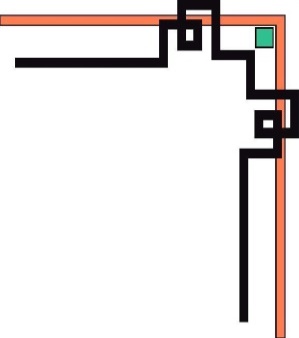
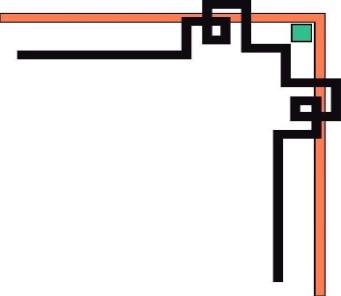
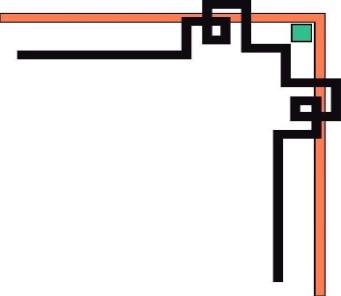
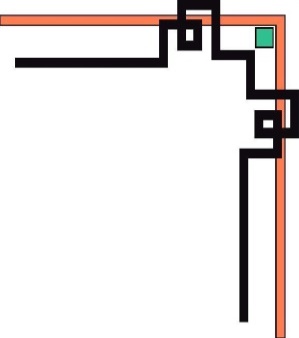
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC: VI XỬ LÝ**

**ĐỀ TÀI: MẠCH KHẢO SÁT NHIỆT ĐỘ, ĐỘ ẨM**

**SỬ DỤNG CẢM BIẾN DHT11 VÀ VI ĐIỀU KHIỂN PIC16F877A**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | : | ThS. Nguyễn Đình Thuận |
| Sinh viên thực hiện | : | Nguyễn Tài Tuấn – MSSV: 20154112  Ngô Thanh Tùng – MSSV: 20154246 |
| Lớp | : | KSTN-Cơ điện tử-K60 |

Hà Nội, tháng 11/2018

MỤC LỤC

[ĐỀ BÀI: Khảo sát nhiệt độ ẩm sử dụng cảm biến DHT11 và vi điều khiển PIC16F877A 3](#_Toc8230991)

[BÀI LÀM 3](#_Toc8230992)

[1. Thiết kế. 3](#_Toc8230993)

[1.1. Cảm biến nhiệt độ, độ ẩm DHT11 3](#_Toc8230994)

[1.2. Vi điều khiển PIC16F877A 7](#_Toc8230995)

[2. Lập trình cho vi điều khiển PIC16F877A. 9](#_Toc8230996)

[3. Mô phỏng bằng phần mềm Proteus 12](#_Toc8230997)

[4. Sơ đồ nguyên lý và sơ đồ mạch in 14](#_Toc8230998)

[5. Lắp đặt 15](#_Toc8230999)

[Tài liệu tham khảo. 15](#_Toc8231000)

