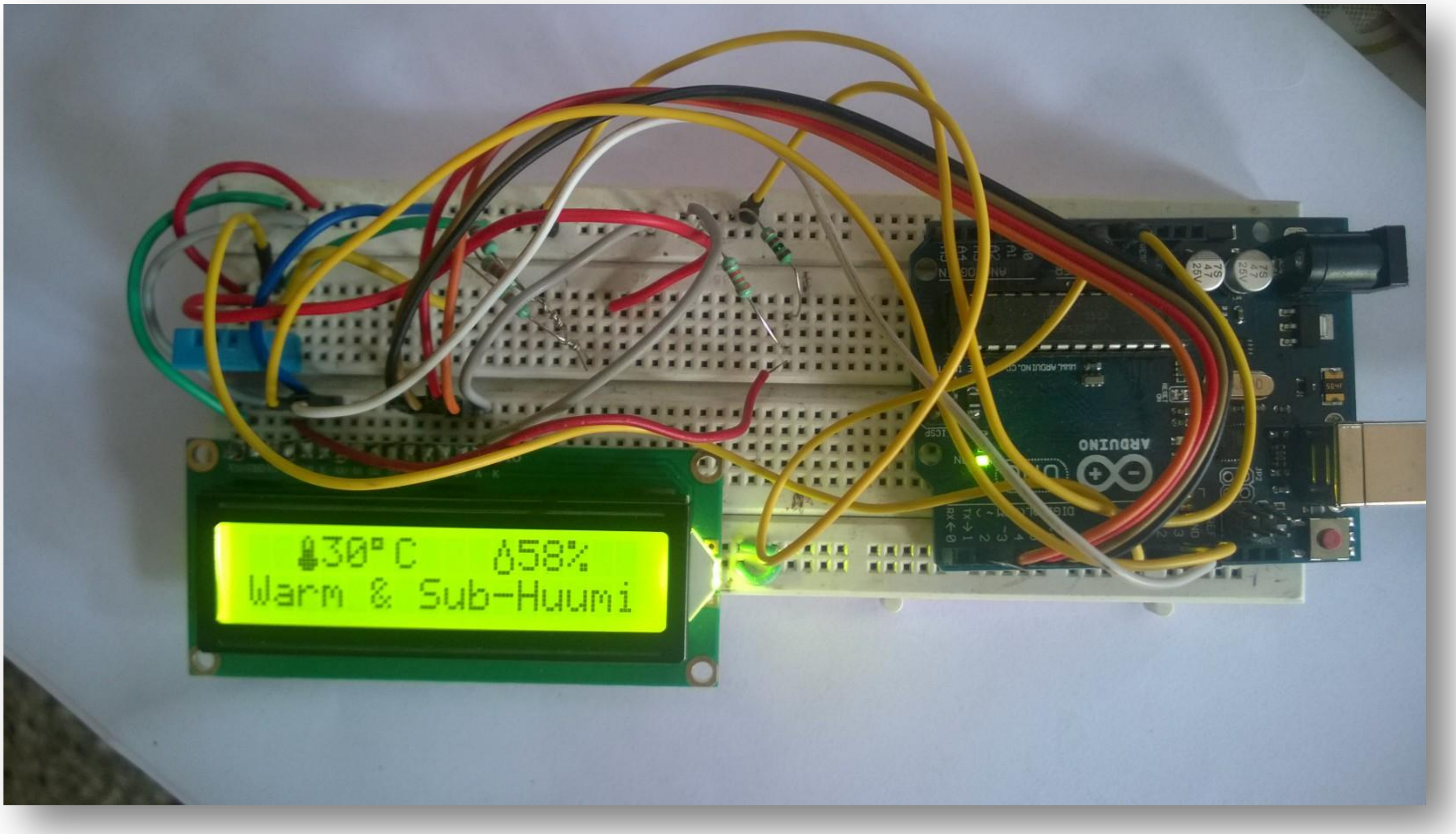
3.3 DHT11



3.4 LCD 16x2 display



4. Design Steps



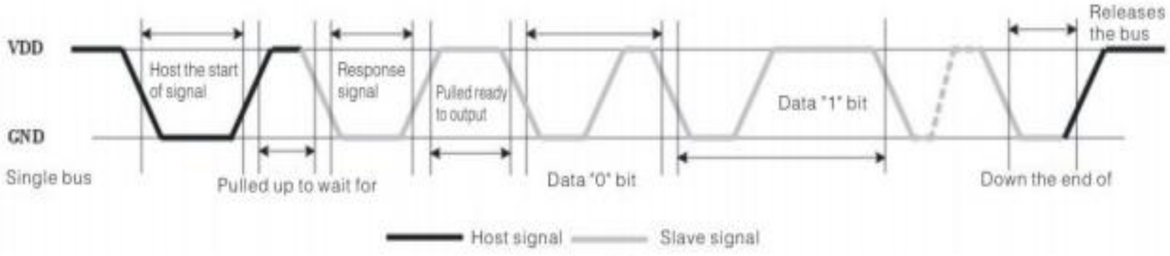
1) Connection between Arduino and sensor

DHT11 sensor works on serial communication its serial communication Data pin is connected to Arduino.

DHT11 sends 40 data bits to Arduino – 16 bits for humidity, 16 bits for temperature and 8 bits of parity

Serial communication Protocol:

User host (MCU) to send a signal, DHT11 converted from low-power mode to high-speed mode, until the host began to signal the end of the DHT11 send a response signal to send 40bit data, and trigger a letter collection. The signal is sent as shown:

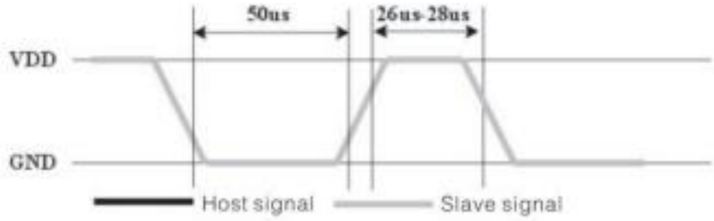


Data Timing Diagram

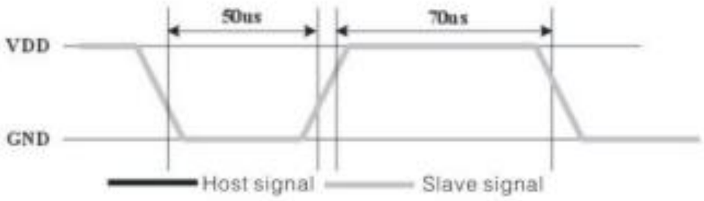
Initially DATA line of DHT11 is pulled-up with high signal from host. Followed by low signal for at-least 18ms is given. On releasing the host signal to the slave, the slave starts to output data in a string of 40 bits.

DHT outputs data through data pin, the microprocessor receives 40 data bits. Data "0" format signal comes as: the low level of 50 microseconds and high for 26-28 microseconds according to the changes in the I/O level, while bit data "1" format comes as: the high level of low plus, 50 microseconds and level high for 70 microseconds.

Bit data "0", "1" signal format as shown:



Bit data "0" bit format



Bit data "1" bit format

2) Connection between Arduino and LCD:

LCD is powered by +5v and ground source by Arduino and 4 bit connection for sending of 4 data bits of 40 received bits.

