

**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**KHOA CÔNG NGHỆ**  
**BỘ MÔN ĐIỆN TỬ VIỄN THÔNG**



**ĐỒ ÁN KỸ THUẬT VI XỬ LÝ**

**GVHD:** TRẦN HỮU DANH

**SVTH:** NGUYỄN LÊ NGUYỄN  
**MSSV:** B1709497

Cần thơ, ngày 2 tháng 7 năm 2020

# XE TRÁNH VẬT CẢN ( OBSTACLE AVOIDING CAR )

## 1 GIỚI THIỆU

Theo dự đoán, sau cuộc cách mạng vĩ đại trên internet sẽ là cuộc cách mạng trong lĩnh vực robot. Robot đang được áp dụng trong cuộc sống hàng ngày càng giống như vận chuyển robot giao hàng hóa, robot kiểm tra nguy hiểm, robot xe lăn cho người khuyết tật. Robot phục vụ các hoạt động gia đình ... Hạn chế của robot tự hành hiện tại là thiếu linh hoạt và khả năng thích ứng khi làm việc ở các vị trí khác nhau. Từ những lý do đó, các vấn đề phát sinh để tránh chướng ngại vật cho robot tự hành để tăng tính linh hoạt cho robot. Hầu hết các robot hiện đại đều có một số loại tránh chướng ngại vật dừng ở khoảng cách ngắn từ chướng ngại vật để tránh va chạm, đến các thuật toán tinh vi hơn, cho phép robot di chuyển dọc theo biên giới xung quanh chướng ngại vật. Để từng bước tiếp cận và tạo cơ sở cho công nghệ quy trình nghiên cứu ứng dụng vào các sản phẩm thực tế, tôi đã chọn: "Xe tự hành tránh vật cản".

## 2 GIỚI THIỆU LINH KIỆN VÀ SƠ ĐỒ KHỐI

### 2.1 Giới thiệu linh kiện

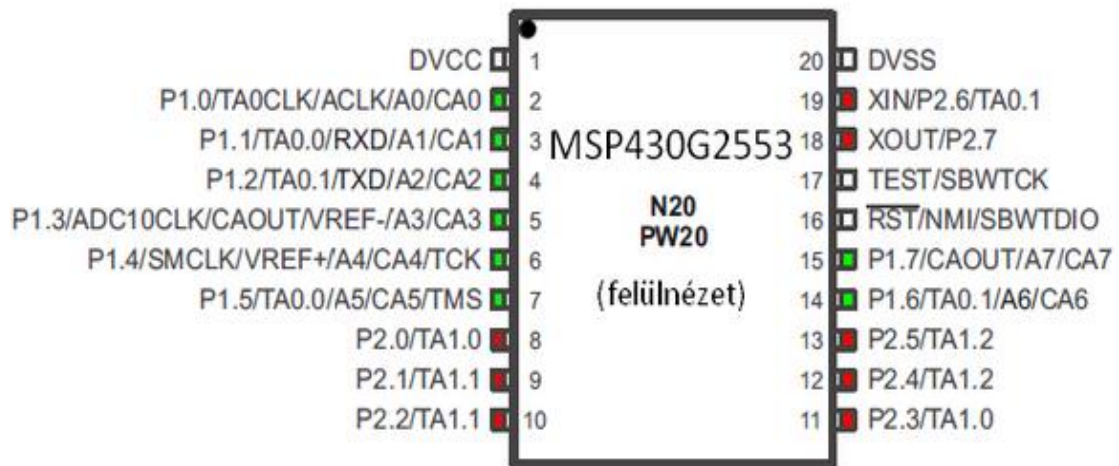
Để hoàn thành đồ án này chúng ta phải sử dụng IC Msp430G2553, module cảm biến siêu âm HC-SR04, module Driver L298N, động cơ Servo G90, motor giảm tốc, phần mềm Protues nhằm mô phỏng kết quả và vẽ mạch, phần mềm IAR.

#### IC MSP430G2553:

- + Cấu trúc sử dụng nguồn thấp giúp kéo dài tuổi thọ của Pin
  - Duy trì 0.1 $\mu$ A dòng nuôi RAM.
  - Chỉ 0.8 $\mu$ A real-time clock.
  - 250  $\mu$ A/ MIPS.
- + Bộ tương tự hiệu suất cao cho các phép đo chính xác
  - 12 bit hoặc 10 bit ADC-200 kskp, cảm biến nhiệt độ, Vref ,
  - 12 bit DAC.
  - Bộ giám sát điện áp nguồn.
- + 16 bit RISC CPU cho phép được nhiều ứng dụng, thể hiện một phần ở kích thước Code lập trình.
  - Thanh ghi lớn nên loại trừ được trường hợp tắt nghẽn tập tin khi đang làm việc.
  - Thiết kế nhỏ gọn làm giảm lượng tiêu thụ điện và giảm giá thành.
  - Tối ưu hóa cho những chương trình ngôn ngữ bậc cao như C, C++
  - Có 7 chế độ định địa chỉ.
  - Khả năng ngắt theo véc tơ lớn.
- + Trong lập trình cho bộ nhớ Flash cho phép thay đổi Code một cách linh hoạt, phạm vi rộng, bộ nhớ Flash còn có thể lưu lại như nhật ký của dữ liệu.

1.Sơ đồ chân : Chip MSP430 có kích thước nhỏ gọn , chỉ với 20 chân đối với kiểu chân DIP.

Bao gồm 2 port I/O (hay GPIO general purpose input/ output : cổng nhập xuất chung).



Ta thấy rằng mỗi port đều có 8 chân.

Port 1 : có 8 chân từ P1.0 đến P1.7 tương ứng với các chân từ 2-7 và 14 , 15.

Port 2 : cũng gồm có 8 chân P2.0 – P2.7 ứng với các chân 8 – 13 , 18,19.

Ngoài chức năng I/O thì trên mỗi pin của các port đều là những chân đa chức năng, ta thể thấy rõ trong hình trên

là chức năng của từng chân , ngoài ra đối với các MCU có kiểu chân SMD loại 28 chân thì nó có thêm port 3 nữa, nhưng ở đây chỉ xét đối với kiểu DIP 20 chân và kiểu TSSOP 20 chân , kiểu còn lại các bạn có thể tìm hiểu trên trang chủ của nhà sản xuất .

### Module cảm biến siêu âm HC-SR04:

**Cảm biến siêu âm HC-SR04** (Ultrasonic Sensor) được sử dụng rất phổ biến để xác định khoảng cách vì giá thành rẻ và khá chính xác. Cảm biến siêu âm HC-SR04 sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300cm.



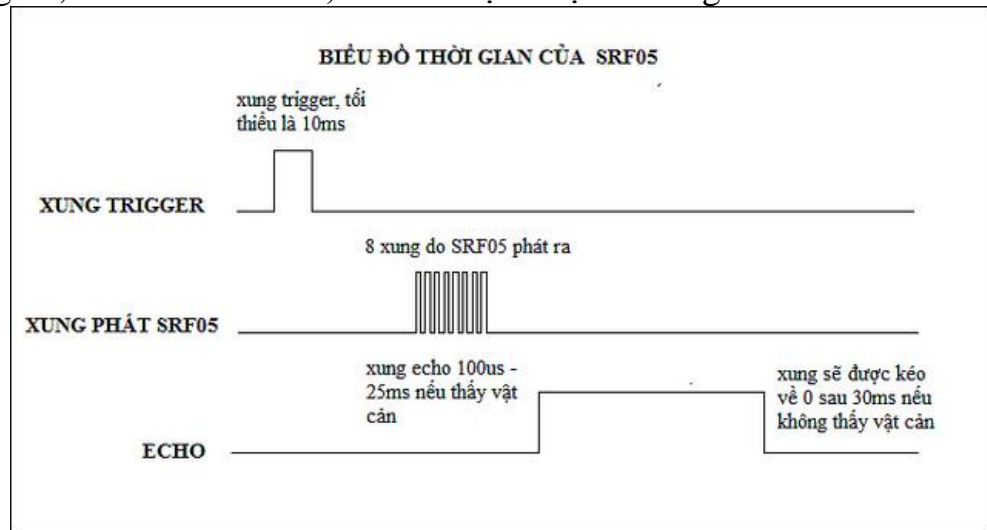
Vcc	5V
Trig	Một chân Digital output
Echo	Một chân Digital input
GND	GND

Cảm biến HC-SR04 có 4 chân là: Vcc, Trig, Echo, GND.

## Nguyên lý hoạt động

Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (5 microSeconds) từ chân Trig. Sau đó, **cảm biến siêu âm** sẽ tạo ra 1 xung HIGH ở chân Echo cho đến khi nhận lại được sóng phản xạ ở pin này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và quay trở lại.

Tốc độ của âm thanh trong không khí là 340 m/s (hằng số vật lý), tương đương với 29,412 microSeconds/cm ( $106 / (340 * 100)$ ). Khi đã tính được thời gian, ta sẽ chia cho 29,412 để nhận được khoảng cách.



## Mạch điều khiển động cơ bước và động cơ một chiều L298N

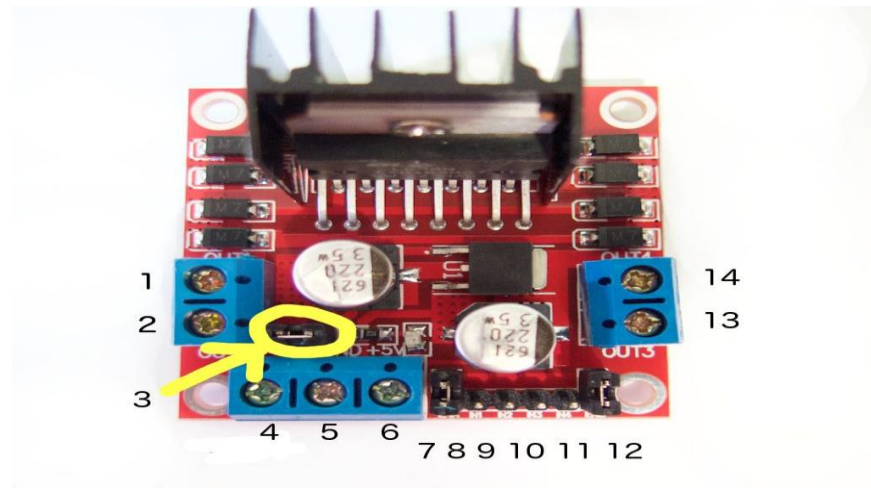
Module điều khiển động cơ (Motor Driver) sử dụng chip cầu H L298N giúp điều khiển tốc độ và chiều quay của động cơ DC một cách dễ dàng, ngoài ra module L298N còn điều khiển được 1 động cơ bước lưỡng cực. Mạch cầu H của IC L298N có thể hoạt động ở điện áp từ 5V đến 35V.

Module L298N có tích hợp một IC nguồn 78M05 để tạo ra nguồn 5V để cung cấp cho các thiết bị khác.

## Thông số kỹ thuật của mạch điều khiển động cơ L298N

- Driver: L298N tích hợp hai mạch cầu H
- Điện áp điều khiển: +5 V ~ +35 V
- Dòng tối đa cho mỗi cầu H: 2A
- Điện áp tín hiệu điều khiển: +5 V ~ +7 V
- Dòng tín hiệu điều khiển: 0 ~ 36mA
- Công suất hao phí: 20W (khi nhiệt độ  $T = 75^{\circ}\text{C}$ )
- Nhiệt độ vận hành:  $-25^{\circ}\text{C} \sim +130^{\circ}\text{C}$

## Sơ đồ chân tín hiệu module L298N



1. DC motor 1 “+” hoặc stepper motor A+.
2. DC motor 1 “-” hoặc stepper motor A-.
3. 12V jumper – tháo jumper này nếu sử dụng nguồn trên 12V cấp vào chân 4. Jumper này dùng để cấp nguồn cho IC ổn áp tạo ra nguồn 5V nếu nguồn trên 12V sẽ làm cháy IC 78M05.
4. Cấp dương nguồn cho motor vào đây từ 6V đến 35V.
5. Cắm chân GND (đất, cực âm) của nguồn vào đây.
6. Nguồn ra 5V, nếu jumper (3) được cắm thì có nguồn ra 5V ở đây.
7. Chân Enable của Motor 1, chân này cũng dùng để cấp xung PWM cho motor. Nếu điều khiển tốc độ thì rút jumper ra và cắm chân PWM của VĐK vào đây. Giữ nguyên khi dùng với động cơ bước.
8. IN1
9. IN2
10. IN3
11. IN4
12. Chân Enable của Motor 2, chân này cũng dùng để cấp xung PWM cho motor. Nếu điều khiển tốc độ thì rút jumper ra và cắm chân PWM của VĐK vào đây. Giữ nguyên khi dùng với động cơ bước.
13. DC motor 2 “+” hoặc stepper motor B+.
14. DC motor 2 “-” hoặc stepper motor B-.

## Động cơ Servo 9G:

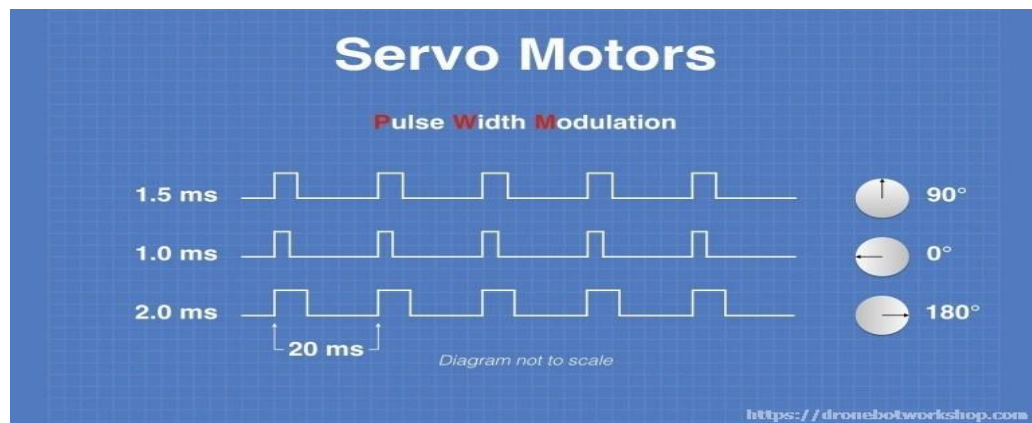
Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng PWM) với góc quay nằm trong khoảng bất kì từ 0o - 180o.

### Thông số kỹ thuật:

- Điện áp hoạt động: 4.8-5VDC
- Tốc độ: 0.12 sec/ 60 degrees (4.8VDC)
- Lực kéo: 1.6KG.CM
- Kích thước: 21x12x22mm
- Trọng lượng: 9g.

Nguyên lý hoạt động:

Trong các động cơ Analog Servo, tín hiệu **PWM** có chu kỳ 20ms được sử dụng để điều khiển động cơ. Độ rộng của xung được thay đổi trong khoảng từ 1 đến 2ms để điều khiển vị trí trục động cơ.



- Độ rộng xung 1,5ms sẽ làm cho trục servo nằm ở vị trí 90 độ.
- Độ rộng xung 1ms sẽ làm cho trục servo nằm ở vị trí 0 độ.
- Độ rộng xung 2ms sẽ làm cho trục servo nằm ở vị trí 180 độ.

Việc thay đổi độ rộng xung giữa 1ms và 2ms sẽ di chuyển trục servo qua góc giới hạn 180 độ. Bạn có thể điều chỉnh ở bất kỳ góc nào bạn muốn bằng cách điều chỉnh độ rộng xung cho phù hợp.

## Động cơ giảm tốc vàng

Động Cơ Giảm Tốc Vàng 3V-9V là mẫu động cơ được sử dụng nhiều nhất để thiết kế các loại *robot mô hình*. Phục vụ cho sinh viên học tập và nghiên cứu chế tạo.

Với giá thành rẻ, dễ sử dụng và mang tính ưu việt nên chiếc động cơ này được sử dụng rất nhiều với sinh viên các ngành kỹ thuật.



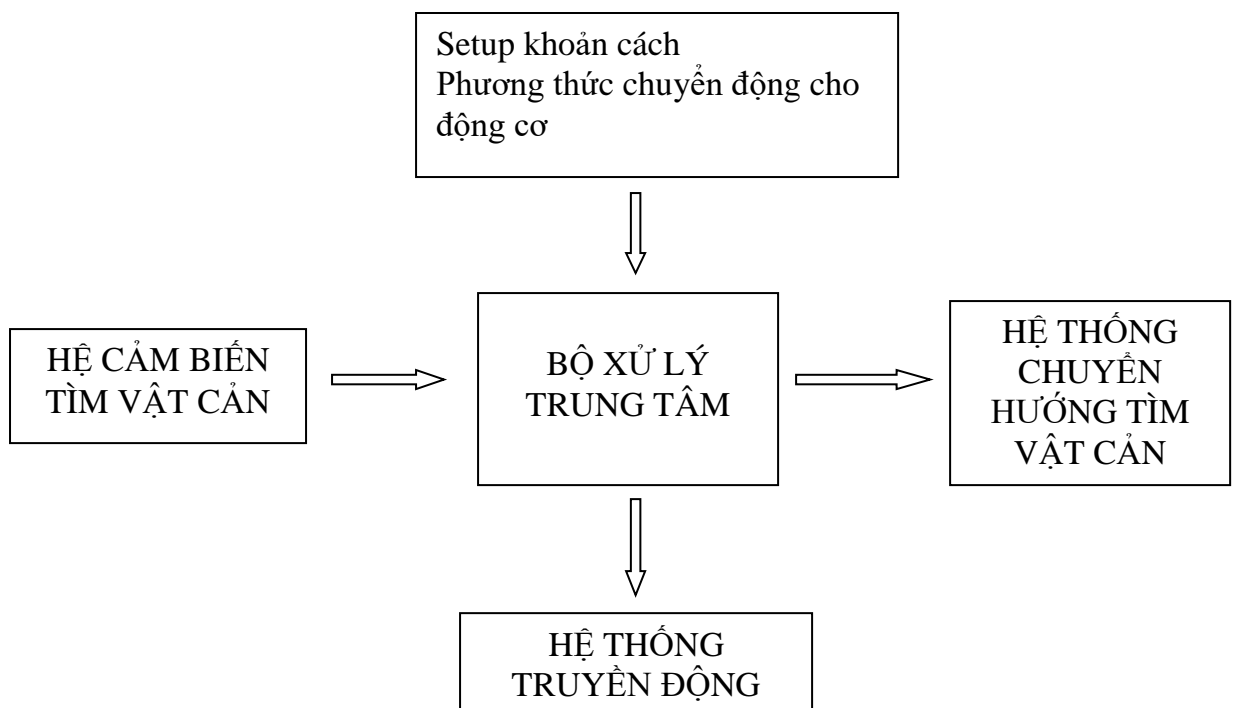
Động cơ vàng

### Thông số kỹ thuật của động cơ giảm tốc Vàng

- Điện áp hoạt động: 3V~ 9V DC (Hoạt động tốt nhất từ 6 - 8V)
- Mômen xoắn cực đại: 800gf cm min 1:48 (3V)
- Tốc độ không tải: 125 Vòng/ 1 Phút (3V)  
(Với bánh 66mm: 26m/1p)
- 208 Vòng/ 1 Phút (5V)  
(Với bánh 66mm: 44m/1p)
- Dòng không tải **động cơ**: 70mA (250mA MAX)

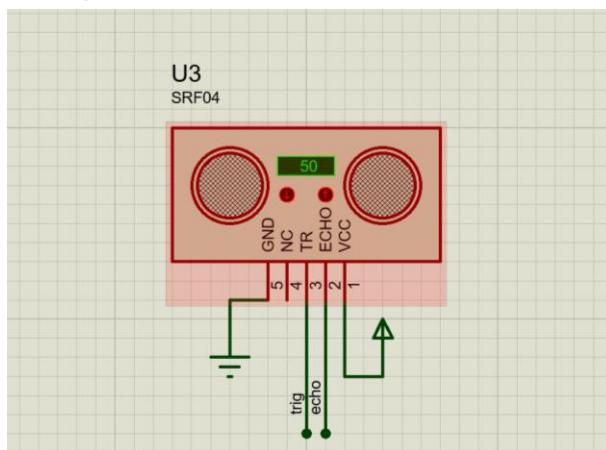
## 2.2 Sơ đồ khối

Sơ đồ khối :

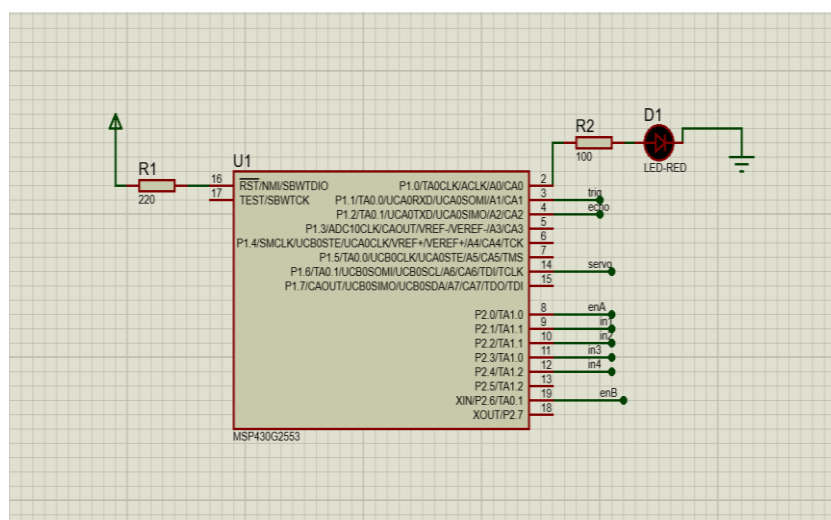




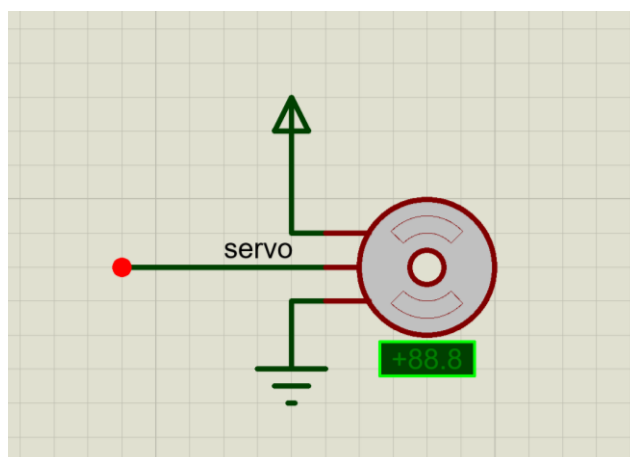
## Hệ cảm biến tìm đường



## Bộ xử lý trung tâm (MSP430G2553)

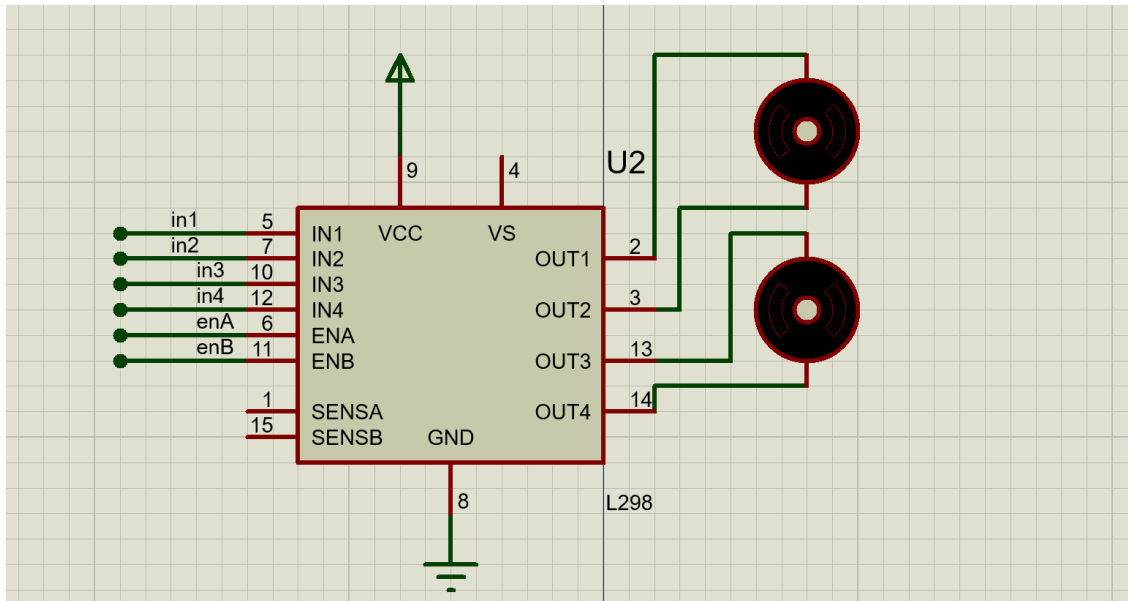


## Hệ thống chuyển hướng tìm đường

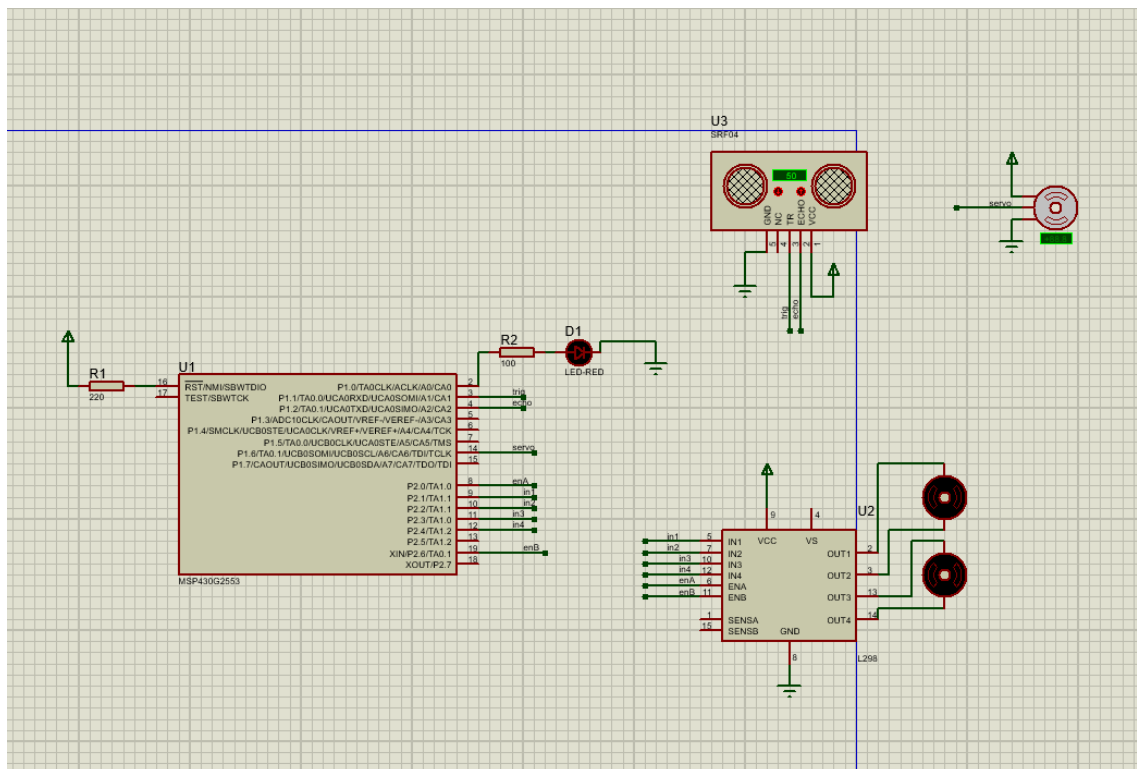




## Hệ thống truyền động



## Sơ đồ mạch hoàn chỉnh



**LẬP TRÌNH**

```

void main(void)
{
    BCSCCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;
    // submainclock 1mhz
    WDTCTL = WDTPW + WDTHOLD;
    // Stop WDT

    CCTLO = CCIE;
    // CCR0 interrupt enabled
    CCR0 = 1000;
    // 1ms at 1mhz
    TACTL = TASSEL_2 + MC_1;
    // SMCLK, upmode
    P1IFG = 0x00;
    //clear all interrupt flags
    P1DIR |= 0x01;
    // P1.0 as output for LED
    P1OUT &= ~0x01;
    // turn LED off

    _BIS_SR(GIE);
    // global interrupt enable

    P1DIR |= BIT6;
    P1SEL |= BIT6;
    TA0CCR0 = 20000;
    TA0CCR1 = 1500;
    // CCR1 PWM duty cycle
    TA0CCTL1 = OUTMOD_7;
    // CCR1 reset/set
    TA0CTL = TASSEL_2 + MC_1;
    // SMCLK, up mode

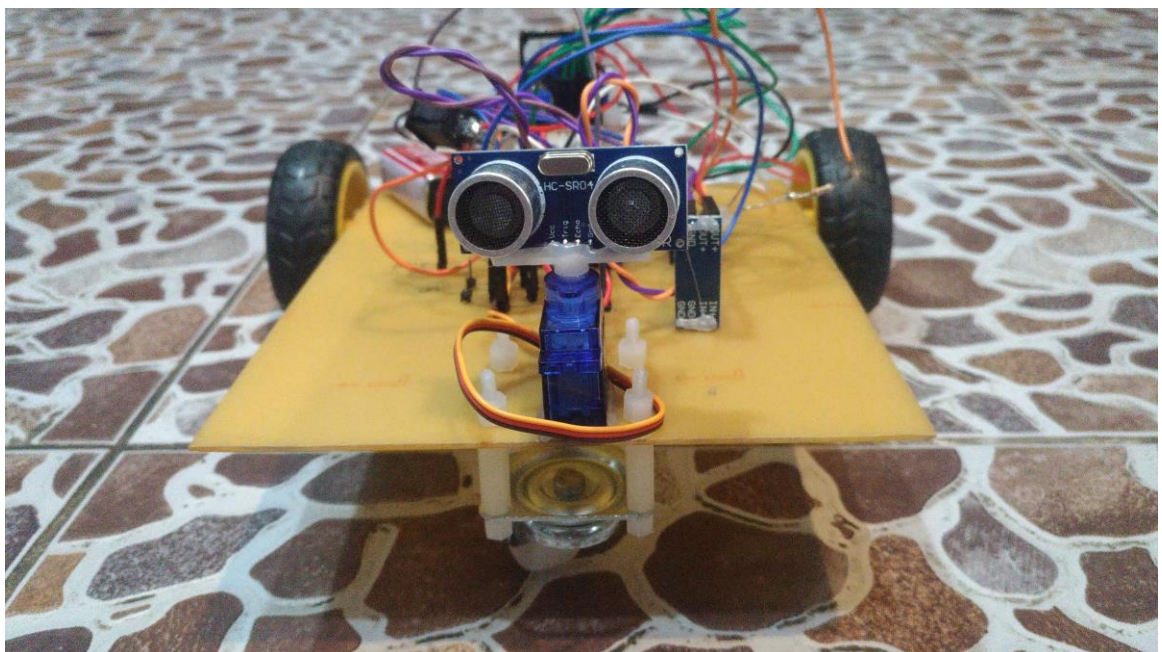
    while(1){
        int disL=0;
        int disR=0;

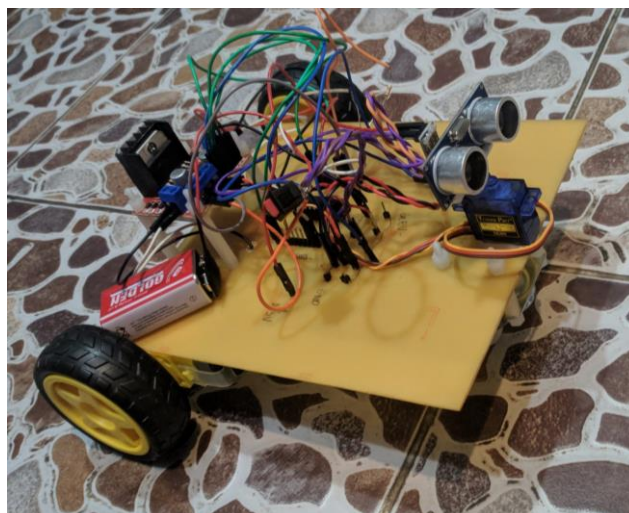
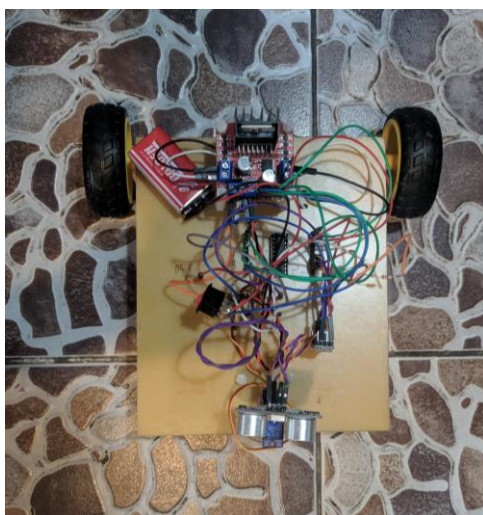
```

```

        init();
        if(distance <= 15 && distance != 0){
            P1OUT |= 0x01;
            BotLocomotion(STOP);
            Delay(5);
            BotLocomotion(REVERSE);
            Delay(2);
            BotLocomotion(STOP);
            Delay(5);
            disR = lookR();
            Delay(5);
            disL = lookL();
            Delay(5);
            if(disR>=disL)
            {
                BotLocomotion(RIGHT_TURN);
                Delay(3);
                BotLocomotion(STOP);
                P1OUT &= ~0x01;
            }
            else
            {
                BotLocomotion(LEFT_TURN);
                Delay(3);
                BotLocomotion(STOP);
                P1OUT &= ~0x01;
            }
        }
        else
        {
            P1OUT &= ~0x01;
            BotLocomotion(FORWARD);
        }
    }
}

```

**HÌNH ẢNH THỰC TẾ**



## KẾT QUẢ

Mạch đáp ứng được nguyên lý hoạt động đề ra , tuy nhiên hệ thống truyền động và tìm đường hoạt động chưa được trọn chu . Hệ thống nguồn vẫn còn chưa ổn định, qua nhiều module trung gian nên gây hao tổn pin.

## TÀI LIỆU THAM KHẢO

<https://www.instructables.com/circuits/>

<https://www.xanthium.in/>

Giáo trình vi điều khiển dùng msp430 - Trần Hữu Danh