# ĐỀ BÀI: Khảo sát nhiệt độ ẩm sử dụng cảm biến DHT11 và vi điều khiển PIC16F877A

# BÀI LÀM

## Thiết kế.

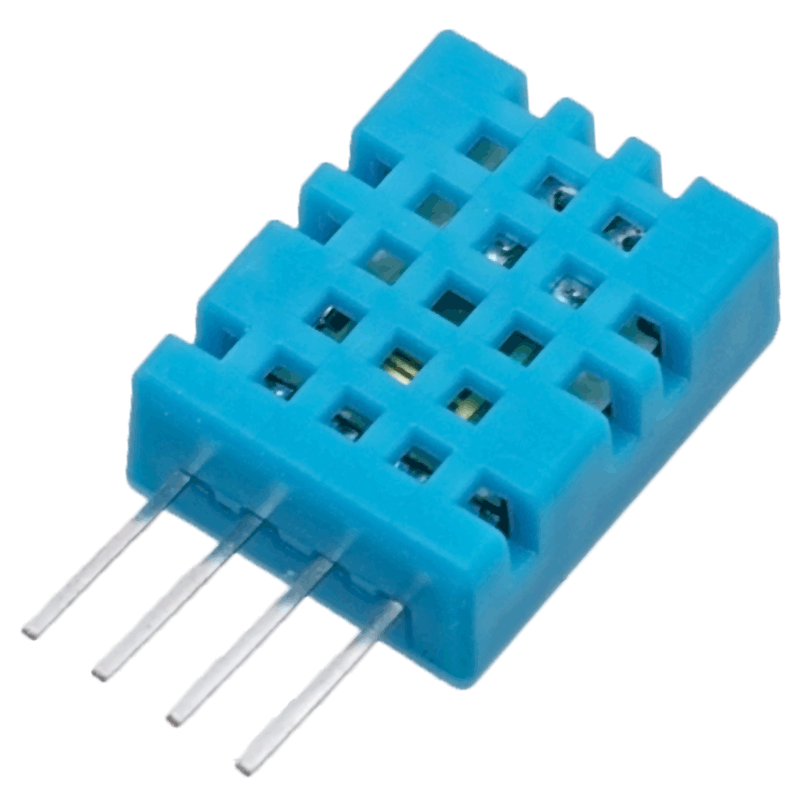
### Cảm biến nhiệt độ, độ ẩm DHT11

a) Giới thiệu

DHT11 Là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1-wire (giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu.

Thông số kỹ thuật

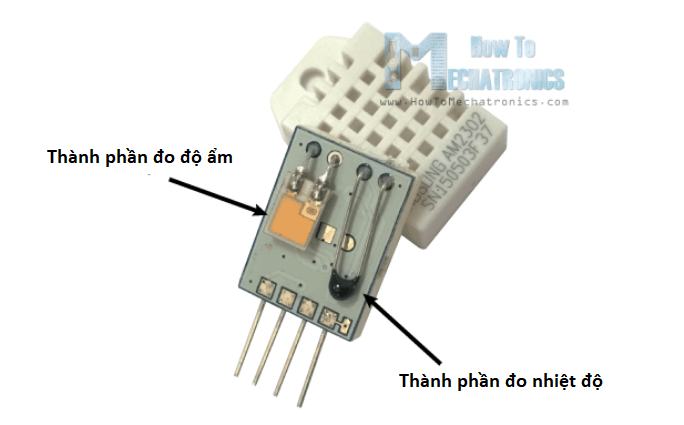
* Điện áp hoạt động: 3V - 5V (DC)
* Dãi độ ẩm hoạt động: 20% - 90% RH, sai số ±5%RH
* Dãi nhiệt độ hoạt động: 0°C ~ 50°C, sai số ±2°C
* Khoảng cách truyển tối đa: 20m



Hình 1. Cảm biến DHT11

b) Cấu tạo của DHT11

DHT11 gồm thành phần đo độ ẩm, thành phần đo nhiệt độ (mặt trước) và IC có chức năng biến đổi tín hiệu tương tự thành tín hiệu số (mặt sau).

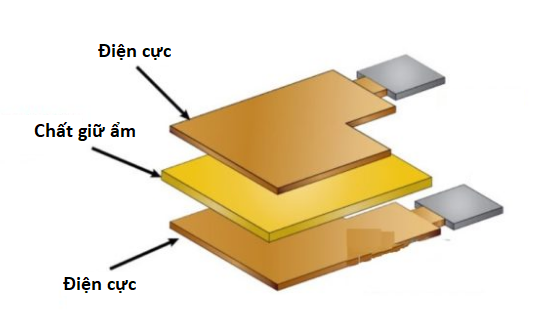


Hình 2. Cấu tạo của DHT11

* Thành phần đo độ ẩm:

- Gồm 2 điện cực với chất giữ ẩm ở giữa.