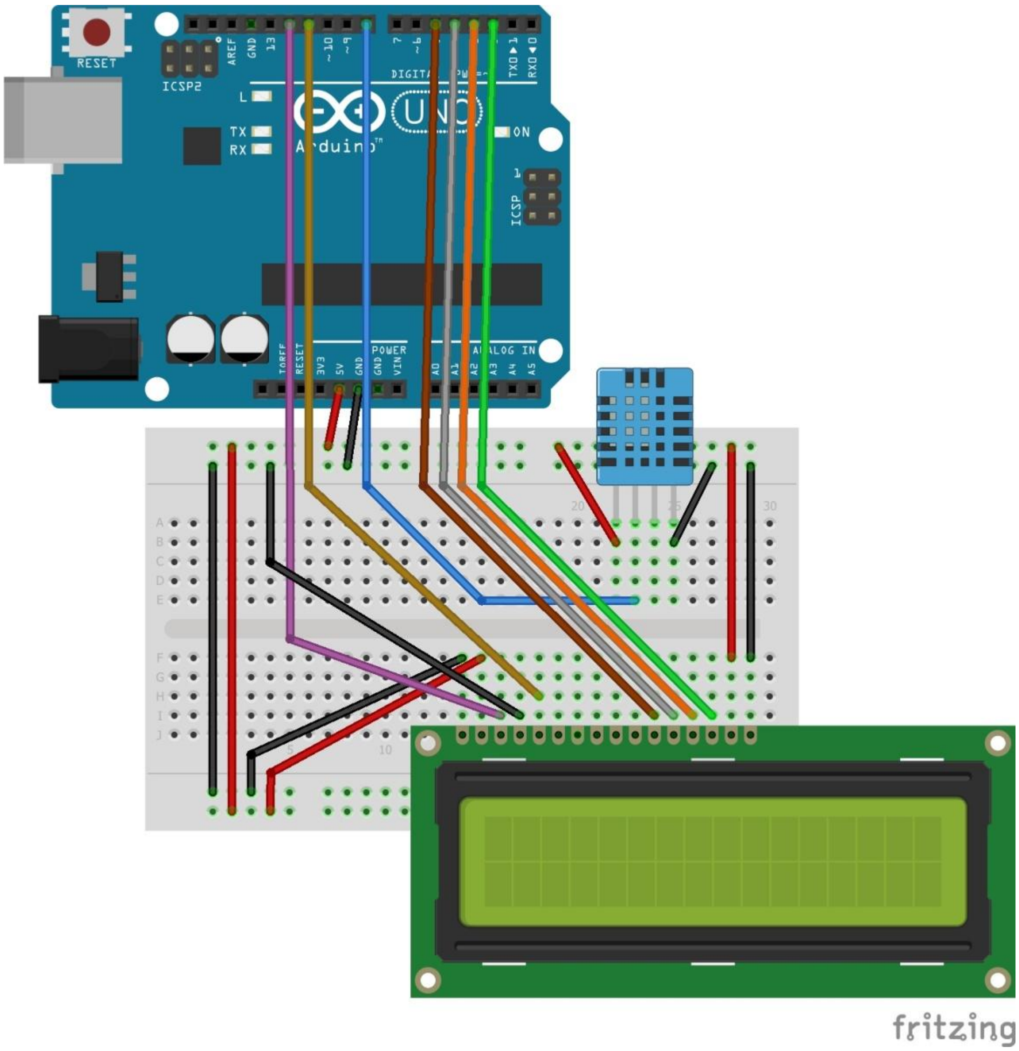
Reference:

