Experiment No.: Exp2

Interfacing a Digital Sensor and Acquiring Data (DHT11 Temperature & Humidity

Objective

To interface a digital sensor with an IoT board and acquire real-time environmental data (temperature & humidity).

Aim

To read **temperature and humidity** values from the **DHT11** sensor using **Arduino UNO** and display the readings on the **Serial Monitor**. Also, control an **LED** based on temperature (hardware output).

Components Used

1. Arduino UNO (or NodeMCU can also be used, but UNO is simplest)
2. DHT11 Sensor (Digital Temperature & Humidity)
3. LED (any color)
4. 220Ω resistor (for LED)
5. Breadboard
6. Jumper wires
7. USB cable (for Arduino + Serial Monitor)

Circuit Diagram

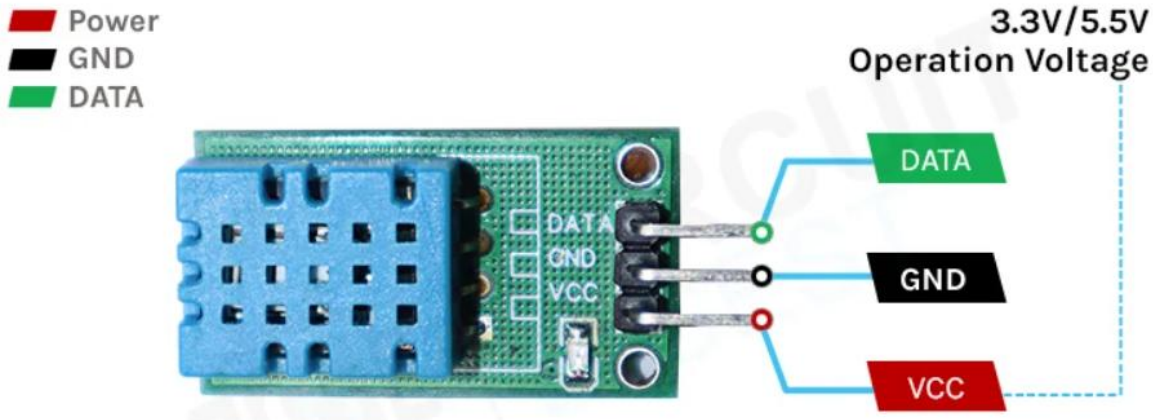
Connections (Arduino UNO → DHT11)

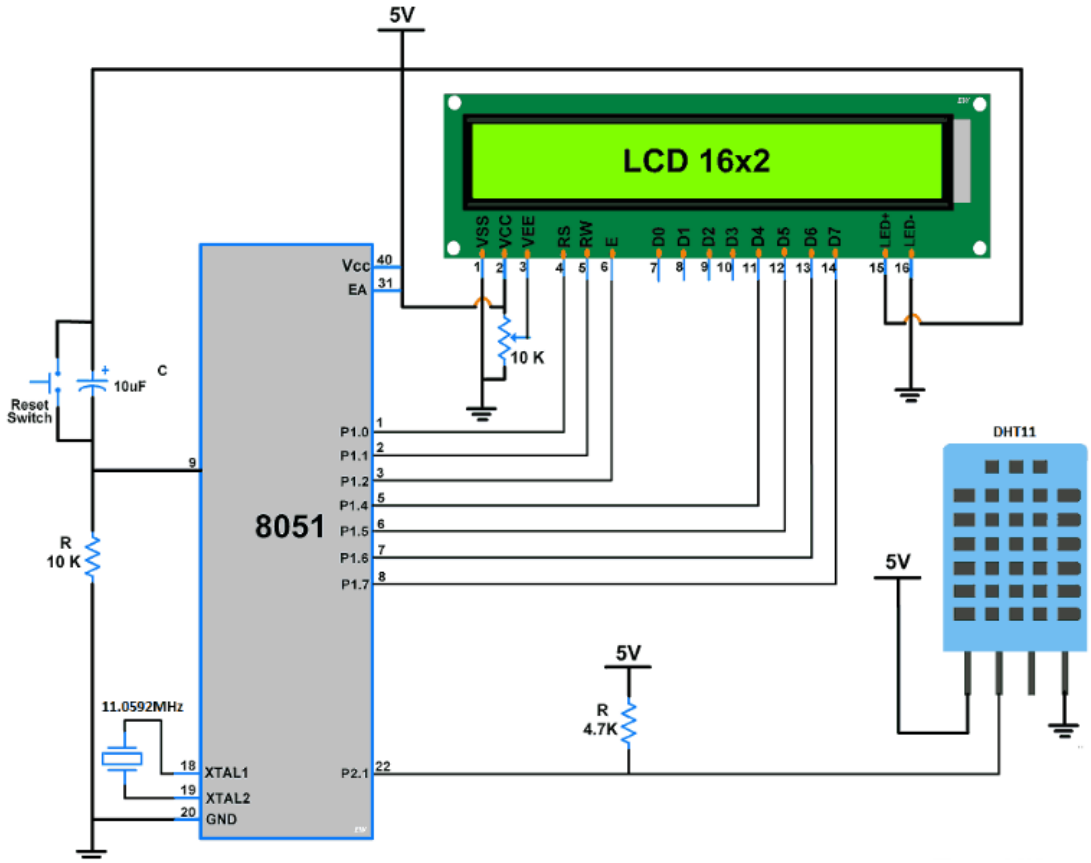
- **DHT11 VCC → 5V**
- **DHT11 GND → GND**
- **DHT11 DATA → D2**

LED Connection

- **Arduino D8 → 220Ω resistor → LED (+)**
- **LED (–) → GND**

ASCII Circuit Diagram





DHT11 Sensor Interfacing with 8051

- The above circuit diagram shows the interfacing of 8051 with the DHT11 sensor.

Code (Arduino UNO)

Install Library: Arduino IDE → Library Manager → install **"DHT sensor library by Adafruit"** + **Adafruit Unified Sensor**

```
#include<reg51.h>
#include<stdio.h>
#include<string.h>
#include <stdlib.h>
#include "LCD16x2_4bit.h"

sbit DHT11=P2^1;                    /* Connect DHT11 output Pin to P2.1 Pin */
int I_RH,D_RH,I_Temp,D_Temp,CheckSum;


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 mode1 (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 */
}

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);
}

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;
}


void main()
{
                unsigned char dat[20];
                LCD_Init();                     /* initialize LCD */

                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 */
                        CheckSum=Receive_data();/* store next eight bit in CheckSum */

                        if ((I_RH + D_RH + I_Temp + D_Temp) != CheckSum)
                        {
                                LCD_String_xy(0,0,"Error");
                        }

                        else
                        {
                                sprintf(dat,"Hum = %d.%d",I_RH,D_RH);
                                LCD_String_xy(0,0,dat);
                                sprintf(dat,"Tem = %d.%d",I_Temp,D_Temp);
                                LCD_String_xy(1,0,dat);
                                LCD_Char(0xDF);
                                LCD_String("C");
                                memset(dat,0,20);
                                sprintf(dat,"%d  ",CheckSum);
                                LCD_String_xy(1,13,dat);
                        }
```

```
                    delay(100);
            }
}
```

Hardware Output (LCD)

Case A: Normal reading
LCD Line 1:

Hum = 55.0

LCD Line 2:

Tem = 29.0°C

(At the end of 2nd line it also prints checksum at column 13, like 123)

So the second line may look like:

Tem = 29.0°C 123

Case B: If sensor communication fails / checksum mismatch
LCD shows:

Error

Conclusion

The **DHT11 digital sensor** was successfully interfaced with **Arduino UNO**. Temperature and humidity values were acquired and displayed on the **Serial Monitor**, and a **hardware action (LED control)** was performed based on the temperature threshold.