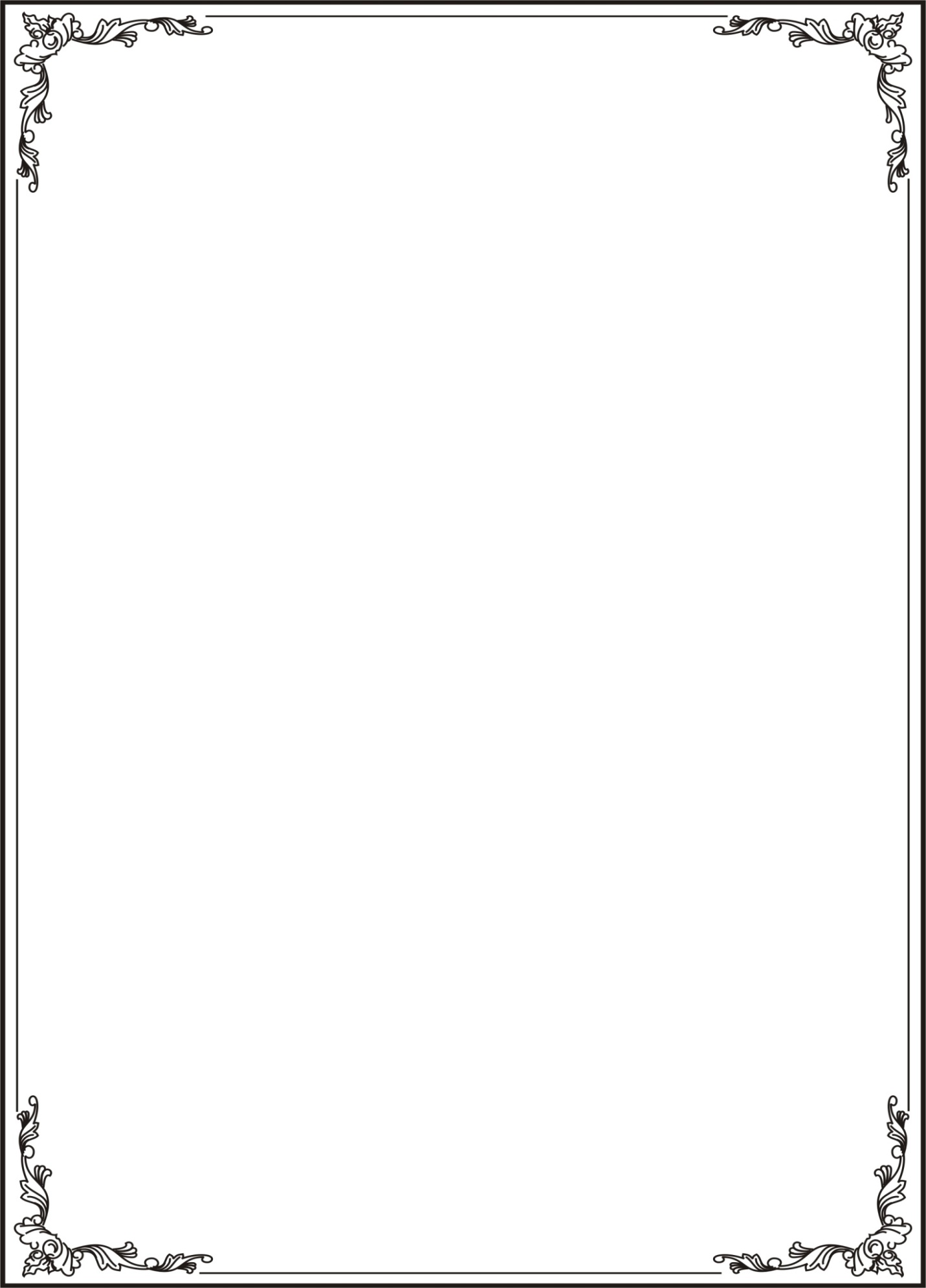
****

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔN**G

**KHOA CÔNG NGHỆ THÔNG TIN I** -----🙞🙜🕮🙞🙜-----

****

**BÁO CÁO MÔN HỌC**

**KỸ THUẬT VI XỬ LÝ**

**ĐỀ TÀI: HIỂN THỊ NHIỆT ĐỘ, ĐỘ ẨM SỬ DỤNG 8051**

**Giảng Viên: Vũ Hoài Nam**

**Nhóm: 08**

***Nhóm sinh viên thực hiện: 01***

***Nguyễn Huy Tâm - B18DCCN525***

***Phạm Văn Thanh* - *B18DCCN607***

***Lê Minh Quang - B18DCCN481***

***Nguyễn Văn Khánh - B18DCCN312***

**Hà Nội, tháng 12/2020**

# LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành cảm ơn thầy vì thầy đã tạo điều kiện thuận lợi nhất cho chúng em hoàn thành bài tập lớn của môn học Kỹ thuật vi xử lí này.

Chúng em với kiến thức hạn chế nên sẽ không thể tránh khỏi nhiều sai sót và với mong muốn được học hỏi, chúng em rất mong nhận được sự góp ý và hướng dẫn thêm của thầy để chúng em rút kinh nghiệm cho những bài tập tiếp theo được tốt hơn.

Chúng em xin chân thành cảm ơn!

*Hà Nội, ngày 03 tháng 12 năm 2020*

# 

# NHẬN XÉT VÀ ĐÁNH GIÁ

**(Của giảng viên hướng dẫn)**

|  |  |
| --- | --- |
|  | *Hà Nội, ngày…...tháng…...năm 2019*  **GIẢNG VIÊN HƯỚNG DẪN**  ***(ký, họ tên)*** |

# QÚA TRÌNH THỰC HIỆN

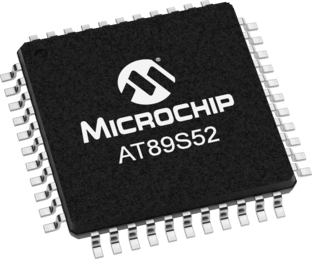
* Lựa chọn **đề tài**: Đọc giá trị cảm biến nhiệt độ, độ ẩm (DHT11) sử dụng vi điều khiển AT89S52
* Lên kế hoạch cho việc làm bài tập lớn : tìm hiểu về các linh kiện cần dùng, cảm biến nhiệt độ và độ ẩm DHT11, màn hình LCD 16x2, vi điều khiển AT89S52
* Thực hiện mua linh kiện
* Tìm hiểu các kiến thức trên internet về cách thức làm việc của vi điều khiển AT89S52, cảm biến, màn hình LCD, xây dựng mạch mô phỏng
* Cắm mạch.
* Code và nạp code cho thiết bị
* Chạy demo sản phẩm
* Trình bày báo cáo về sản phẩm trước lớp

1. **CÔNG VIỆC CỦA TỪNG THÀNH VIÊN**

* Nguyễn Huy Tâm, Phạm Văn Thanh
* Mua linh kiện cho thiết bị
* Thực hiện vẽ mô phỏng và ghép nối mạch
* Nguyễn Văn Khánh, Lê Minh Quang
* Tìm hiểu code và nạp code cho thiết bị
* Thực hiện vẽ mô phỏng và ghép nối mạch.

# GIỚI THIỆU MỘT SỐ LINH KIỆN TRONG MẠCH

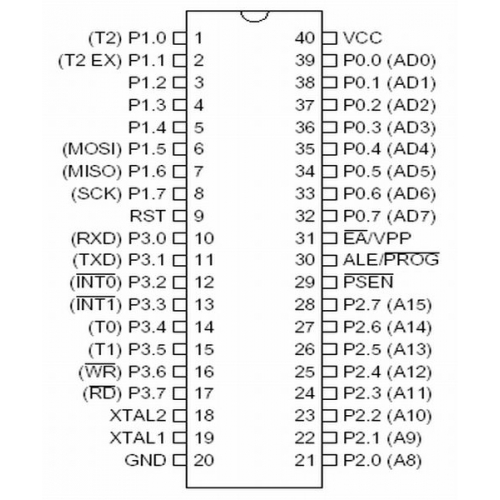
1. **Giới thiệu về vi điều khiển AT89S52**

** **

AT89S52 là họ IC vi điều khiển do hãng Atmel sản xuất. Các sản phẩm AT89S52 thích hợp cho những ứng dụng điều khiển. Việc xử lý trên byte và các toán số học ở cấu trúc dữ liệu nhỏ được thực hiện bằng nhiều chế độ truy xuất dữ liệu nhanh trên RAM nội. Tập lệnh cung cấp một bảng tiện dụng của những lệnh số học 8 bit gồm cả lệnh nhân và lệnh chia.

AT89S52 cung cấp những đặc tính chuẩn như: 8Kbyte bộ nhớ chỉ đọc có thể xóa và lập trình nhanh,128 byte, RAM32 đường I/O, 3 timer/counter 16 bit ,5 vector ngắt có cấu trúc 2 mức ngắt, 1 Port nối tiếp bán song công , 1 mạch dao động tạo xung clock và bộ dao động ON/CHIP.

**Sơ đồ chân:**



Các đặc điểm của chip AT89S52:

* 8 Byte bộ nhớ có thể lập trình, có khả năng tới 1000 chu kỳ ghi/xóa
* Tần số hoạt động từ: 0Hz đến 24 MHz
* 3 mức khóa bộ nhớ lập trình
* 3 bộ timer/couter 16 bit
* 128 byte RAM
* 4 Port xuất nhập I/O 8 bit
* 64 KB vùng nhớ mã ngoài
* 64 KB vùng dữ liệu ngoại
* 4 micro giây cho hoạt động nhân hoặc chia.

1. **Module do nhiệt độ, độ ẩm DHT11 **

***Giới thiệu:***

DHT11 là một cảm biến nhiệt độ và độ ẩm kỹ thuật số cơ bản, chi phí cực thấp. Nó sử dụng cảm biến độ ẩm điện dung và một điện trở nhiệt để đo không khí xung quanh và phát ra tín hiệu kỹ thuật số trên chân dữ liệu (không cần chân đầu vào tương tự). Nó khá đơn giản để sử dụng, nhưng yêu cầu thời gian cẩn thận để lấy dữ liệu. Nhược điểm thực sự duy nhất của cảm biến này là bạn chỉ có thể lấy dữ liệu mới từ nó cứ sau 2 giây một lần, vì vậy khi sử dụng thư viện của chúng tôi, số đọc cảm biến có thể cũ lên đến 2 giây.

**Thông số kỹ thuật:**

* Điện áp hoạt động: 3V - 5V (DC)
* Dãi độ ẩm hoạt động: 20% - 90% RH, sai số ±5%RH
* Dãi nhiệt độ hoạt động: 0°C ~ 50°C, sai số ±2°C
* Khoảng cách truyển tối đa: 20m
* Tốc độ lấy mẫu không quá 1 Hz
* 4 chân với khoảng cách 0,1

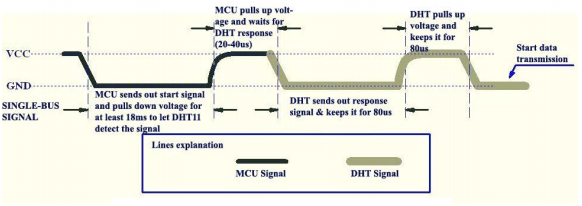
***Ứng dụng thực tế:***

* Dùng đo nhiệt độ
* Nó sử dụng cảm biến độ ẩm điện dung và một điện trở nhiệt để đo không khí xung quanh và phát ra tín hiệu kỹ thuật số trên chân dữ liệu (không cần chân đầu vào tương tự)

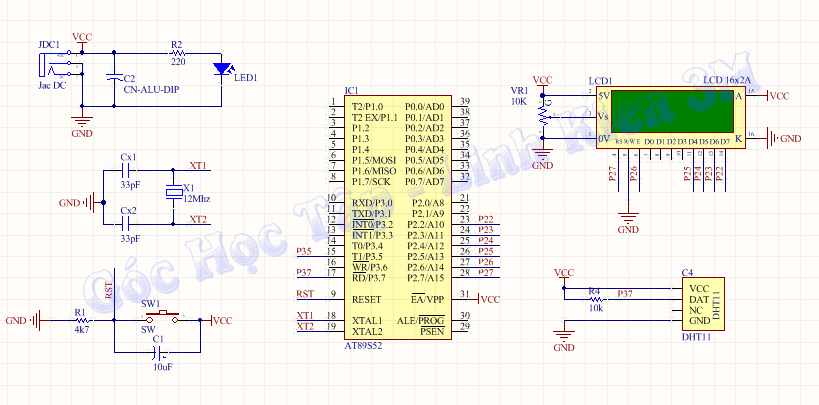
**Màn hình hiển thị LCD 16x2**

* Chi tiết các chân:
  + VSS: tương đương với GND - cực âm
  + VDD: tương đương với VCC - cực dương (5V)
  + Constrast Voltage (Vo): điều khiển độ sáng màn hình
  + Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ liệu
  + Read/Write (RW): Đọc (read mode) hay ghi (write mode) dữ liệu?
  + Enable pin: Cho phép ghi vào LCD
  + D0 - D7: 8 chân dư liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu đang ở chế độ đọc (read mode) và nó sẽ nhận giá trị HIGH hoặc LOW nếu đang ở chế độ ghi (write mode)
  + Backlight (Backlight Anode (+) và Backlight Cathode (-)): Tắt bật đèn màn hình LCD.

1. **NGUYÊN LÝ HOẠT ĐỘNG**
2. **Giao tiếp giữa DHT11 và AT89S52**



1. **Sơ đồ mạch**

****

**Các bước lập trình:**

* 1. Xác định chân của vi điều khiển giao tiếp với cảm biến DHT11
* 2. Gửi xung nhịp tới cảm biến DHT11 bằng cách thực hiện từ thấp đến cao trên các chân dữ liệu
* 3. Nhận xung phản hồi từ cảm biến DHT11
* 4. Sau khi nhận được phản hồi, nhận 40 bit liên tiếp từ cảm biển DHT11
* 5. Hiển thị dữ liệu nhận được ra màn led

**Nguyên lý hoạt động:**

- Để đo độ ẩm, họ sử dụng thành phần cảm biến độ ẩm có hai điện cực với chất giữ ẩm giữa chúng.

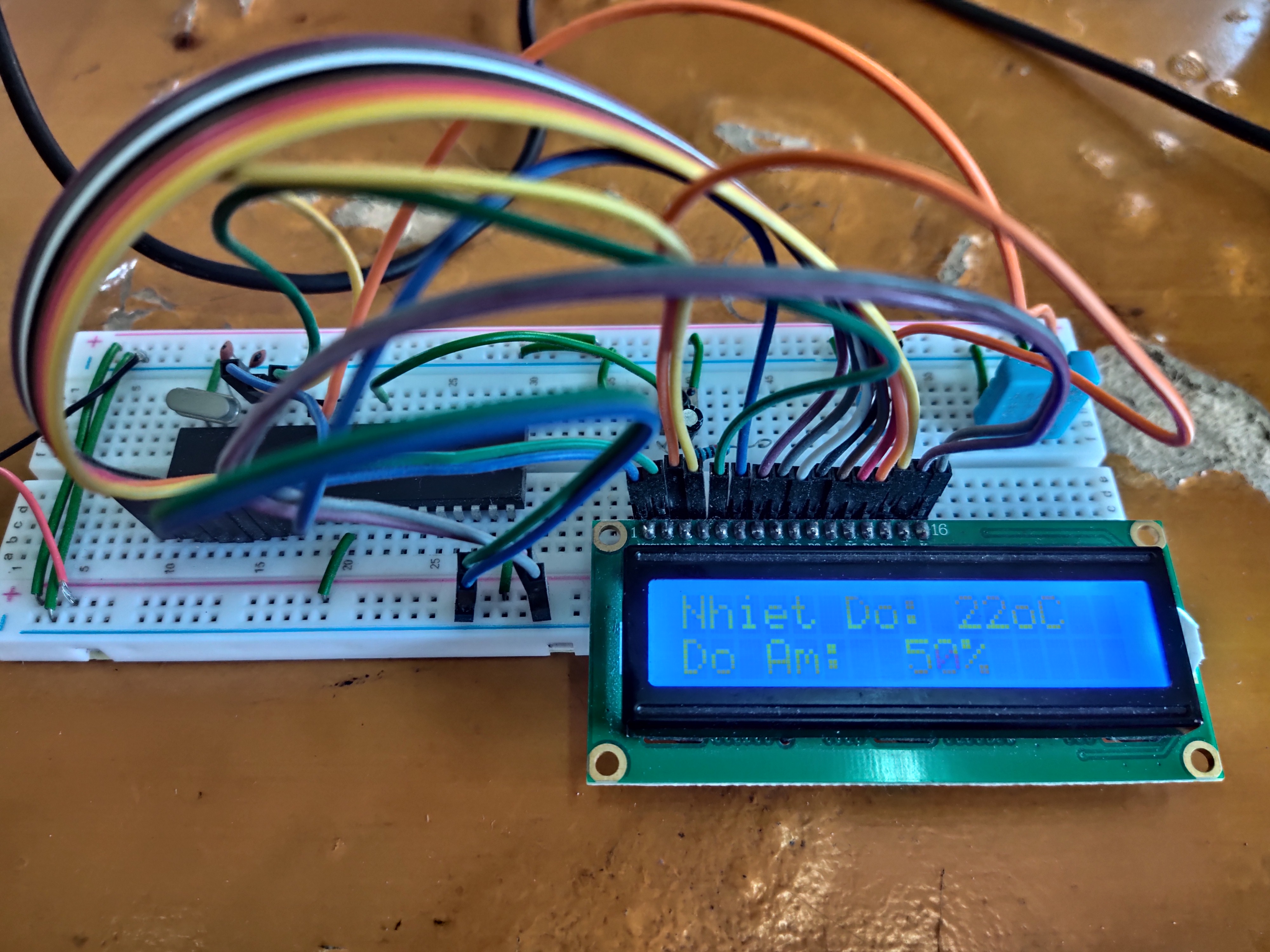
=> Vì vậy, khi độ ẩm thay đổi, độ dẫn của chất nền thay đổi hoặc điện trở giữa các điện cực này thay đổi. Sự thay đổi điện trở này được đo và xử lý bởi IC khiến cho vi điều khiển luôn sẵn sàng để đọc.

- Một nhiệt điện trở thực sự là một điện trở thay đổi điện trở của nó với sự thay đổi của nhiệt độ.

-Những cảm biến này được chế tạo bằng cách thiêu kết các vật liệu bán dẫn như gốm hoặc polyme để cung cấp những thay đổi lớn hơn trong điện trở chỉ với những thay đổi nhỏ về nhiệt độ.

-Thuật ngữ có tên là “NTC” có nghĩa là hệ số nhiệt độ âm, có nghĩa là điện trở giảm khi nhiệt độ tăng**.**

**Sản phẩm:**



Code:

DHT11 EQU P3.2 ;define chân DHT11 là P3.2

RW EQU P3.5 ;define chân RW là P3.5

RS EQU P3.6 ;define chân RS là P3.6

E EQU P3.7 ;define chân E là P3.7

ORG 000H ; bắt đầu

MOV DPTR,#LUT ; địa chỉ của LUT tới DPTR

SETB DHT11 ; Set chân data của DHT11 lên 1

MOV TMOD,#00100001B ; sets Timer1

MOV TL1,#00D ; nạp địa chỉ TL1 = 0

MAIN:

MOV R0,#40H ;nạp địa chỉ con trỏ R0 = 40H, địa chỉ RAM

MOV R1,#5D ; nạp R1 = 5

SETB DHT11 ; set chân data lên 1

CLR DHT11 ; clears chân data = 0

ACALL DELAY1 ; gọi DELAY1

SETB DHT11 ; sets set chân data lên 1

HERE: JB DHT11,HERE ; chờ cho đến khi chân data = 0

HERE1:JNB DHT11,HERE1 ; chờ cho đến khi chân data = 1

HERE2:JB DHT11,HERE2 ; chờ cho đến khi chân data = 0

; bat đầu nhận dữ liệu

LOOP1:MOV R3,#8D ; nạp R3 = 8

LOOP: JNB DHT11,LOOP ; chờ cho đến khi chân data = 1

RL A ; dịch trái A

MOV @R0,A ; sao chép nội dung của A to địa chỉ R0 trỏ tới

SETB TR1 ; start timer 1

HERE4:JB DHT11,HERE4 ; chờ cho đến khi chân data = 0

CLR TR1 ; stop timer 1

MOV A,TL1 ; TL1 chứa độ rộng của xung

SUBB A,#50D ; nếu carry flag là 1 thì độ rộng xung nhỏ hơn 50us

;nếu carry flag là 0 thì độ rộng xung lớn hơn 50us

MOV A,@R0 ; Sao chép nội dung R0 tới A

JB CY, NEXT ; nhảy tới next nếu CY = 1, bit = 0

SETB ACC.0 ; set bit ACC.0 = 1, bit = 1

SJMP ESC ; nhảy tới nhãn ESC

NEXT:CLR ACC.0 ; clear bít ACC.0 = 0, bit = 0

ESC: MOV TL1,#00D ; nạp TL1 = 0

CLR CY ; clear cờ carry flag

DJNZ R3,LOOP ; nhảy đến nhãn LOOP nếu R3 khác 0

INC R0 ; tăng con trỏ R0

DJNZ R1,LOOP1 ; nhảy đến nhãn LOOP1 nếu R1 khác 0

ACALL LCD\_INIT ; gọi chương trình con LCD\_INIT

ACALL TEXT1 ; gọi TEXT1

ACALL TEMP ; gọi TEMP

ACALL LINE2 ; Gọi LINE2

ACALL TEXT2 ; gọi TEXT2

ACALL HUM ; gọi HUM

ACALL DELAY2 ; gọi chương trình con DELAY2

LJMP MAIN ; nhảy tới nhãn MAIN

DELAY1: MOV TH0,#0B9H ; Nạp TH0 = B9

MOV TL0,#0B0H ; Nạp TL0 = B0

SETB TR0 ; start timer0

HERE5: JNB TF0,HERE5 ; chờ cho đến khi cờ tràn TF0 = 1

CLR TR0 ; stop timer0

CLR TF0 ; clear timer flag 0

RET

DELAY2:MOV R1,#112D ; Nạp R1 = 112

BACK:ACALL DELAY1 ; gọi DELAY1

DJNZ R1,BACK ; lặp 112 lần cho đủ 2s

RET

TEMP:MOV R1,#42H ; nạp con trỏ R1 = 42H, địa chỉ RAM lưu 1 byte phần nguyên của nhiệt độ

MOV A, @R1 ; sao chép nội dung của R1 tới A

MOV B,#10D ; nạp B = 10

DIV AB ; A chia cho B, được A phần nguyên, B phần dư

MOV R5,B ; lưu phần dư B vào R5

MOV B,#10D ; nạp B =10

DIV AB ; A chia cho B, được A phần nguyên, B phần dư

MOV A,B ; sao chép B tới A

ACALL ASCII ; gọi chương trình con ASCII

ACALL DISPLAY ; gọi chương trình con DISPLAY

MOV A,R5 ; sao chép R5 tới A

ACALL ASCII ; gọi chương trình con ASCII

ACALL DISPLAY ; gọi chương trình con DISPLAY

MOV A,#"o" ; nạp kí tự "o" to A

ACALL DISPLAY ; gọi chương trình con DISPLAY

MOV A,#"C" ; nạp kí tự "C" to A

ACALL DISPLAY ; gọi chương trình con DISPLAY

RET

HUM: MOV R0, #40H ; nạp con trỏ R0 = 40H, địa chỉ RAM lưu 1 byte phần nguyên của độ ẩm

MOV A,@R0 ; sao chép nội dung của R0 tới A

MOV B,#10D ; nạp B = 10

DIV AB ; A chia cho B, được A phần nguyên, B phần dư

MOV R2,B ; lưu phần dư B vào R5

MOV B,#10D ; nạp B =10

DIV AB ; A chia cho B, được A phần nguyên, B phần dư

MOV A,B ; sao chép B tới A

ACALL ASCII ; gọi chương trình con ASCII

ACALL DISPLAY ; gọi chương trình con DISPLAY

MOV A,R2 ; sao chép R2 tới A

ACALL ASCII ; gọi chương trình con ASCII

ACALL DISPLAY ; gọi chương trình con DISPLAY

MOV A,#"%" ; nạp kí tự "%" tới A

ACALL DISPLAY ; gọi chương trình con DISPLAY

RET

TEXT1: MOV A,#"N" ; Hiển thị text "Nhiet Do: "

ACALL DISPLAY

MOV A,#"h"

ACALL DISPLAY

MOV A,#"i"

ACALL DISPLAY

MOV A,#"e"

ACALL DISPLAY

MOV A,#"t"

ACALL DISPLAY

MOV A,#" "

ACALL DISPLAY

MOV A,#"D"

ACALL DISPLAY

MOV A,#"o"

ACALL DISPLAY

MOV A,#":"

ACALL DISPLAY

MOV A,#" "

ACALL DISPLAY

RET

TEXT2: MOV A,#"D" ; Hiển thị text "Do Am: "

ACALL DISPLAY

MOV A,#"o"

ACALL DISPLAY

MOV A,#" "

ACALL DISPLAY

MOV A,#"A"

ACALL DISPLAY

MOV A,#"m"

ACALL DISPLAY

MOV A,#":"

ACALL DISPLAY

MOV A,#" "

ACALL DISPLAY

MOV A,#" "

ACALL DISPLAY

RET

LCD\_INIT:MOV A,#0CH ;hiển thị điểm nháy ON OFF

ACALL LCD\_CMD

MOV A,#01H ;xóa màn hình

ACALL LCD\_CMD

MOV A,#06H ;tăng điểm nháy

ACALL LCD\_CMD

MOV A,#80H ;điểm nháy ở vị trí đầu tiên của line 1

ACALL LCD\_CMD

MOV A,#3CH ;kích hoạt line 2

ACALL LCD\_CMD

RET

LCD\_CMD: MOV P2,A ; di chuyển A tới P2

CLR RS ; clear chân Rs của LCD

CLR RW ; clears chân Rw cảu LCD

SETB E ; set chân E của LCD

CLR E ; clear chân E của LCD

ACALL DELAY ; gọi chương trình con DELAY

RET

LINE2:MOV A,#0C0H ; nhảy tới dòng 2 vị trí 0

ACALL LCD\_CMD

RET

DISPLAY:MOV P2,A ; di chuyển A tới P2

SETB RS ; set chân RS của LCD

CLR RW ; clear chân RW của LCD

SETB E ; set chân E của LCD

CLR E ; clear chân E của LCD

ACALL DELAY ; gọi chương trình con DELAY

RET

DELAY: CLR E ;clear chân E của LCD

CLR RS ;clear chân RS của LCD

SETB RW ;set chân RW của LCD

MOV P2,#0FFH ;set chân P2 input

SETB E ;set chân E của LCD

MOV A,P2 ;di chuyển P2 tới A

JB ACC.7,DELAY ;nhảy tới nhãn DELAY nếu ACC.7 =1

CLR E ;clear chân E của LCD

CLR RW ;clear chân RW của LCD

RET

ASCII: MOVC A,@A+DPTR ;viết mã ascii code của data tới A

RET

LUT: DB 48D ;ascii of "0"

DB 49D ;ascii of "1"

DB 50D ;ascii of "2"

DB 51D ;ascii of "3"

DB 52D ;ascii of "4"

DB 53D ;ascii of "5"

DB 54D ;ascii of "6"

DB 55D ;ascii of "7"

DB 56D ;ascii of "8"

DB 57D ;ascii of "9"

END