1. https://akizukidenshi.com/download/ds/aosong/DHT11.pdf

2. http://playground.arduino.cc/Main/DHT11Lib

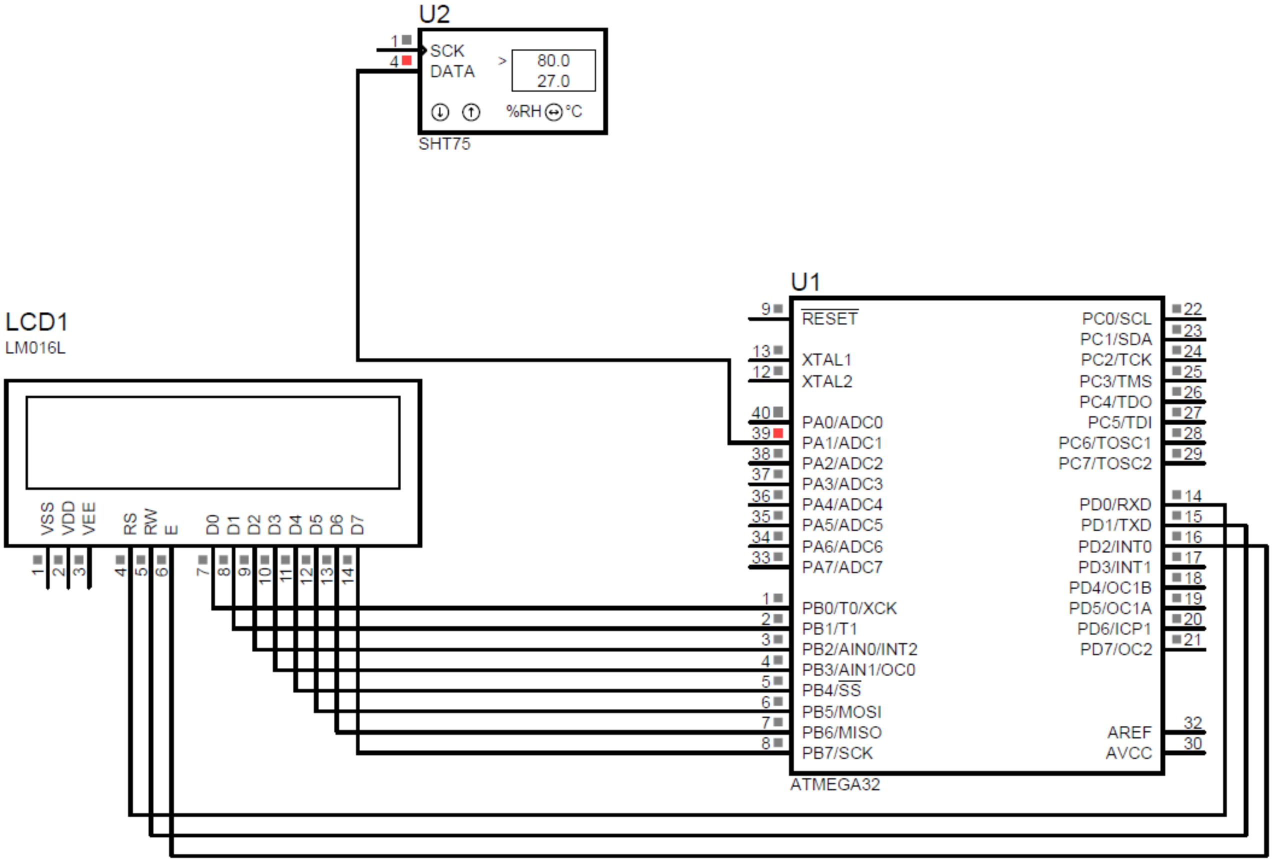
5. Schematic Diagram

5.1 Using Arduino Uno



*created using Fritzing

5.2 Using Atmega32



*created using Proteus 8 Professional

6. Arduino Code

```
byte thermometer[8] = //icon for thermometer
{
   B00100,
   B01010,
   B01010,
   B01110,
   B01110,
   B11111,
   B11111,
   B01110
};

byte droplet[8] = //icon for water droplet
{
   B00100,
   B00100,
   B01010,
   B01010,
   B10001,
   B10001,
   B10001,
   B01110,
};

#include <LiquidCrystal.h>
#include "DHT.h"

// set the DHT Pin
#define DHTPIN 8

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // set up the LCD's number of columns and rows:
  Serial.begin(9600);
  lcd.begin(16, 2);
  dht.begin();

  // Print a message to the LCD.
  lcd.setCursor(0,0);
  lcd.print("Weather Classifier using DHT11 ");
  for (int positionCounter = 0; positionCounter < 14; positionCounter++) {
```

```
  // scroll one position left:
  delay(250);
  lcd.scrollDisplayLeft();
  // wait a bit:
  delay(150);
 }
 lcd.createChar(1,thermometer);        //defining thermometer
 lcd.createChar(2,droplet);          //defining droplet
 lcd.setCursor(15,1);
 lcd.print("Group 9");

 delay(2000);
 lcd.clear();
}

void loop() {
 delay(500);
 // set the cursor to column 0, line 1
 // (note: line 1 is the second row, since counting begins with 0):
 lcd.setCursor(1, 0);

 // read humidity
 float h = dht.readHumidity();

 //read temperature in Fahrenheit
 float f = dht.readTemperature(true);

 //Fahrenheit to Celcius
 float c=(f-32)/1.8;
 if (isnan(h) || isnan(f)) {
  lcd.print("ERROR");
  Serial.println("ERROR");
  return;
 }
 lcd.clear();
 // set the cursor to column 0, line 2
 lcd.setCursor(2, 0);
 lcd.write(1);        // thermometer
 lcd.print(int(c));
 lcd.print((char)223); //degree sign
 lcd.print("C");
 Serial.println(int(f));

 // set the cursor to column 0, line 10
 lcd.setCursor(10, 0);
 lcd.write(2);        // droplet
 lcd.print(int(h));
 lcd.print("%");
```

```
    Serial.println(h);

    lcd.setCursor(0,1);
    //hot, cold, warm / humid, sub-humid, dry
    char *hu,*te;
     //setting limit for humidity
    if(h<=40){
      hu ="Dry";
      }
    else if( h>40 && h<60){
      hu = "Sub-Huumid";
    }
    else {
    hu = "Humid";
    }

  //setting limit for temperature
    if(c<20){
      te ="Cold";
      }
    else if( c>=20 && c<35){
      te = "Warm";
    }
    else {
    te = "Hot";
    }
//Printing Temperature and humidity
    lcd.print( te);
    lcd.print(" & ");
    lcd.print(hu);


  }
```

7. Atmega Assembly language Code

```
.include "M32DEF.INC"

       .EQU DATA_PIN = 1;          PIN1 FOR DATA COMMUNICATION

       .DEF TEMP = R16;
       .DEF BITCNT = R17;
       .DEF CKSUM = R14;
       .DEF DBYTE = R15;
       .DEF RH_DATA_H = R18;
       .DEF RH_DATA_L = R19;
       .DEF T_DATA_H = R20;
       .DEF T_DATA_L = R21;

       .DEF NUMCNT = R24;
       .DEF ZERO_1 = R25;
       LDI ZERO_1,'0';

;******LCD DELCLARATIONS (8 BIT DATA BUS) ************
       .EQU LCD_DPRT=PORTB
       .EQU LCD_DDDR=DDRB
       .EQU LCD_DPIN=PINB
       .EQU LCD_CPRT=PORTD
       .EQU LCD_CDDR=DDRD
       .EQU LCD_CPIN=PIND
       .EQU LCD_RS=0
       .EQU LCD_RW=1
       .EQU LCD_EN=2

       LDI R21 , HIGH(RAMEND);
       OUT SPH , R21;
       LDI R21 , LOW(RAMEND);
       OUT SPL , R21;

       RCALL DELAY_1s;
HERE:
       CALL READ_DHT;
       CALL LCD_INIT;
       RJMP HERE;

;****************************************************************************************************

READ_DHT:
       SBI DDRA, 1;         PORTA PIN1 O/P
       SBI PORTA, 1;        PORTA PIN1 HIGH
       RCALL DELAY_250ms;   PUT HIGH FOR 250ms

       CBI PORTA , 1;              PORTA PIN1 LOW
       RCALL DELAY_18ms;    PUT LOW FOR 18ms
       SBI PORTA, 1;        PORTA PIN1 HIGH
       RCALL DELAY_40us;    HIGH FOR 40us

       SBI PORTA, 1;****************************
       RCALL DELAY_10us;

       CBI DDRA, 1;         PIN1 I/P TO CHECK FOR RESPONSE SIGNAL
       RCALL DELAY_10us;

       SBIC PINA, 1;        SKIPS NEXT LINE IF PIN1 IS LOW
       RJMP ERROR;

       RCALL DELAY_80us;

       LDI TEMP, 0;
       LIRE1:
              SBIC PINA, 1; JUMP TO LIRE2 IF PIN1 IS "SET" ELSE SKIP NEXT LINE
              RJMP LIRE2;
              INC TEMP;
              BRNE LIRE1;          REPEAT IF PIN1 LOW


       LIRE2:
              RCALL DELAY_80us;
              CLR TEMP;

              LIRE2A:
                     SBIS PINA, 1; CHECK FOR END OF RESPONSE SIGNAL, IF LOW MEANS DATA TRANSMISSION HAS
STARTED
                     RJMP LIRE3;          IF PINA LOW JUMP TO LIRE3
                     RJMP LIRE2A;  ELSE REPEAT


       LIRE3:
              CLR CKSUM;
              RCALL ReadByte;
              MOV RH_DATA_H, TEMP;        HIGHER BYTE OF HUMIDITY
              ADD CKSUM, TEMP;

              RCALL ReadByte;
              MOV RH_DATA_L, TEMP;        LOWER BYTE OF HUMIDITY
              ADD CKSUM, TEMP;

              RCALL ReadByte;
              MOV T_DATA_H, TEMP;                HIGHER BYTE OF TEMPERATURE
              ADD CKSUM, TEMP;

              RCALL ReadByte;
              MOV T_DATA_L, TEMP;                LOWER BYTE OF TEMPERATURE
              ADD CKSUM, TEMP;

              RCALL ReadByte;
              CP CKSUM, TEMP;                           PARITY BYTE CHECK
              BRNE ReadDHT4_END;

       RET


;************************************************************************************
       ReadDHT_END:                ;STOP RECIEVING DATA
              CBI DDRA,1;
              RET

       ERROR:
              RCALL DELAY_1s;
              RJMP READ_DHT;

;************************************************************************************

       ReadByte:
              RCALL DELAY_30us;
              LDI BITCNT, 8;
              CLR DByte;

              ReadByteLP:
                     CLR TEMP;
              ReadDHT3:
                     INC TEMP;
                     SBIC PINA, 1;              WAIT FOR DATA TO GO HIGH
                     RJMP ReadDHT3_END;         IF DATA LINE HIGH JUMP TO ReadDHT3_END
                     CPI TEMP, 0;
                     BRNE ReadDHT3;                     ELSE REPEAT ReadDHT3
                     RJMP ReadByteEnd;          WE CAN SET LIMIT TO TIMEOUT AFTER CERTAIN ITERATIONS


              ReadDHT3_END:
                     RCALL DELAY_30us;          DELAY OF 30us, AS DATA LINE GOES HIGH FOR MIN 26-28us,
LATER IT WOULD BE HIGH OR LOW DEPENDING ON THE BIT
                     CLC;                              CLEAR CARRY BIT
                     SBIC PINA, 1;              IF LINE LOW AFTER DELAY, BIT=0
                     SEC;                              ELSE WE SET CARRY WHICH WE USE TO MAKE BIT=1
                     ROL DByte;                        ROTATE LEFT, SHIFT TO BIT 0 OF DByte

                     CLR TEMP;

              ReadDHT4:
                     INC TEMP;
                     SBIS PINA,1;               WAIT FOR DATA LINE TO GO LOW AGAIN, TO DETECT NEST BIT
                     RJMP ReadDHT4_END;

                     CPI TEMP,0;
                     BRNE ReadDHT4;
                     RJMP ReadByteEnd;

              ReadDHT4_END:
                     DEC BITCNT
                     BRNE ReadByteLP;
                     MOV TEMP, DByte;

              ReadByteEnd:
                     RET;

;******************************************************************************************************
**
;*********************LCD INITIALIZATION
LABELS**********************************************************************

LCD_INIT:
              PUSH R22;
              LDI R22,0xFF;
              OUT LCD_DDDR,R22;
              OUT LCD_CDDR,R22;
              CBI LCD_CPRT,LCD_EN;     ;LCD_EN=0
              CALL SDELAY;
              CALL DELAY_2ms;
              POP R22;

              PUSH R16;
              LDI R16,0x38;
              CALL CMNDWRT
              CALL SDELAY
              CALL DELAY_2ms;
              LDI R16,0x0E
              CALL CMNDWRT
              LDI R16,0x01
              CALL CMNDWRT
              CALL SDELAY
              CALL DELAY_2ms;
              LDI R16,0x06
              CALL CMNDWRT
        LDI R16,0x80
              CALL CMNDWRT

              LDI R16, 'T';
              CALL DATAWRT;
              LDI R16, '-';
              CALL DATAWRT;

              MOV R26, T_DATA_H;
              CALL ToASCII;**********
              MOV R16, NUMCNT;
              CALL DATAWRT;
              MOV R16, R17;
              CALL DATAWRT;
              LDI R16, 'C';
              CALL DATAWRT;
              LDI R16, ' ';
              CALL DATAWRT;

              LDI R16, 'H';
              CALL DATAWRT;
              LDI R16, '-';
              CALL DATAWRT;

              MOV R26, RH_DATA_H;
              CALL ToASCII;***************
              MOV R16, NUMCNT;
              CALL DATAWRT;
              MOV R16, R17;
              CALL DATAWRT;
              LDI R16, '%';
              CALL DATAWRT;

              POP R16;



;*********************LCD LABELS**********************************************************************

CMNDWRT:
       OUT LCD_DPRT,R16
          CBI LCD_CPRT,LCD_RS
          CBI LCD_CPRT,LCD_RW
          SBI LCD_CPRT,LCD_EN
          CALL SDELAY
          CBI LCD_CPRT,LCD_EN
          CALL DELAY_100us
          RET
DATAWRT:
       OUT LCD_DPRT,R16
          SBI LCD_CPRT,LCD_RS
          CBI LCD_CPRT,LCD_RW
          SBI LCD_CPRT,LCD_EN
          CALL SDELAY
          CBI LCD_CPRT,LCD_EN
          CALL DELAY_100us
          RET
SDELAY:
      NOP
         NOP
         NOP
         RET

DELAY_100us:
       PUSH R17
          LDI R17,60
  DR0: CALL SDELAY
       DEC R17
          BRNE DR0
          POP R17
          NOP
          NOP
          RET
DELAY_2ms:
       PUSH R17
          LDI R17,20
```

```
   LDR0: CALL DELAY_100us
       DEC R17
          BRNE LDR0
          POP R17
          NOP
          NOP
          RET





;*******************************************************************************************************
*****************Delay Code*****************************************************************************
DELAY_1ms:
       PUSH R21;
       LDI R21 , 248;
       LOOP1:
              DEC R21;
              NOP
              BRNE LOOP1;
       POP R21;
       NOP
       NOP
RET


DELAY_18ms:
       PUSH R22;
       LDI R22, 18;
       LOOP2:

              PUSH R21;
              LDI R21 , 248;
              LOOP1A:
                     DEC R21;
                     NOP
                     BRNE LOOP1A;
              POP R21;

              NOP
              DEC R22;
              BRNE LOOP2;
       NOP
       NOP
       POP R22;
       RET


DELAY_250ms:
       PUSH R22;
       LDI R22, 248;
       NOP
       NOP
       LOOP3:
              CALL DELAY_1ms;
              DEC R22;
              NOP
              BRNE LOOP3;

       NOP
       NOP
       POP R22;
       RET
```

```
Delay_10us:
       PUSH R21;     2CLK
       LDI R21,1;    1CLK
       DEC R21;      1CLK
       POP R21;      2CLK
       NOP
       NOP
       NOP
       RET;          4CLK



Delay_40us:
       PUSH R21;     2CLK
       LDI R21,10;   1CLK
       LP1:
              DEC R21;      1CLK
              BRNE LP1;
       POP R21;      2CLK
       NOP
       NOP
       NOP
       RET;          4CLK


DELAY_30us:
       PUSH R21;     2CLK
       LDI R21,7;    1CLK
       LP2:
              DEC R21;      1CLK
              BRNE LP2;
       NOP;
       POP R21;      2CLK
       NOP
       NOP
       NOP
       RET;          4CLK


DELAY_80us:
       PUSH R21;     2CLK
       LDI R21,24;   1CLK
       LP3:
              DEC R21;      1CLK
              BRNE LP3;
       POP R21;      2CLK
       NOP
       NOP
       NOP
       RET;          4CLK


DELAY_1s:                   ; For CLK(CPU) = 1 MHz
    PUSH R21;
       PUSH R22;
       PUSH R23;
       LDI R21, 8       ; One clock cycle;
       Delay1:
              LDI R22,   125     ; One clock cycle
       Delay2:
              LDI R23,   250     ; One clock cycle
       Delay3:
              DEC R23            ; One clock cycle
              NOP                     ; One clock cycle
              BRNE Delay3          ; Two clock cycles when jumping to Delay3, 1 clock when continuing
to DEC
```

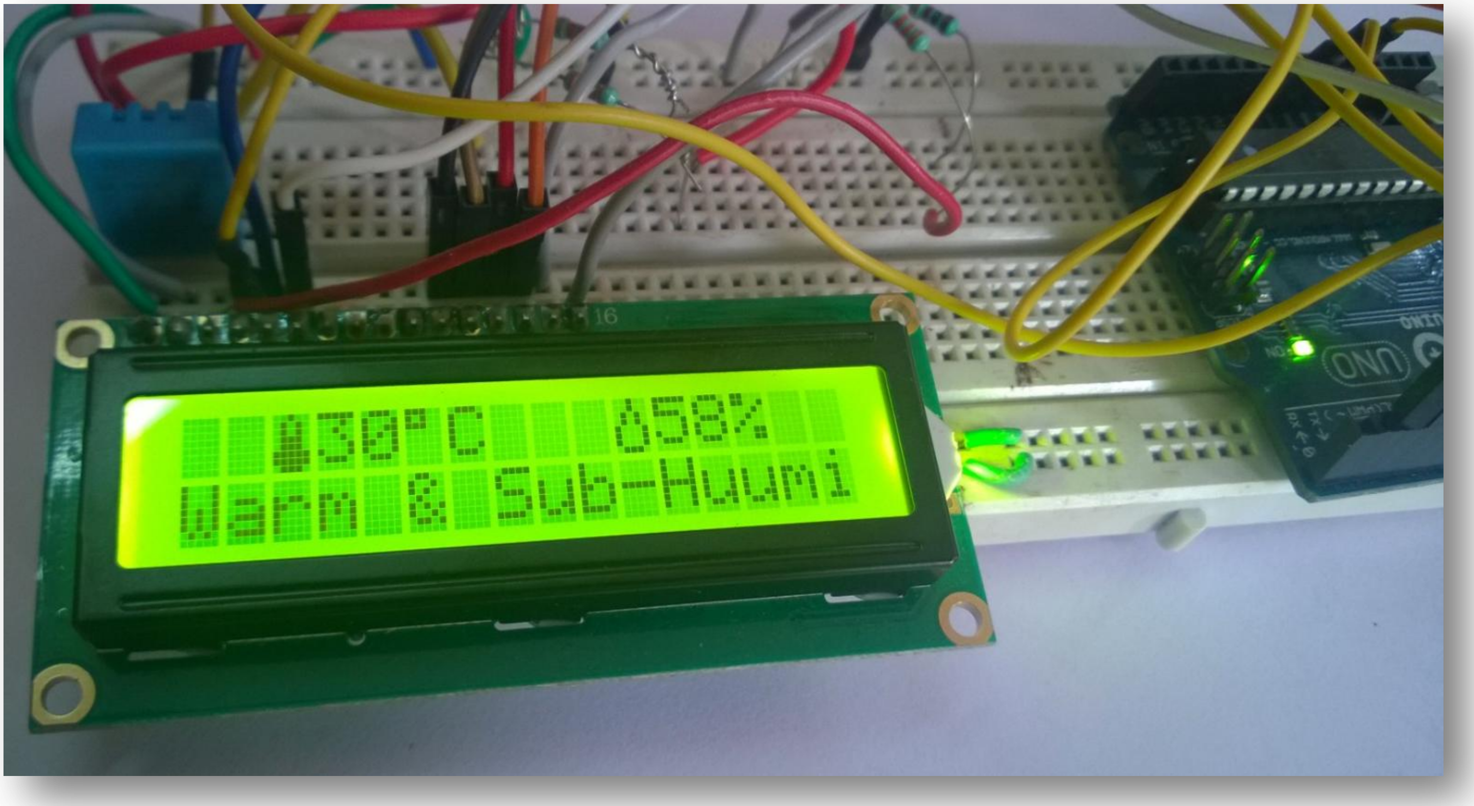
```
              DEC R22             ; One clock cycle
              BRNE Delay2           ; Two clock cycles when jumping to Delay2, 1 clock when continuing
to DEC

              DEC R21            ; One clock Cycle
              BRNE Delay1           ; Two clock cycles when jumping to Delay1, 1 clock when continuing
to RET
POP R23;
POP R22;
POP R21;
RET


;********************************************************************************************************
****************************code for number to ASCII Conversion

ToASCII:
       CLR NUMCNT;
       MOV R17,R26;
       LOOP11:
       CPI R17,10;
       BRGE LINE1;
       RJMP LINE2;
       LINE1:
              SUBI R17,10;
              INC NUMCNT;
              RJMP LOOP11;
       LINE2:
       ADD NUMCNT, ZERO_1;
       ADD R17, ZERO_1;
       RET;
```

Video Link:

https://www.youtube.com/watch?v=HaaHxZingJg

8. Results

Indoor Condition:



Outdoor Condition: