**HUMIDITY SENSOR USING 8051**

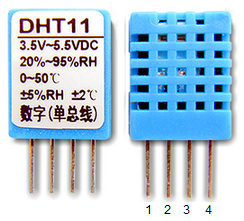
**INTRODUCTION:**

This project is about a simple humidity sensor based on 8051 microcontroller. Humidity sensor is also called ‘Hygrometer’. The humidity information is displayed on a 16×2 LCD display. A relay is also provided which is set to be active when the humidity crosses a certain trip point. The circuit is mains operated and it is very easy to install. DHT11 is the humidity sensor used here.

**8051 MICROCONTROLLER:**

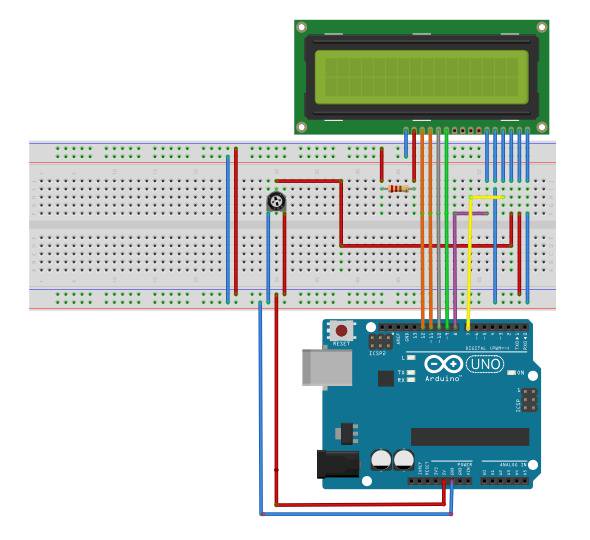
* A smaller computer.
* Consists of RAM, ROM, I/O devices, counters and clock circuits.
* Optimized for specific applications.
* Consists of all features of a microprocessor as well as the features required to build a complete computer.
* First 8 bit microcontroller, 8048 was introduction by Intel in 1976.
* Belongs to MCS51 family was designed for 8 bit mathematical and single   
  Boolean operations.
* They provide high speed of operation and have low system cost.

**DHT11 HUMIDITI SENSOR:**

[](http://www.circuitstoday.com/wp-content/uploads/2015/09/DHT11-pinout.png)DHT11 is a low cost humidity cum temperature sensor which has a digital output. Capacitive method is used for sensing the humidity and a thermistor is used for measuring the temperature. The sensor can sense relative humidity from 20% to 95% at a resolution of 5%. Temperature measurement is up to 50°C at a resolution of 2°C. The communication with the microcontroller is through a single wire. The to and fro communication with DHT11 sensor is very easy. Pin 2 of the DHT11 is connected to the port pin of the microcontroller.

**LCD 16\*2 A**

The Arduino IDE has a built-in library (LiquidCrystal.h) that supports LCDs based on the Hitachi HD44780 chipset (and/or ones that are compatible with it). Other than displaying text on an LCD, the library can also handle printing floating point numbers to a specified amount of decimal places, which makes things easier for the developer.



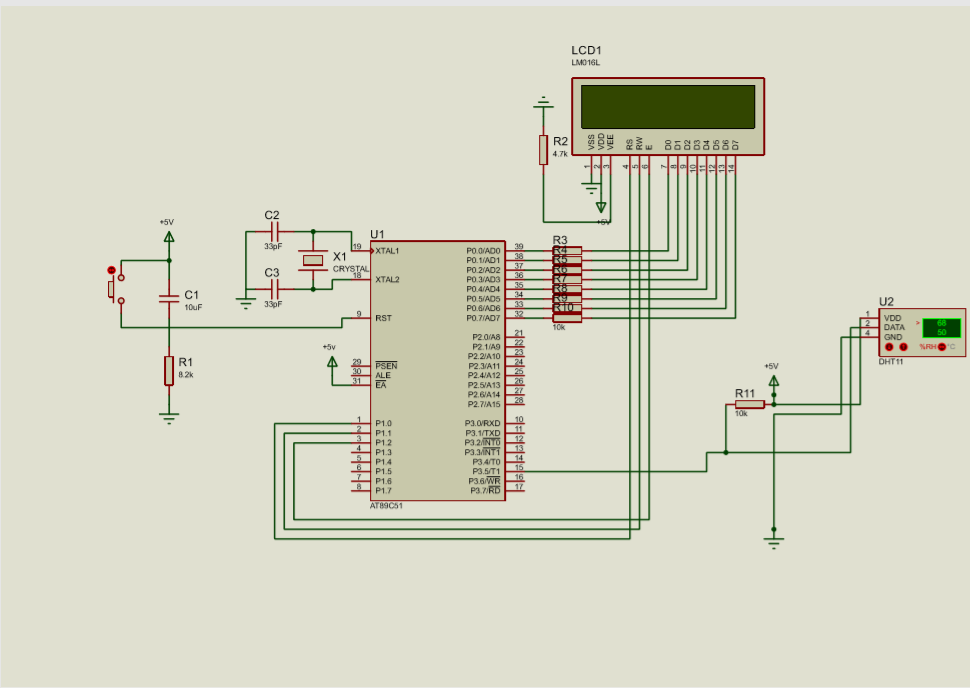
For example:

double Pi = 3.1415926535;

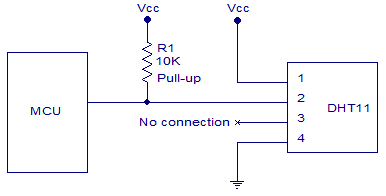
Lcd.print (Pi,4);

This will print 3.1415 on the LCD where 4 represents 4 decimal places. The LCD in this example uses 4-bit data transfer Pins D4-D7 (11-14 on the board). The potentiometer acts as a voltage divider and controls contrast of the displayed text. Its middle pin connects to Pin V0 (pin3) of the display.

**CIRCUIT DIAGRAM:**



**WORKING:**

The data pin (pin2) of the DHT11 requires an external 10K pull-up resistor. [](http://www.circuitstoday.com/wp-content/uploads/2015/09/DHT11-connection-diagram.png)The MCU (microcontroller unit) first sends a low signal of width 18mS to the DHT11. After this signal, the MCU pulls up the communication line and waits for the response from DHT11. It make take up to 2 - 40uS. Then the DHT11 pulls down the communication line and keeps it low for 80uS. Then DHT11 pulls up the line and keeps it high for 80uS. Then the DHT pulls down the line for 50uS and the next high pulse will be the first bit of the data. The data is send in bursts of 8 bits. Each high pulse of the burst indicates a data signal. The logic of the data bit is identified by measuring the width of it.

**PROGRAM:**

ORG 0000H

START: MOV DPTR,#LUT

MOV TMOD,#12H

MOV TL0,#00H

MOV SCON,50H

MAIN:  MOV R1,#08H

       SETB P3.5

       CLR P3.5

       ACALL DELAY1

       SETB P3.5

HERE: JB P3.5,HERE

HERE1: JNB P3.5,HERE1

HERE2: JB P3.5,HERE2

LOOP: JNB P3.5,LOOP

      RL A

      MOV R0,A

      SETB TR0

HERE4: JB P3.5,HERE4

       CLR TR0

       MOV A,TL0

       SUBB A,#32H

       MOV A,R0

       JB PSW.7, NEXT

SETB ACC.0

       SJMP ESC

NEXT: CLR ACC.0

ESC:  MOV TL0,#00H

      CLR PSW.7

      DJNZ R1,LOOP

      ACALL DISP

      ACALL HMDTY

      ACALL DELAY2

      LJMP MAIN

DISP: MOV A,#38H

LCALL CMD

MOV A,#0EH

LCALL CMD

MOV A,#01H

LCALL CMD

MOV A,#06H

LCALL CMD

MOV A,#84H

LCALL CMD

MOV A,#'H' ;DISPLAYS HYGROMETER

        LCALL DISPLAY

        MOV A,#'Y'

        LCALL DISPLAY

        MOV A,#'G'

        LCALL DISPLAY

        MOV A,#'R'

        LCALL DISPLAY

        MOV A,#'O'

        LCALL DISPLAY

        MOV A,#'M'

        LCALL DISPLAY

        MOV A,#'E'

        LCALL DISPLAY

        MOV A,#'T'

        LCALL DISPLAY

        MOV A,#'E'

        LCALL DISPLAY

        MOV A,#'R'

        LCALL DISPLAY

MOV A,#0C0H

       ACALL CMD

MOV A,#"H" ;DISPLAYS RH =

        ACALL DISPLAY

        MOV A,#"U"

        ACALL DISPLAY

MOV A,#"M"

        ACALL DISPLAY

MOV A,#"I"

        ACALL DISPLAY

MOV A,#"D"

        ACALL DISPLAY

MOV A,#"I"

        ACALL DISPLAY

MOV A,#"T"

        ACALL DISPLAY

MOV A,#"Y"

        ACALL DISPLAY

        MOV A,#" "

        ACALL DISPLAY

        MOV A,#"="

        ACALL DISPLAY

RET

CMD:  MOV P0,A ;TO OPERATE COMMANDS ON LCD

MOV P1,#0F4H

LCALL DELAYX

MOV P1,#0F0H

     LCALL DELAYX

     RET

DISPLAY:MOV P0,A ;TO DISPLAY TEXT

MOV P1,#0F5H

LCALL DELAYX

MOV P1,#0F1H

     LCALL DELAYX

     RET

DELAYX: MOV A,#0FFH

BACKX: DEC A

JNZ BACKX

RET

DELAY1: MOV TH1,#0B9H

        MOV TL1,#0B0H

        SETB TR1

HERE5:  JNB TF1,HERE5

        CLR TR1

        CLR TF1

MOV TL1,#00H

MOV TH1,#00H

        RET

DELAY2: MOV R1,#70H ;2 SEC DELAY

BACK: ACALL DELAY1

        DJNZ R1,BACK

        RET

HMDTY: MOV A,R0 ;TO CALC HUMIDITY

       MOV B,#0AH

       DIV AB

       MOV R2,B

       MOV B,#0AH

       DIV AB

       ACALL ASCII

RR A

       ACALL DISPLAY

       MOV A,B

       ACALL ASCII

       ACALL DISPLAY

       MOV A,R2

       ACALL ASCII

       ACALL DISPLAY

       MOV A,#"%"

       ACALL DISPLAY

       RET

ASCII: MOVC A,@A+DPTR

        RET

LUT: DB  30H ;asciii codes from 0-9

     DB  31H

      DB  32H

      DB  33H

      DB  34H

      DB  35H

      DB  36H

      DB  37H

      DB  38H

      DB  39H

      END

**OUTPUT:**

****

****