- Khi độ ẩm thay đổi, độ dẫn điện của chất giữ ẩm thay đổi, khiến điện trở thay đổi.



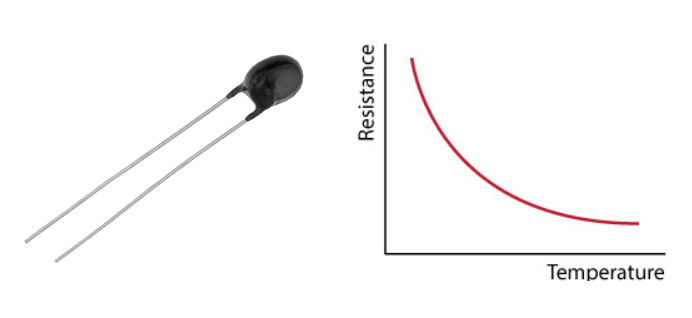
Hình 3. Thành phần đo độ ẩm

* Thành phần đo nhiệt độ:

- Gồm 1 cảm biến nhiệt độ NTC có khả năng thay đổi điện trở tùy theo nhiệt độ.

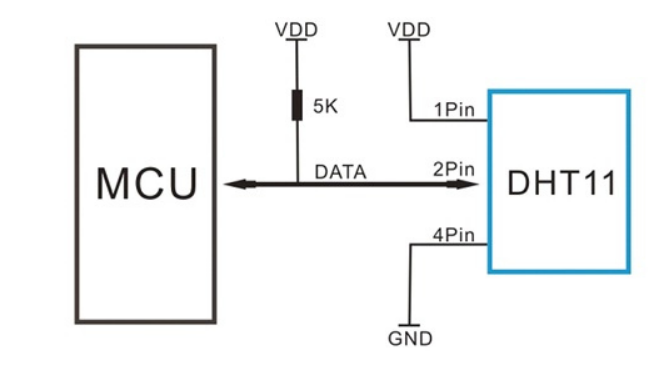
- Có khả năng tạo ra sự thay đổi lớn trong điện trở chỉ với sự thay đổi nhỏ trong nhiệt độ nhờ chất bán dẫn (gốm hoặc polymers) được thiêu kết.

- Điện trở thay đổi tỷ lệ nghịch với nhiệt độ (NTC nghĩa là “Negative Temperature Coefficient”).



Hình 4. Cảm biến nhiệt độ NTC

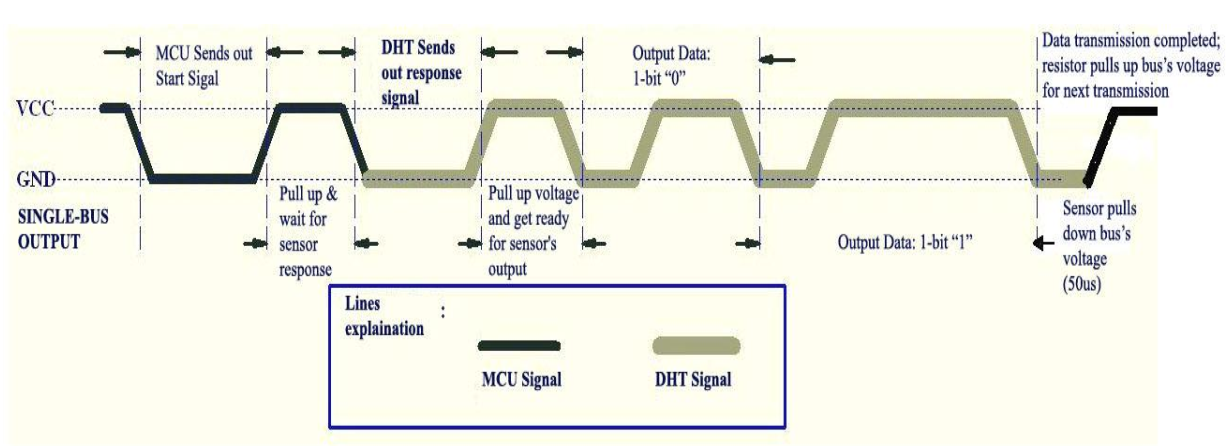
c) Giao tiếp giữa DHT11 và MCU



Hình 5. Sơ đồ kết nối MCU và DHT11

Để có thể giao tiếp với DHT11 theo chuẩn 1 dây, vi xử lý thực hiện theo 2 bước:

* Gửi tin hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.
* Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.



Hình 6. Quá trình giao tiếp của MCU và DHT11

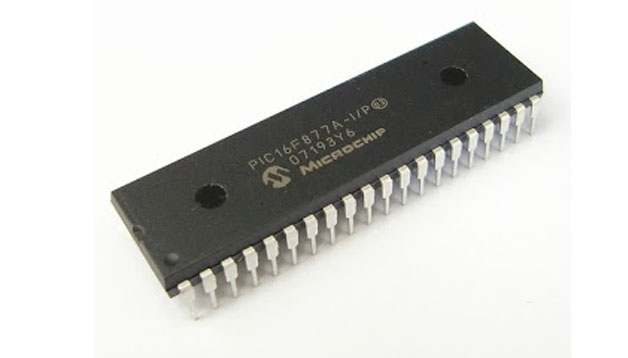
### Vi điều khiển PIC16F877A

a) Giới thiệu

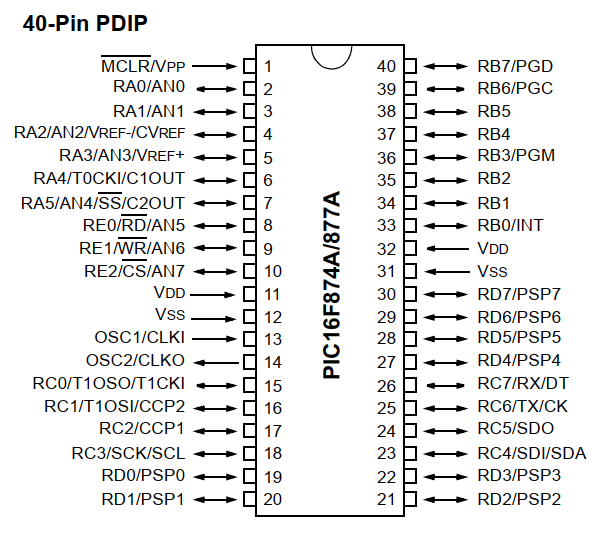
Đây là một dòng vi điều khiển khá phổ biến, đầy đủ các chức năng, phù hợp với các ứng dụng cơ bản.

Một số đặc điểm của PIC16F877A:

* PIC16F877A là loại vi điều khiển 8bit tầm trung của hãng microchip.
* PIC16F877A có kiến trúc Havard, sử dụng tập lệnh kiểu RISC (Reduced Instruction Set Computer) với chỉ 35 lệnh cơ bản.
* Tất cả các lệnh được thực hiện trong một chu kì lệnh ngoại trừ các lệnh rẽ nhánh.



Hình 7. Vi điều khiển PIC16F877A



Hình 8. Sơ đồ chân vi điều khiển PIC16F877A loại 40 chân

b) Giao tiếp UART PIC16F877A (chế độ truyền)

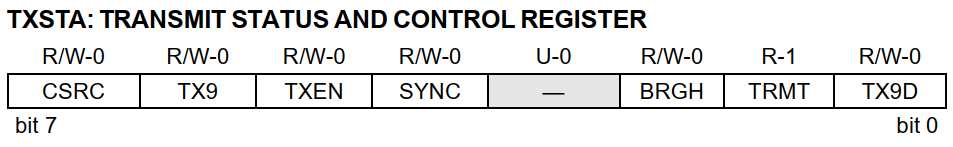
Quá trình gửi dữ liệu từ vi điều khiển:

* Khởi tạo baund rate: ở thanh ghi SPBRG

Cho phép quá trình truyền thông không đồng bộ bằng cách thiết lập SPEN = 1; SYNC = 0;

* Cho phép truyền dữ liệu bằng cách thiết lập bit TXEN = 1;
* Khi cần truyền dữ liệu thì cần set dữ liệu đó lên TXREG.

