

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**TT Kiến Trúc Và Tổ Chức Máy Tính**

**TÌM HIỂU KHỐI MODULE ULN2003 VỚI MOTOR BƯỚC**

**GVHD: Huỳnh Hoàng Hà**

**Mã học phần: COOL325364\_23\_1\_11**

**Ngày, tiết học: Thứ sáu tiết 1-6**

**SVTH: Đặng Minh Kha**

**MSSV: 21139025**

**TP. Thủ Đức - 08/2023**

**IC ULN2003 và DC Motor Bước**

1. **IC ULN2003 là gì?**

ULN2003 là một IC 16 chân, bao gồm 7 cặp darlington (mỗi cặp được bảo vệ bằng supression diode) và do đó có khả năng xử lý tối đa 7 tải (có thể cảm ứng).

Nói một cách dễ hiểu, chúng ta có 7 driver trong một chip ULN2003 và do đó có thể kiểm soát tối đa 7 tải.

Mỗi cặp Darlington có thể xử lý tải tối đa 500mA, trong khi giá trị đỉnh là 600mA.

Tương tự, điện áp đầu ra tối đa của mỗi cặp darlington là 50V.

Trong hình bên dưới, bạn có thể thấy ULN2003 có 16 chân, các đầu vào và đầu ra tương ứng đối diện nhau (để dễ thiết kế mạch).

Ngoài các chân I / O, còn có chân ground cần cấp 0V và chân Vcc (Chung).

1. **Sơ đồ chân**

|  |  |  |
| --- | --- | --- |
| **Số chân** | **Tên chân** | **Mô tả** |
| 1 đến 7 | Input 1 đến input 7 | Bảy chân đầu vào của cặp Darlington, mỗi chân được kết nối với cực gốc của transistor và có thể được kích hoạt bằng cách sử dụng + 5V |
| 8 | Ground | Điện áp tham chiếu đất 0V |
| 9 | COM | Được sử dụng làm chân kiểm tra hoặc chân triệt điện áp (tùy chọn để sử dụng) |
| 10 đến 16 | Output 1 đến ouput 7 | Đầu ra tương ứng của bảy chân đầu vào. Mỗi chân đầu ra sẽ chỉ được nối đất khi chân đầu vào tương ứng của nó ở mức cao (+ 5V) |

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự độngẢnh có chứa văn bản, biểu đồ, số, hàng

Mô tả được tạo tự động**

**Hình 1: Cấu trúc bên ngoài và bên trong của ULN2003**

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, biểu đồ

Mô tả được tạo tự động**

**Hình 2: Schematic của ULN2003 trên KIT học tập 8051**

1. **Thông số kĩ thuật**

IC ULN2003 là một trong những IC điều khiển động cơ được sử dụng phổ biến nhất. IC này rất hữu ích khi chúng ta cần điều khiển tải dòng điện cao bằng cách sử dụng các mạch logic kỹ thuật số như Op-amp, Timer, Gate, Arduino, PIC, ARM, ... Ví dụ, một động cơ yêu cầu 9V và 300mA để chạy không thể được cấp nguồn bởi arduino I / O do đó chúng ta sử dụng IC này để cung cấp đủ dòng điện và điện áp cho tải. IC này thường được sử dụng để điều khiển module relay, động cơ, LED dòng cao và thậm chí cả động cơ bước. Vì vậy, nếu bạn có bất cứ thứ gì lớn hơn 5V 80mA để hoạt động, thì IC này sẽ là sự lựa chọn phù hợp.

1. **Cơ chế và ứng dụng của ULN2003**

IC ULN2003 là một trong những IC điều khiển động cơ được sử dụng phổ biến nhất. IC này rất hữu ích khi chúng ta cần điều khiển tải dòng điện cao bằng cách sử dụng các mạch logic kỹ thuật số như Op-amp, Timer, Gate, Arduino, PIC, ARM, ... Ví dụ, một động cơ yêu cầu 9V và 300mA để chạy không thể được cấp nguồn bởi arduino I / O do đó chúng ta sử dụng IC này để cung cấp đủ dòng điện và điện áp cho tải. IC này thường được sử dụng để điều khiển module relay, động cơ, LED dòng cao và thậm chí cả động cơ bước. Vì vậy, nếu bạn có bất cứ thứ gì lớn hơn 5V 80mA để hoạt động, thì IC này sẽ là sự lựa chọn phù hợp.

ULN2003 là một IC 16 chân. Nó có bảy cặp Darlington bên trong, mỗi cặp có thể điều khiển tải lên đến 50V và 500mA. Đối với bảy Cặp Darlington này, chúng ta có bảy chân đầu vào và đầu ra. Thêm vào đó chúng ta có thể nối đất và chân chung. Chân ground được nối đất và việc sử dụng chân chung là tùy chọn. Lưu ý rằng vi mạch này không có bất kỳ chân Vcc (nguồn) nào; điều này là do nguồn điện cần thiết để các transistor hoạt động sẽ được lấy từ chính chân đầu vào. Mạch dưới đây là một mạch đơn giản có thể được sử dụng để kiểm tra hoạt động của IC ULN2003.

Ảnh có chứa biểu đồ, hàng, văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

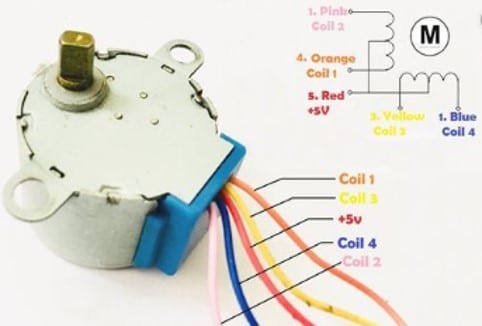
**Hình 3: Ví dụ về cơ chế hoạt động của ULN2003**

Trong mạch coi LED là các tải và các chân logic (màu xanh) là các chân kết nối với mạch kỹ thuật số hoặc vi điều khiển như Arduino. Chú ý rằng chân dương của LED được kết nối với điện áp tải dương và chân âm được kết nối với chân đầu ra của IC. Điều này là do khi chân đầu vào của IC lên cao, chân đầu ra tương ứng sẽ được nối đất. Vì vậy, khi cực âm của LED được nối đất, nó sẽ hoàn thành mạch và do đó phát sáng. Các tải kết nối với chân đầu ra tối đa là 50C và 500mA mỗi loại. Tuy nhiên, bạn có thể chạy tải dòng cao hơn bằng cách kết hợp hai hoặc nhiều chân đầu ra. Ví dụ: nếu bạn kết hợp ba chân, bạn có thể điều khiển lên đến (3 \* 500mA) ~ 1.5A.

Chân COM được nối đất thông qua một công tắc, kết nối này là tùy chọn. Nó có thể được sử dụng một công tắc thử nghiệm, nghĩa là khi chân này được nối đất thì tất cả các chân đầu ra sẽ được nối đất.

1. **Động cơ bước**

Động cơ bước 28BYJ-48, sơ đồ động cơ có 5 dây. Trong đó, dây màu đỏ là dây cấp điện và được nối với nguồn 5V. Bốn dây còn lại là 4 dây được nối tương ứng vào 4 cuộn dây bên trong động cơ. Hình ảnh bên dưới mô tả sơ đồ chân thực tế của động cơ bước 5V 28BYJ-48



**Hình 4: Động cơ bước 5 dây**

**Thông số kỹ thuật của động cơ bước S8BYJ-48:**

* Là cuộn dây 5 chân đơn cực với điện áp một chiều định mức 5V
* Có 4 pha có góc bước 5.625°/64.
* Tần số của động cơ bước là 100Hz và khả năng cách điện 600VAC/1mA/1s.
* Khuyến khích điều khiển bằng phương pháp half-step.

1. **Ứng dụng dùng IC ULN2003A làm mạch đệm điều khiển động cơ bước DC thông qua IC điều khiển 8051.**

Động cơ bước thực hiện dưới dạng half-step. Có 8 hướng là OUT1 OUT1OUT2 OUT2 OUT2OU3 OUT3 OUT3OU4 OUT4 OUT4OUT1. Trong đó có 4 hướng chính là OUT1 OUT2 OUT3 OUT4. Ở đây chúng ta nối 2 chân của động cơ trên mô phỏng lại 1 chân nối nguồn, thì thành động cơ bước 5 chân

K1: Để xoay theo chiều kim đồng hồ

K2: Để xoay theo ngược chiều kim đồng hồ

K3: Quay với tốc độ cao

K4: Quay với tốc độ thấp

**Ảnh có chứa văn bản, biểu đồ, hàng, Kế hoạch

Mô tả được tạo tự động**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tên san pham: Bo dieu khien he thong

Su dung IO: Su dung P1 cho he thong chính, su dung P3.0, P3.1, P3.2, P3.3 cho các chuc nang phu.

Ket qua dieu khien: K1 du?i, th?i gian thay d?i, K2 du?i, th?i gian thay d?i, K3 du?i, th?p, K4 du?i, cao.

Chú ý: Do s? d?ng P3.2 d? k?t n?i bên ngoài, do dó các dèn LED bên ngoài s? không sáng khi h? th?ng chuy?n sang ch? d? ngoài vi?c di?u khi?n b?i P3.2.

K?t qu? di?u khi?n t?t nh?t s? du?c hi?n th? du?i dây sau khi ra kh?i ch? d? bên ngoài

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "reg52.h"

//µç»úIO

#define GPIO\_MOTOR P1

//sbit F1 = P1^0;

//sbit F2 = P1^1;

//sbit F3 = P1^2;

//sbit F4 = P1^3;

//°´¼üIO

sbit K1=P3^0;

sbit K2=P3^1;

sbit K3=P3^2;

sbit K4=P3^3;

unsigned char code FFW[8]={0xf1,0xf3,0xf2,0xf6,0xf4,0xfc,0xf8,0xf9}; //·Xoay tu phai qua trai theo half-step

unsigned char code FFZ[8]={0xf9,0xf8,0xfc,0xf4,0xf6,0xf2,0xf3,0xf1}; //Xoay tu trai qua phai theo half-step

unsigned char Direction,Speed;

void Delay(unsigned int *t*);

void  Motor();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tên hàm chính: main

Chuc nang cua hàm: Hàm chính

Tham so dau vào: Không có

Giá tri tra ve: Không có

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

    unsigned char i;

    Speed=30;

  while(1)

    {

        if(K1==0)       //Kiem tra nut K1 co nhan hay khong

        {

            Delay(1);   //Ham delay don nut

            if(K1==0)

            {

                Direction=1;

            }

            while((i<200)&&(K1==0))  //Kiem tra nut co duoc nha hay khong (trong 200ms)

            {

                Delay(1);

                i++;

            }

            i=0;

        }

        if(K2==0)       //¼ì²â°´¼üK1ÊÇ·ñ°´ÏÂ

        {

            Delay(1);   //Ïû³ý¶¶¶¯

            if(K2==0)

            {

                Direction=2;

            }

            while((i<200)&&(K2==0))  //¼ì²â°´¼üÊÇ·ñËÉ¿ª

            {

                Delay(1);

                i++;

            }

            i=0;

        }

        if(K3==0)       //¼ì²â°´¼üK1ÊÇ·ñ°´ÏÂ

        {

            Delay(1);   //Ïû³ý¶¶¶¯

            if(K3==0)

            {

                Speed=13;

            }

            while((i<200)&&(K3==0))  //¼ì²â°´¼üÊÇ·ñËÉ¿ª

            {

                Delay(1);

                i++;

            }

            i=0;

        }

        if(K4==0)       //¼ì²â°´¼üK1ÊÇ·ñ°´ÏÂ

        {

            Delay(1);   //Ïû³ý¶¶¶¯

            if(K4==0)

            {

                Speed=40;

            }

            while((i<200)&&(K4==0))  //¼ì²â°´¼üÊÇ·ñËÉ¿ª

            {

                Delay(1);

                i++;

            }

            i=0;

        }

        Motor();

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tên chuc nang: Dong co

Chuc nang chính: Quay dong co

Ðau vào: Không

Ðau ra: Không

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void  Motor()

{

    unsigned char i;

    for(i=0;i<8;i++)

    {

        if(Direction==1)

            GPIO\_MOTOR = FFW[i]&0x1f;  //Truyền dữ liệu ra mạch đệm

        if(Direction==2)

            GPIO\_MOTOR = FFZ[i]&0x1f;

        Delay(Speed);   //µ÷½Ú×ªËÙ

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tên hàm: Delay

Chuc nang hàm: Ðoi mot khoang thoi gian

Tham so vào: t

Tham so ra: Không

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Delay(unsigned int *t*)

{

    unsigned int k;

    while(*t*--)

    {

        for(k=0; k<80; k++)

        { }

    }

}