Thanh ghi quy định chế độ truyền:



Trong đó:

- CSRC: Không sử dụng trong giao tiếp UART

- TX9: Cho phép truyền nhận chế độ 9 bit

1 = Hoạt động với chế độ 9 bit.

0 = Hoạt động với chế độ 8 bit.

- TXEN: Cho phép truyền UART

1 = Cho phép

0 = Không cho phép

- SYNC: Cho phép chế độ đồng bộ

1 = Truyền chế độ đồng bộ.

0 = Truyền chế độ bất đồng bộ.

- BRGH: Chọn chế độ baund rate

1 = Tốc độ cao (bất đồng bộ).

0 = Tốc độ thấp (bất đồng bộ).

- TRMT: Trạng thái thanh ghi truyền.

1 = Thanh ghi TSR trống

0 = Thanh ghi TSR có dữ liệu.

- TX9D: Dữ liệu bit thứ 9 trong chế độ truyền 9 bit.

## Lập trình cho vi điều khiển PIC16F877A.

Chương trình chính “main.c”

#include <stdio.h>

#include <stdlib.h>

#include "main.h"

#include "uart.h"

#include "dht11.h"

#define UART\_TX RC6

#define UART\_RX RC7

int uart1[32];

int str[20];

unsigned char humid,temp;

void main(){

unsigned char str[20];

TRISC6 = 0; //Set output

TRISC7 = 1; //Set input

//Transmit status and control register

TX9 = 0;

TXEN = 1;

SYNC = 0;

BRGH = 1;

SPBRG = 129;

while(1)

{

if(DHT11\_init())

{

humid = check\_humid();

temp = check\_temp();

sprintf(str,"%u", temp);

UART\_Write\_Text(str);

\_\_delay\_ms(500);

sprintf(str,"%u", humid);

UART\_Write\_Text(str);

\_\_delay\_ms(500);

}

}

}

Thư viện điều khiển DHT11 “dht11.h”

#include <stdlib.h>

#include <stdint.h>

#define DHT11\_DATA RB2

#define DHT11\_TRIS TRISB2

unsigned char H\_humid;

unsigned char H\_temp;

unsigned char check\_humid(void)

{

return H\_humid;

}

unsigned char check\_temp(void)

{

return H\_temp;

}

int dht11\_data[40];

uint8\_t DHT11\_init(void)

{

DHT11\_TRIS = 0; //Set output

DHT11\_DATA = 0; //send from MCU 0

\_\_delay\_ms(20);

DHT11\_DATA = 1; //signal from MCU goes to 1

\_\_delay\_us(30);

DHT11\_TRIS = 1; //Set input

\_\_delay\_us(60);

if(DHT11\_DATA) return 0; // Test if signal is low, if high->error

else while(!DHT11\_DATA); // Wait for the signal to be high

\_\_delay\_us(60);

if(!DHT11\_DATA) return 0; // Test if signal is high, if low->error

else while(DHT11\_DATA); // Wait for the signal to be low

unsigned char a[5]; // receive data

char j = 0;

char k = 0;

while(j < 5)

{

a[j] = 0;

j++;

}

j = 0;

for(j = 0; j < 5; j++)

{

for(k = 0; k < 8; k++)

{

while(!DHT11\_DATA); //wait data to go high

\_\_delay\_us(50);

if(DHT11\_DATA)

{

a[j] |= (1<<(7 - k));

while(DHT11\_DATA); //wait data to go low

}

}

}

int L\_humid,L\_temp;

int sum;

sum = a[0] + a[1] + a[2] + a[3];

if(sum != a[4]) return 0; //Error

H\_humid = a[0];

L\_humid = a[1];

H\_temp = a[2];

L\_temp = a[3];

return 1;

}

Thư viện điều khiển UART “uart.h”

#define UART\_TX RC6

#define UART\_RX RC7

#include <string.h>

void UART\_Write(char data)

{

while(!TRMT);

TXREG = data;

}

void UART\_Write\_Text(char \*text)

{

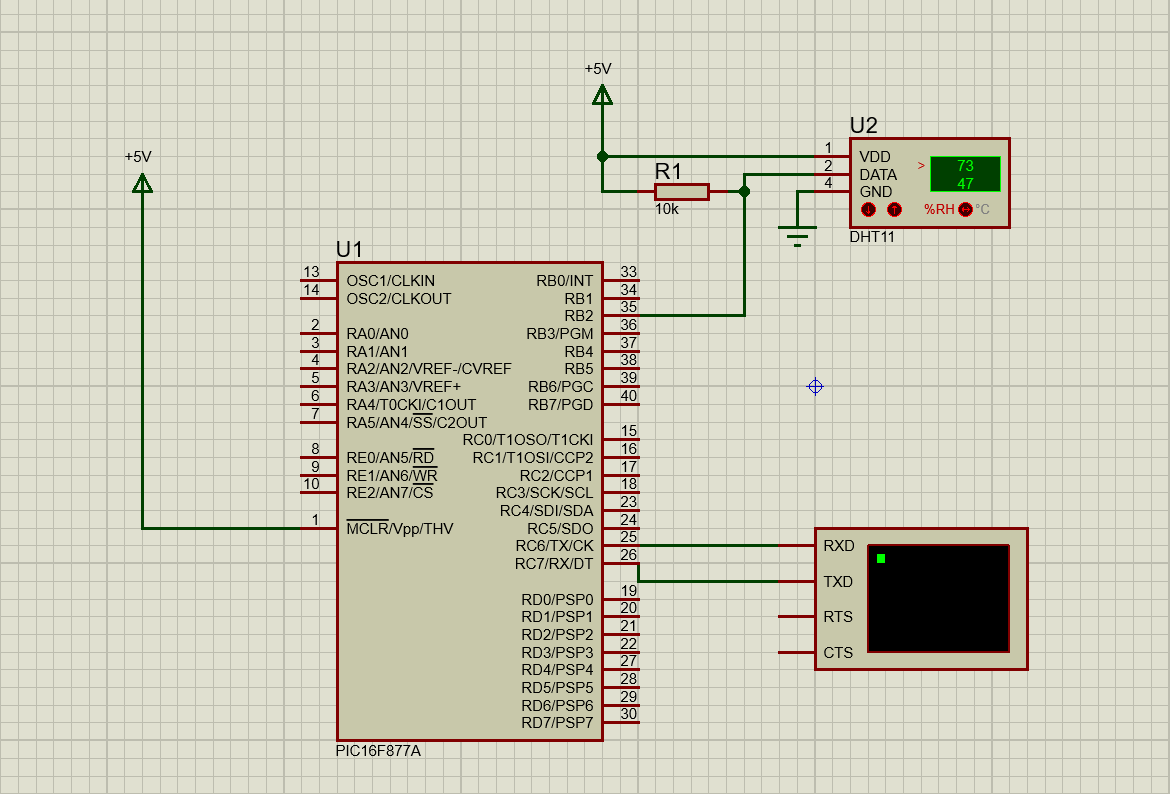
int i;

for(i=0;text[i]!='\0';i++)

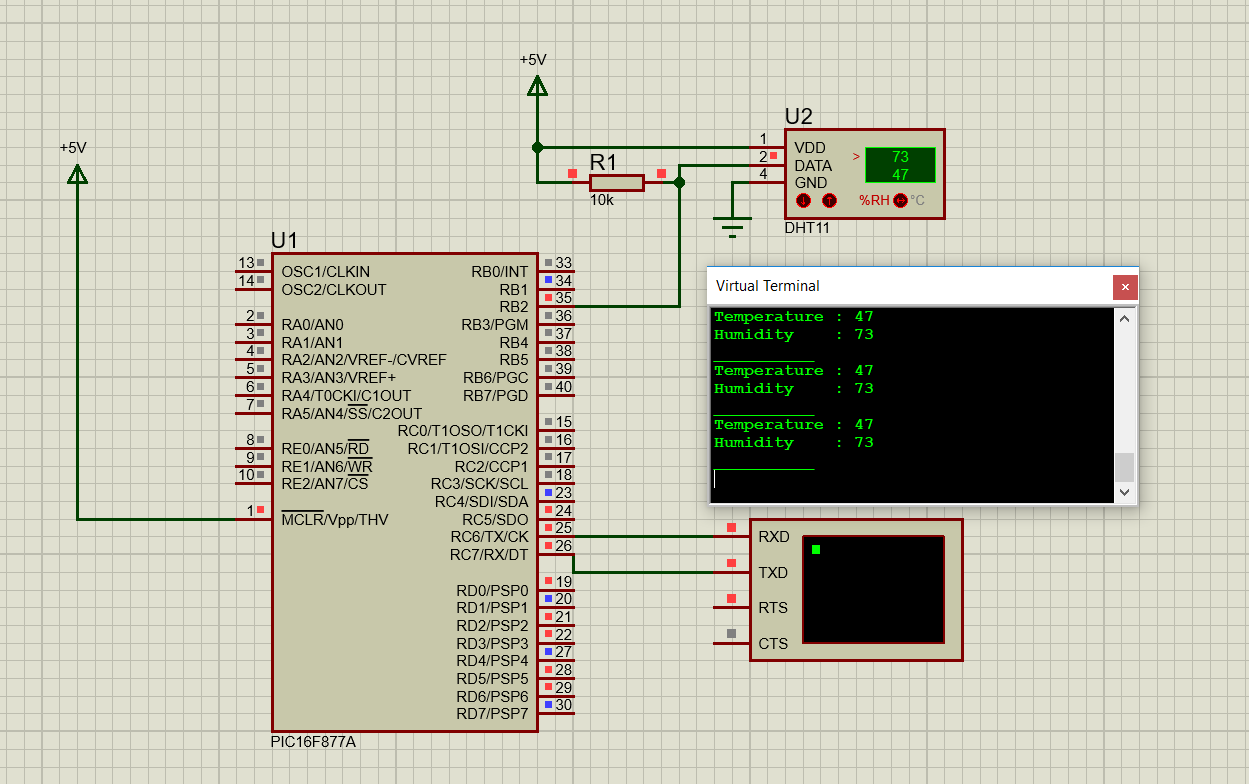
UART\_Write(text[i]);

}

## Mô phỏng bằng phần mềm Proteus

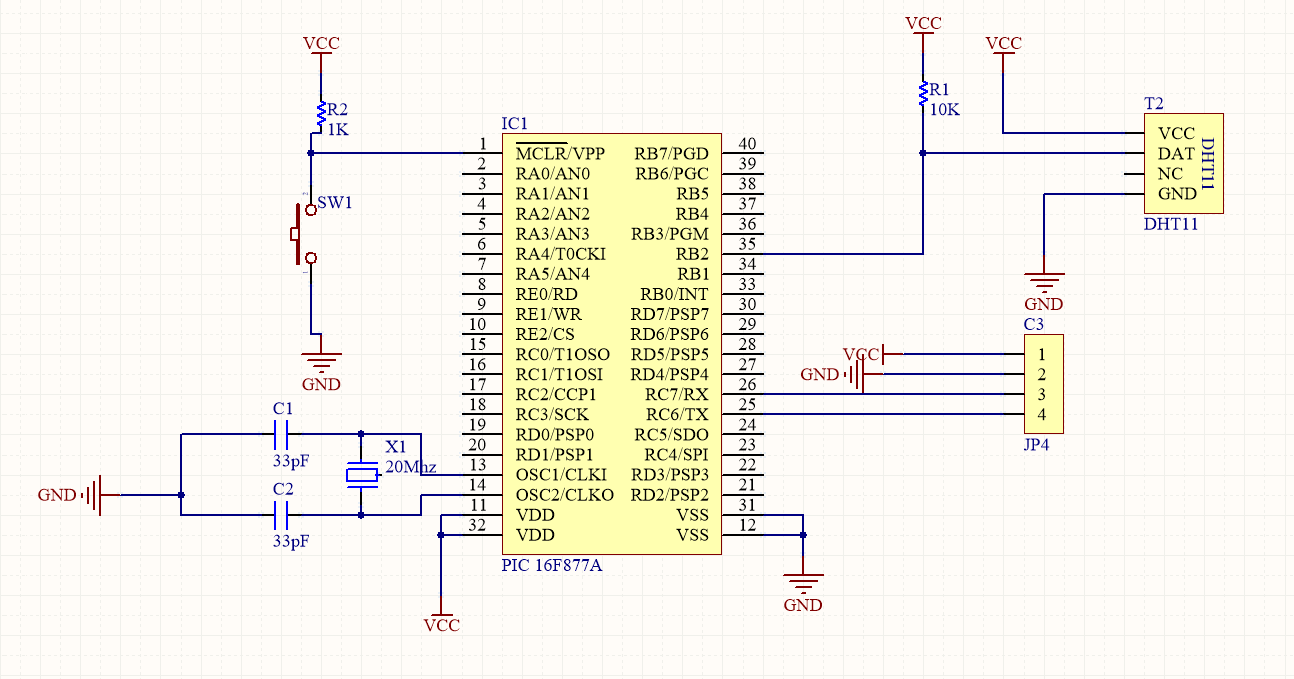


Hình 9. Mạch mô phỏng

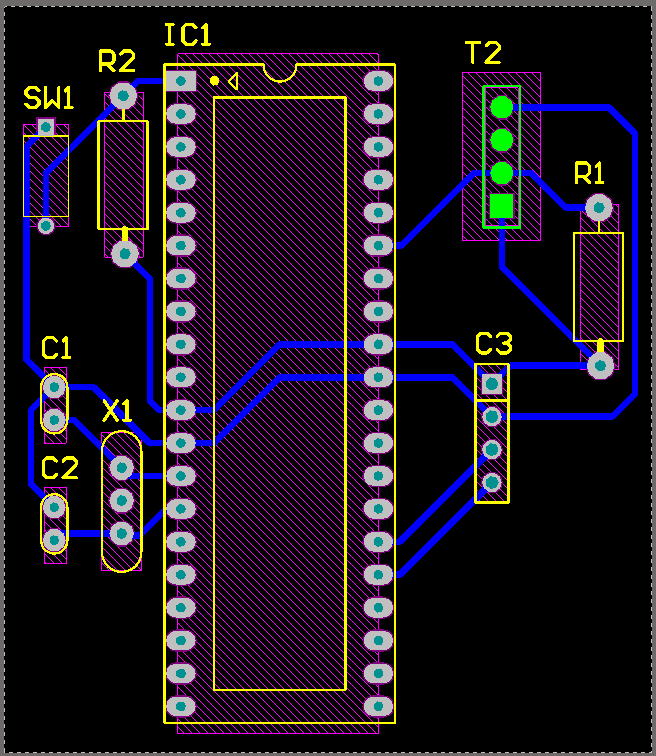


Hình 10. Mạch mô phỏng khi hoạt động

## Sơ đồ nguyên lý và sơ đồ mạch in



Hình 11. Sơ đồ nguyên lý



Hình 12. Sơ đồ mạch in

## Lắp đặt

## Tài liệu tham khảo.

1. [DHT11 Humidity & Temperature Sensor, Mouser Electronics.](https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf)
2. [PIC16F877A Data Sheet, Microchip.](https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf)
3. [LCD HD44780 - Giao Tiếp Và Lập Trình Điều Khiển, Trietnguyen, SPKT, 30/6/2007.](https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf)