

CHƯƠNG 8

ROBOT DI ĐỘNG

Robot di động là đối tượng nghiên cứu đa dạng, được định nghĩa là máy tự động có khả năng di chuyển trên không, trên mặt đất, dưới nước..., robot di động được phân chia làm robot tự hành và robot có điều khiển. Robot tự hành có khả năng di chuyển trong môi trường không biết trước, được trang bị trí thông minh ở một mức độ nào đó, có khả năng học để nâng cao nhận thức về môi trường xung quanh và hành động nhằm đạt mục đích nào đó. Robot có điều khiển được điều khiển từ xa thông qua các kênh liên lạc trực tiếp bằng mắt, camera và kết nối với trạm điều khiển qua kênh truyền thông không dây.

Phạm vi nghiên cứu robot di động rất rộng và gồm các lĩnh vực chính sau đây:

- Cơ cấu chuyển động
- Xác định vị trí
- Qui hoạch quỹ đạo
- Nhận thức và hành động

Việc nghiên cứu chế tạo robot đòi hỏi các kiến thức về cơ khí, điện điện tử, công nghệ thông tin. Trong chương này chúng ta tìm hiểu các nét chính về phân loại, cấu trúc robot, các cảm biến thông dụng trang bị cho robot và các phương pháp điều khiển chuyển động robot.

8.1 CƠ CẤU DI CHUYỂN

8.1.1 Di chuyển trên mặt đất

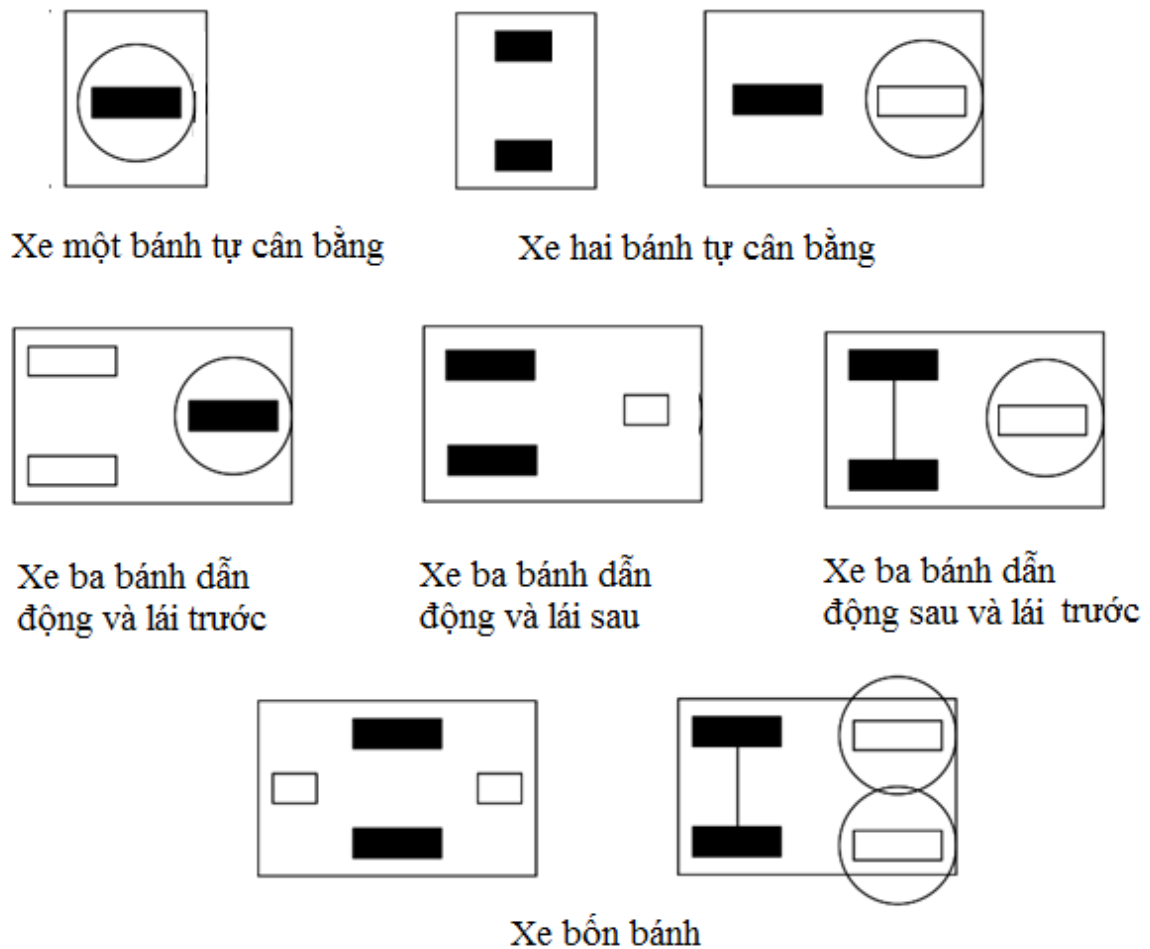
Robot có thể di chuyển trên mặt đất dùng bánh xe, bánh xích, bước bằng chân hoặc trườn như rắn. Cơ cấu bánh xe, bánh xích thông dụng hơn cả vì dễ chế tạo và có tính cơ động cao, bánh xe quay tạo chuyển động dọc tới lui cho robot nhờ ma sát, ngoài ra cần có cơ cấu đổi hướng chuyển động rẽ trái phải nhờ bánh lái và bộ vi sai, thông thường dùng ba hoặc bốn bánh xe để tạo cân bằng trong đó có hai bánh xe chủ động được điều khiển riêng hoặc một bánh xe được điều khiển quay và chuyển hướng.

Cơ cấu vi sai sử dụng trong xe ô tô, khi đi vào đường vòng, bánh xe ngoài đi quãng đường lớn hơn nên cần có vận tốc quay lớn hơn vận tốc quay bánh xe trong. Đối với robot di động, bộ vi sai thường được tạo ra bằng cách điều khiển phối hợp hai động cơ quay bánh xe.

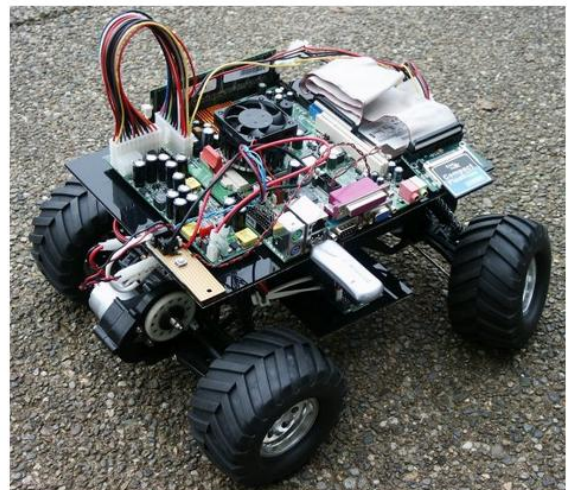
Robot bánh xe cần ít nhất hai động cơ. Trong Hình 8.1 bánh xe tô đen tạo chuyển động dọc, bánh xe khoanh tròn là bánh lái có điều khiển, bánh xe trắng là bánh xe không điều khiển dùng để cân bằng. Cơ cấu bánh xe có giới hạn là không thể chuyển động theo hướng trục quay của bánh xe, muốn chuyển hướng phải thực hiện phép quay, cơ cấu chuyển động này gọi là chuyển động *non holonome*, ta không thể thay đổi hướng của robot mà không thay đổi vị trí. Một cơ cấu chuyển động khác là bánh xe omni đa hướng Hình 8.4 giúp robot chuyển động ngang mà không cần quay

Robot Di Động

dùng ba hay bốn bánh, ta nói là robot có chuyển động *holonome* vì có thể điều khiển độc lập vị trí x , y và góc của robot



Hình 8.1 Các kiểu truyền động bánh xe



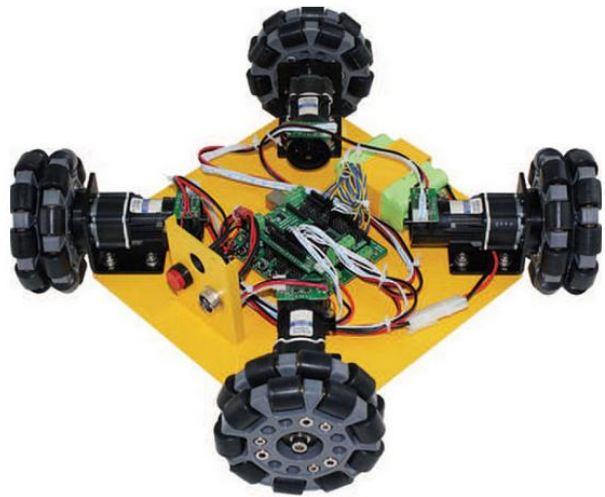
Hình 8.2 Robot hai bánh và bốn bánh

Một loại robot bốn bánh thường dùng trong các phân xưởng để chở hàng gọi là AGV (Automatic Guided Vehicle), robot này được dẫn đường nhờ sóng vô tuyến phát

ra từ đường dây đi dưới sàn hay các băng dán trên mặt sàn có từ tính hoặc có màu phân biệt với sàn.

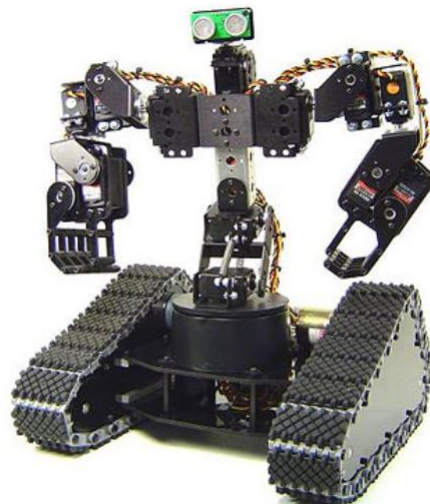
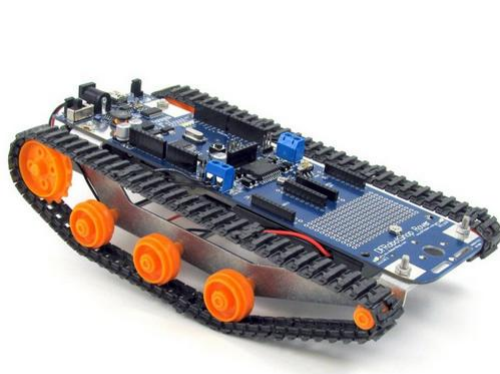


Hình 8.3 Robot chở hàng AGV



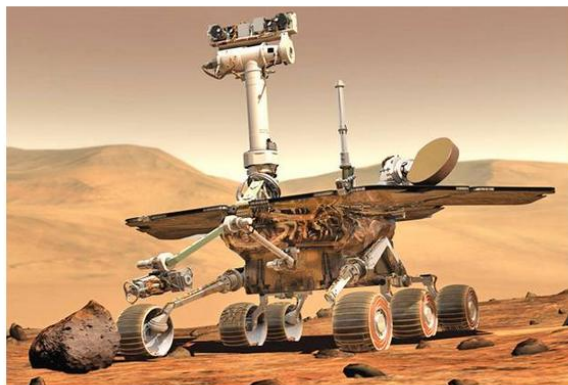
Hình 8.4 Robot dùng bánh xe đa hướng

Khi di chuyển trên đường gồ ghề, trơn trượt ta dùng hai bánh xích Hình 8.5 hoặc dùng nhiều bánh xe kết hợp thay đổi cơ cấu vị trí giữa các bánh xe có thể leo vượt qua chướng ngại hay leo cầu thang Hình 8.6



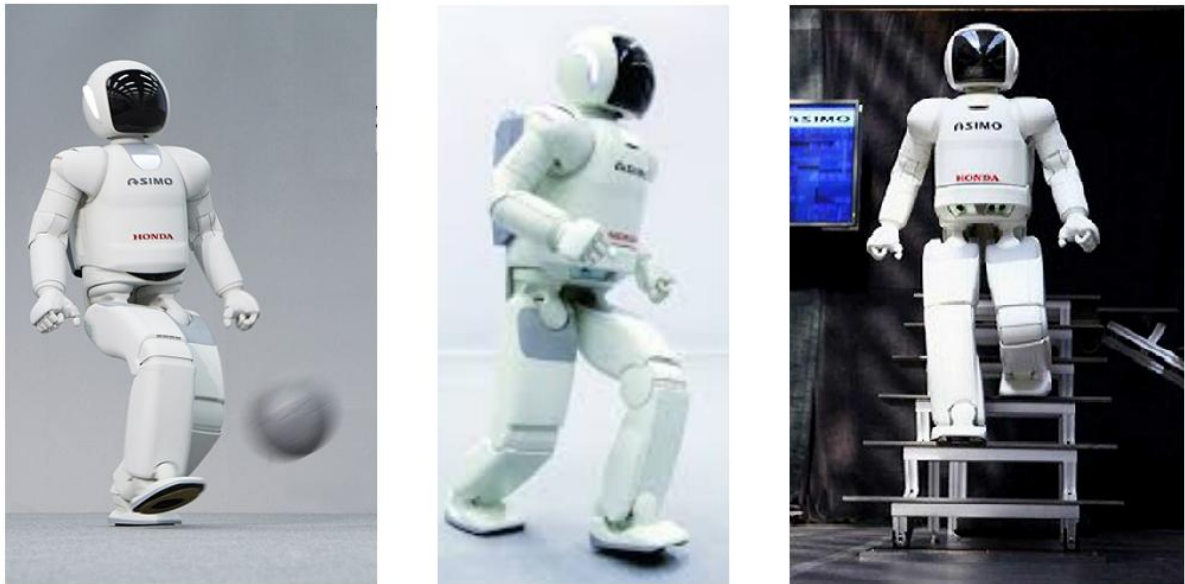
Hình 8.5 Robot dùng bánh xích

Robot Di Động



Hình 8.6 Robot địa hình

Một cơ cấu chuyển động khác là bước bằng hai chân hay nhiều chân, robot hai chân là thách thức lớn vì phải bảo đảm cân bằng khi chuyển động và robot ASIMO đã thực hiện chuyển động đi chạy, leo cầu thang đứng một chân rất tốt.



Hình 8.7 Robot hai chân ASIMO

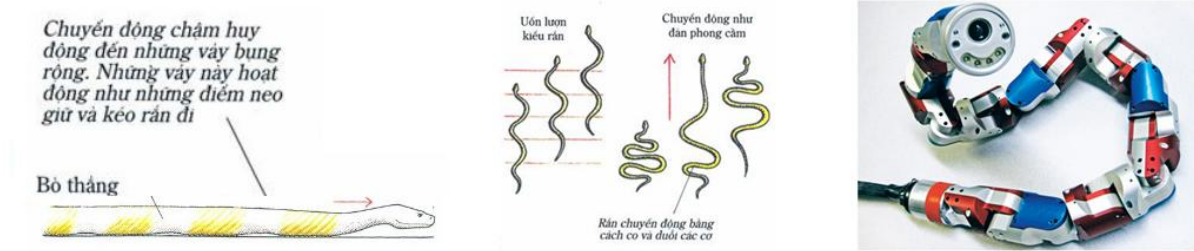
Robot có chân kết cấu phức tạp, nhiều bậc tự do, cần nhiều động cơ và điều khiển phức tạp hơn robot bánh xe.



Hình 8.8 Robot nhiều chân

Rắn có thể bò vào các khe hẹp, leo cây, do đó nhiều nhà nghiên cứu đã chế tạo các robot rắn, chuyển động bắt chước gần giống chuyển động con rắn. Con rắn chuyển động bằng cách uốn lượn thân mình, co duỗi các cơ hay vẩy bụng và nhờ sự ma sát để chuyển động. Rắn còn có thể phóng tới trước bằng cách dùng đuôi lần điềm tựa. Các đường ống nhỏ thường dùng các robot chui đường ống để kiểm tra rò rỉ. Các robot này thường gồm nhiều module kết nối mềm dẻo với nhau.

Robot Di Động



Hình 8.9 Robot rắn



Hình 8.10 Robot kiểm tra đường ống

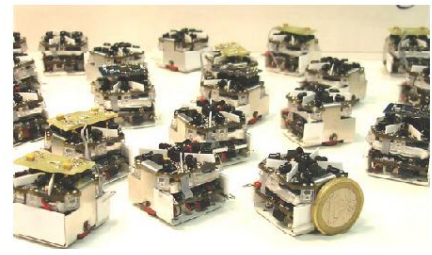
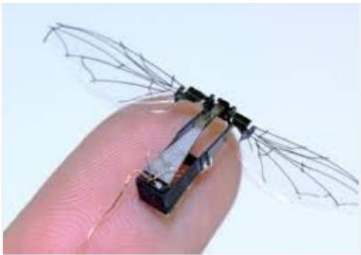
8.1.1 Di chuyển trên không

Robot bay có tính cơ động cao và có nhiều ứng dụng, trong quân sự ta có UAV Unmanned Aerial Vehicle là những máy bay nhỏ dùng trong do thám hay tấn công có người điều khiển, các dụng cụ bay bốn cánh quạt quadrotor, drone dùng để quay phim hay giao hàng, ngoài ra các robot bay có vồ cánh như côn trùng hay chim cũng được nghiên cứu. Các robot bay dùng nguồn năng lượng là pin hay xăng, đôi khi kết hợp pin mặt trời, sử dụng động cơ điện hay động cơ đốt trong, động cơ phản lực, sức nâng được tạo ra nhờ cánh hay chong chóng, robot chuyển động nhờ phản lực hay độ nghiêng của chong chóng.

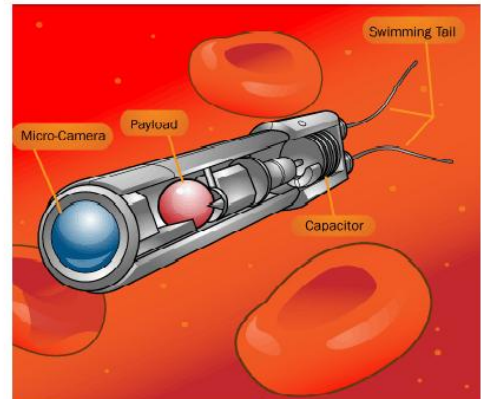
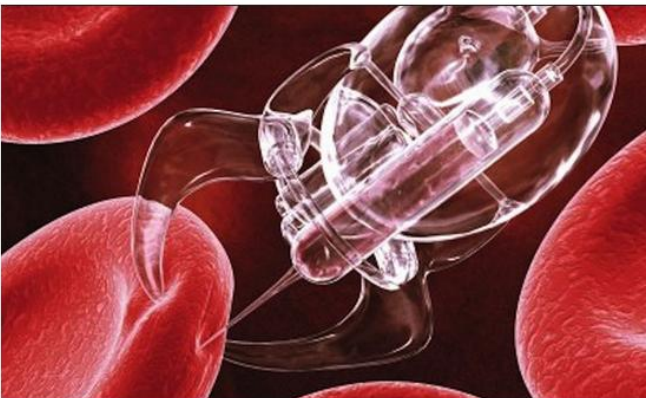


Hình 8.11 UAV

Bên cạnh các robot kích thước thông thường còn có các micro robot kích thước vài mm dùng trong nghiên cứu khoa học, do thám hay nano robot kích thước nhỏ hơn mm dùng trong y học, chủ yếu di chuyển trong mạch máu, có thể dùng để phá cục máu đông, huỷ tế bào ung thư, mang thuốc hay chất phóng xạ đến vị trí cần thiết, các robot này dùng động cơ là tinh thể áp điện



Hình 8.12 Micro Robot



Hình 8.13 Nano Robot

8.2 CẢM BIẾN

Cảm biến rất quan trọng đối với robot giống như giác quan đối với con người. Ta phân biệt hai loại cảm biến:

-Cảm biến trong (Proprioceptive sensors): cảm biến trạng thái bên trong của robot như điện áp nguồn, nhiệt độ bo mạch, khói, lửa, encoder đo góc quay động cơ, đo độ nghiêng, gia tốc kế, gyroscope đo hướng trong không gian, la bàn đo hướng so với cực từ trái đất...

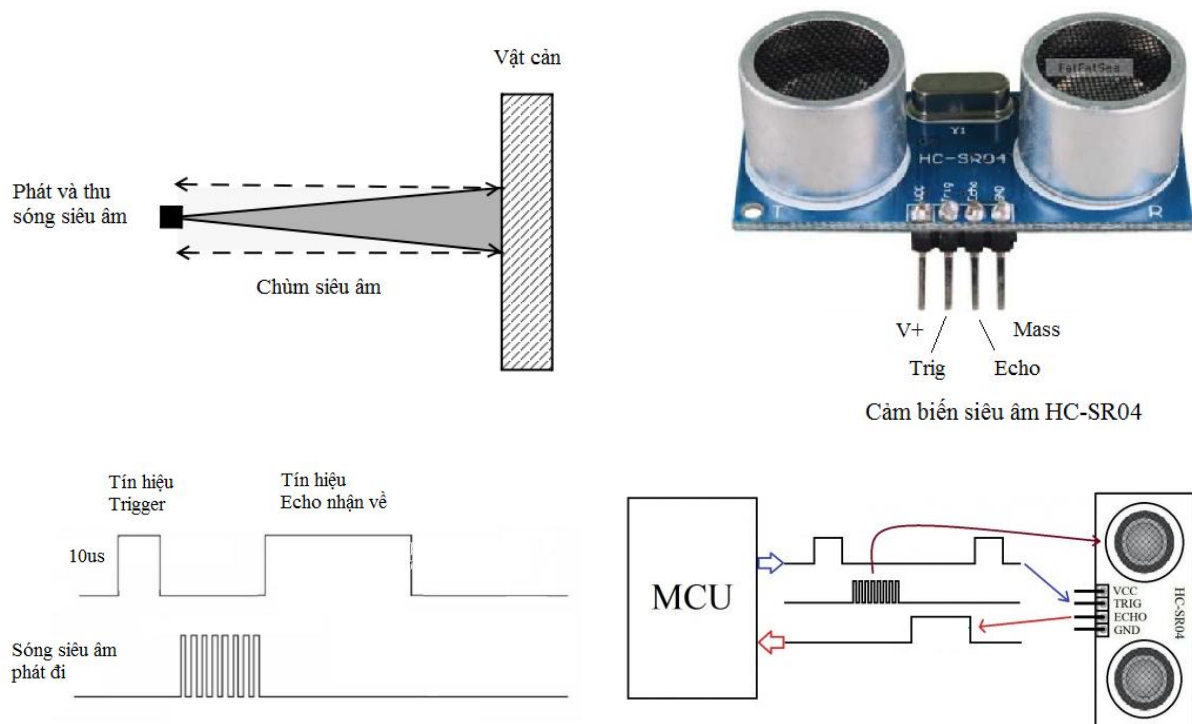
Robot Di Động

- Cảm biến ngoài (Exteroceptive sensors): dựa vào tương tác với bên ngoài như cảm biến chạm, camera, siêu âm, hồng ngoại, laser, hệ định vị GPS...

Trong mục này chúng ta chỉ khảo sát một số cảm biến quan trọng.

8.2.1 Cảm biến siêu âm đo khoảng cách

Cảm biến siêu âm dùng để đo khoảng cách, ta phát một chùm sóng siêu âm hình nón tần số vài chục KHz khoảng vài μs và đo thời gian nhận sóng phản hồi t từ vật có khoảng cách d , gọi v vận tốc truyền sóng, ta có $d = \frac{vt}{2}$. Trong không khí vận tốc truyền âm khoảng 343m/s và thay đổi theo nhiệt độ cũng như độ ẩm. Khoảng cách có thể đo từ vài cm đến vài mét. Cảm biến phát và thu siêu âm thường dùng loại áp điện có kích thước nhỏ gọn. Cảm biến HC-SR04 có thể đo khoảng cách từ 2cm đến 4m, khi chân Trig nhận xung kích, cảm biến phát ra 8 chu kỳ sóng siêu âm 40KHz, góc chùm tia khoảng 30° , chân Echo lên 1 sau đó nếu nhận được sóng phản hồi thì chân Echo xuống không, vậy bề rộng xung Echo tỷ lệ khoảng cách, ta dùng Vi xử lý để tạo xung Trig và đo bề rộng t xung Echo, khoảng cách là $d_{cm} = 5 \cdot 10^{-3} v_{m/s} t_{us} = 0.0172 t_{us} = t_{us} / 58$. Nếu bề rộng xung Echo lớn hơn 38ms tức là không có phản hồi. Mỗi giây ta có thể đo 20 lần.



Hình 8.14 Cảm biến siêu âm

Đoạn code sau dùng MikroC minh họa lập trình giao tiếp PIC 16F877 8MHz với cảm biến, chân Trig nối RB0, chân Echo nối RB4, khoảng cách hiển thị lên LCD nối vào PortD, khi chân Echo đổi mức thì gây ngắt khởi động và ngừng timer đo thời gian và tính khoảng cách (<https://electrosome.com/hc-sr04-ultrasonic-sensor-pic/>)

```
// LCD module connections  
sbit LCD_RS at RD2_bit;
```



```

sbit LCD_EN at RD3_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;

sbit LCD_RS_Direction at TRISD2_bit;
sbit LCD_EN_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
// End LCD module connections
int a;

//Interrupt function will be automatically executed on Interrupt
void interrupt()
{
    if(INTCON.RBIF == 1)    //Makes sure that it is PORTB On-Change Interrupt
    {
        INTCON.RBIE = 0;    //Disable On-Change Interrupt
        if(PORTB.F4 == 1)    //If ECHO is HIGH
            T1CON.F0 = 1;    //Start Timer
        if(PORTB.F4 == 0)    //If ECHO is LOW
        {
            T1CON.F0 = 0;    //Stop Timer
            a = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance
        }
    }
    INTCON.RBIF = 0;        //Clear PORTB On-Change Interrupt flag
    INTCON.RBIE = 1;        //Enable PORTB On-Change Interrupt
}

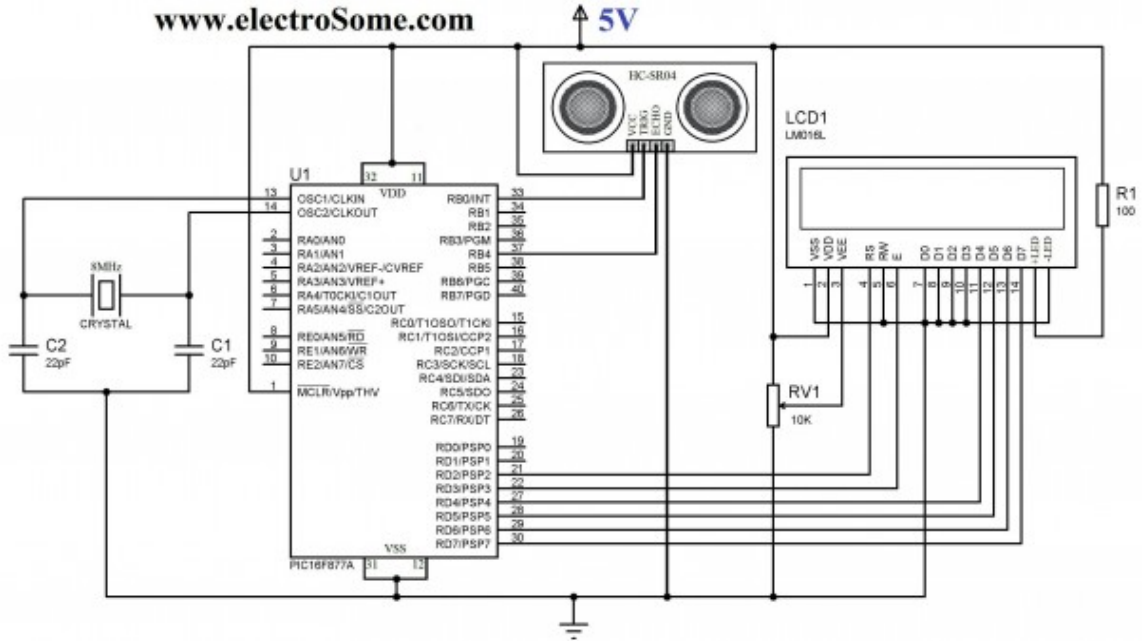
void main()
{
    char txt[7];
    Lcd_Init();
    Lcd_Cmd(_LCD_CLEAR);    // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
    TRISB = 0b00010000;
    INTCON.GIE = 1;        //Global Interrupt Enable
    INTCON.RBIF = 0;        //Clear PORTB On-Change Interrupt Flag
    INTCON.RBIE = 1;        //Enable PORTB On-Change Interrupt
    Lcd_Out(1,1,"SONAR SENSOR");
    Lcd_Out(2,1," DISTANCE ");
    Delay_ms(3000);
    Lcd_Cmd(_LCD_CLEAR);
    T1CON = 0x10;           //Initializing Timer Module
    while(1)
    {
        TMR1H = 0;        //Setting Initial Value of Timer
        TMR1L = 0;        //Setting Initial Value of Timer
        a = 0;
        PORTB.F0 = 1;      //TRIGGER HIGH
        Delay_us(10);      //10uS Delay
        PORTB.F0 = 0;      //TRIGGER LOW
        Delay_ms(100);      //Waiting for ECHO
        a = a + 1;          //Error Correction Constant
        if(a>2 && a<400)    //Check whether the result is valid or not
        {
            IntToStr(a,txt);
            Ltrim(txt);
            Lcd_Cmd(_LCD_CLEAR);
            Lcd_Out(1,1,"Distance = ");
            Lcd_Out(1,12,txt);
        }
    }
}

```

```

    Lcd_Out(1,15,"cm");
}
else
{
    Lcd_Cmd( LCD_CLEAR);
    Lcd_Out(1,1,"Out of Range");
}
Delay_ms(400);
}
}

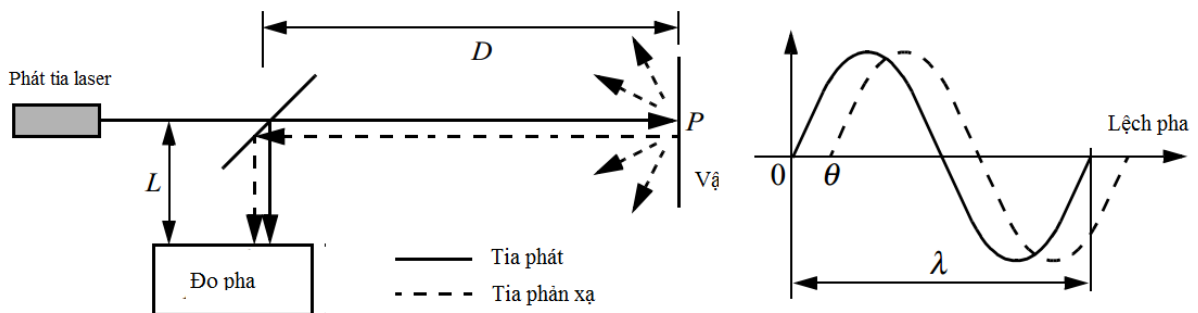
```



Hình 8.15 Giao tiếp cảm biến siêu âm và vi xử lý PIC

8.2.2 Cảm biến laser đo khoảng cách Laser RangeFinder

Khuyết điểm của cảm biến siêu âm là vận tốc truyền âm thay đổi và chùm tia không tập trung, nếu dùng tia laser thì chùm tia sẽ tập trung hơn, tuy nhiên vận tốc ánh sáng rất lớn nên thời gian sóng phản hồi nhỏ cỡ pico giây, đòi hỏi mạch xử lý nhanh, làm tăng giá thành thiết bị. Ta dùng phương pháp điều chế để giải quyết vấn đề này, sóng ánh sáng được điều chế bởi tín hiệu sin có tần số thấp, ví dụ 5MHz, lúc này bước sóng sẽ là $\lambda = c/f = 60m$, sóng phản hồi được giải điều chế và là sóng sin trễ pha so với sóng phát, độ trễ pha θ được đổi sang khoảng cách $D = \frac{\lambda}{4\pi} \theta$



Hình 8.16 Đo khoảng cách bằng phương pháp đo lệch pha tia laser điều chế



Máy đo khoảng cách
đến 20km dùng laser



Máy đo khoảng cách cầm tay

Hình 8.17 Một số máy đo khoảng cách chuyên nghiệp dùng tia laser

Cảm biến laser có giá cao hơn nhiều so với cảm biến siêu âm từ vài trăm đến vài ngàn USD. Thông thường cảm biến laser dùng cho robot di động có trang bị bộ quét tia chung quanh để tăng khả năng đo khoảng cách, ví dụ RPLIDAR 360° có tầm đo từ 0.2m đến 6m, giá 400USD, sai số đo <0.5mm. Độ chính xác của cảm biến laser có thể bị ảnh hưởng bởi ánh sáng mặt trời, độ phản xạ của vật cản và các nguồn phát laser khác.

9



RPLIDAR 360° Laser Scanner



URG-04LX-UG01 Laser Scanner

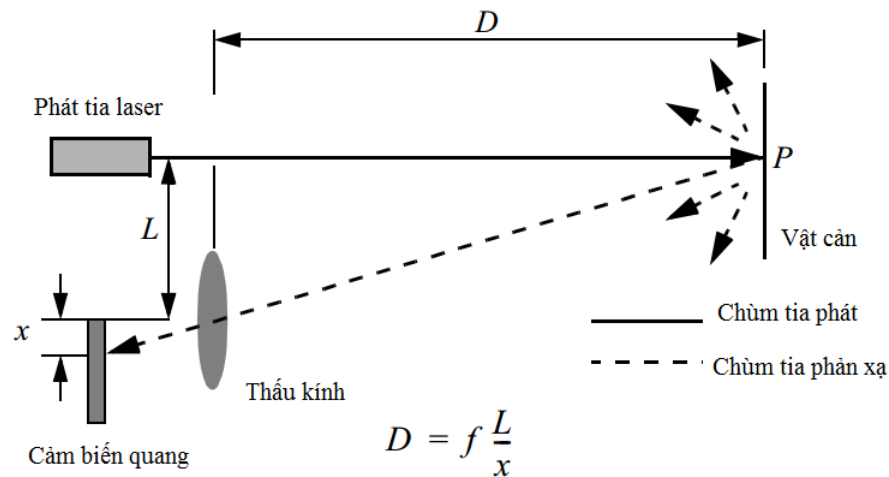


SICK Laser Scanner

Hình 8.18 Một số cảm biến laser quét đo khoảng cách

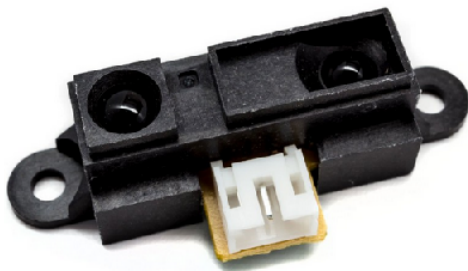
Một số cảm biến laser có giá thấp hơn dùng nguyên tắc tam giác lượng, chùm tia phát được phản xạ và chiếu trên cảm biến quang tuyến tính CCD CMOS đặt tại tiêu điểm f của thấu kính, gọi L khoảng cách từ nguồn phát đến cảm biến, D khoảng cách từ nguồn phát đến vật cản, x tọa độ điểm sáng rơi trên cảm biến, ta có

$$\frac{D}{L} = \frac{f}{x} \rightarrow D = \frac{Lf}{x}$$

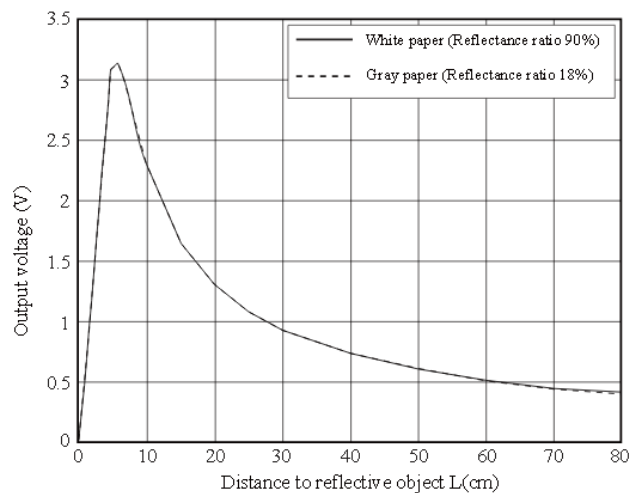


Hình 8.19 Đo khoảng cách dùng phương pháp tam giác lượng

Độ chính xác cảm biến phụ thuộc độ phân giải của cảm biến quang, nếu vật ở gần độ chính xác sẽ cao hơn. Cảm biến hồng ngoại Sharp GP có tầm đo từ 10cm đến 80cm, điện áp ra tỷ lệ nghịch khoảng cách, giá 10USD.

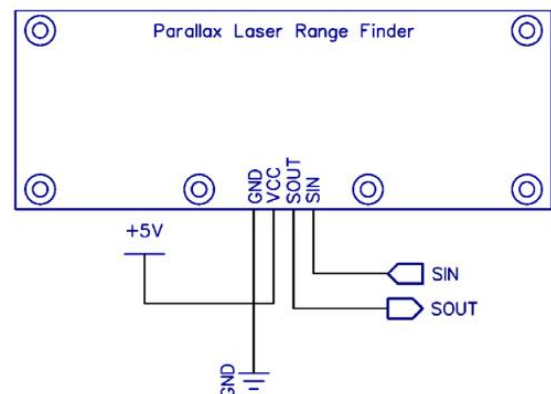


Sharp GP



Hình 8.20 Cảm biến Sharp GP

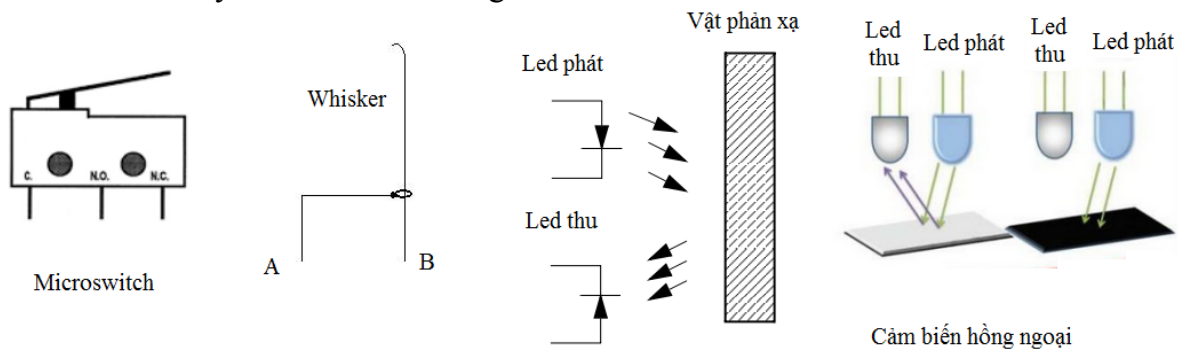
Parallax Laser Range Finder có tầm đo từ 15 đến 122cm, dùng tia laser, giao tiếp qua cổng nối tiếp



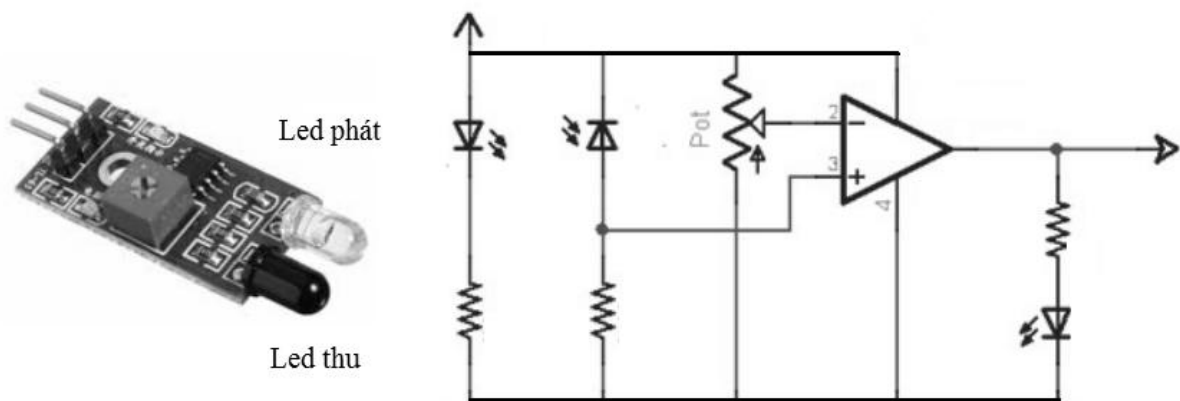
Hình 8.21 Cảm biến Parallax

8.2.3 Cảm biến tiếp cận

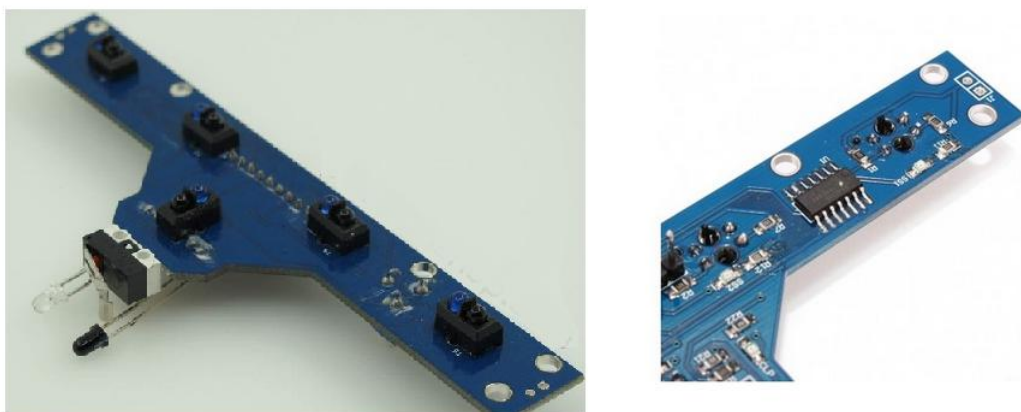
Cảm biến tiếp cận dùng để phát giác một vật nào đó ở gần robot, ngõ ra cảm biến có hai giá trị 0 và 1, loại cảm biến tiếp xúc thường là microswitch, khi đụng vào vật cản thì đóng tiếp điểm, hoặc cấu tạo dạng ria mèo (whisker), khi ria đụng vật cản thì đóng mạch điện, loại cảm biến không tiếp xúc thường dùng cặp đèn Led phát và thu hồng ngoại, khi có vật cản ở gần thì ánh sáng từ đèn phát phản xạ về đèn thu, ta dùng mạch so sánh để chuyển đổi dòng điện qua Led thu sang giá trị nhị phân, cường độ ánh sáng phản xạ cũng thay đổi theo màu của vật cản, do đó loại cảm biến hồng ngoại này thường dùng để phát giác vạch màu dán trên sàn, dùng làm đường dẫn để robot đi theo, thông thường ta dùng nhiều cặp đèn Led thu phát, tổ hợp các trạng thái của Led thu sẽ được xử lý để điều khiển động cơ lái robot.



Hình 8.22 Cảm biến tiếp cận



Hình 8.23 Cảm biến hồng ngoại

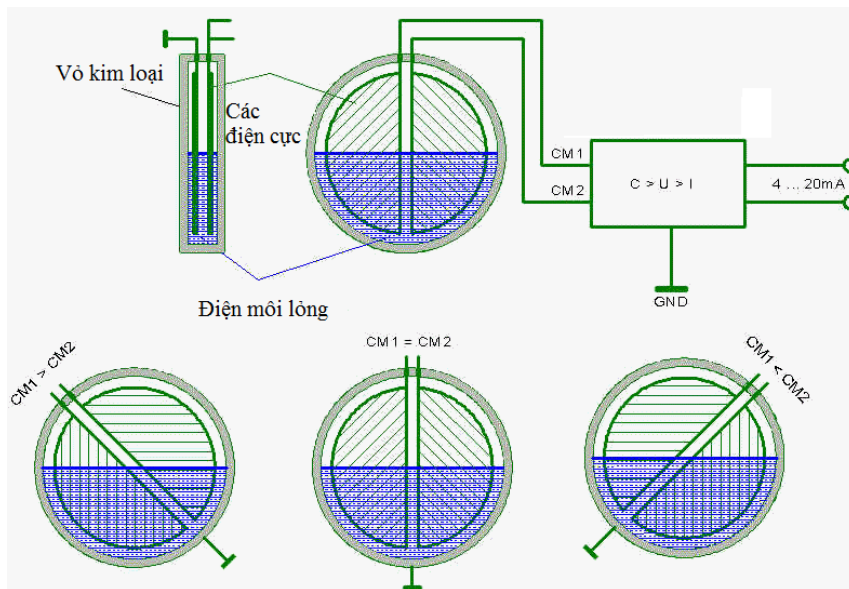


Hình 8.24 Cảm biến đường BFD1000

8.2.4 Cảm biến hướng và vị trí

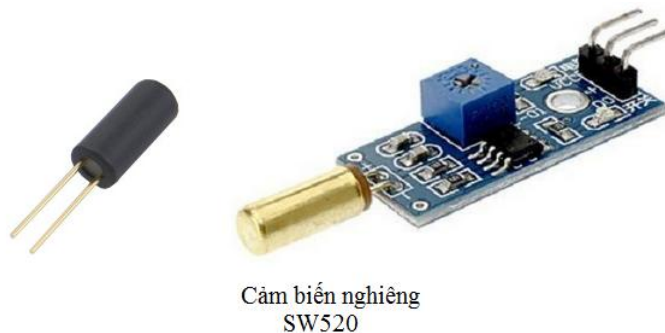
a/ Cảm biến độ nghiêng (Inclinometer)

Cảm biến này đo góc lệch so với một trục, thường là trục đứng, dùng để tạo tín hiệu để đưa vào mạch điều khiển giữ cân bằng robot. Thông thường cảm biến sử dụng phương pháp đo điện dung giữa hai bản cực được đặt trong dung dịch cách điện, có hai cặp bản cực, khi bị nghiêng điện dung cặp này sẽ lớn hơn cặp kia, hiệu số điện dung được chuyển đổi sang điện áp hay số.



Hình 8.25 Cảm biến nghiêng Seika

Một số loại cảm biến nghiêng giá rẻ có đầu ra hai trị (Tilt sensor), bên trong vỏ có viên bi lăn sẽ đóng/mở tiếp điểm khi bị nghiêng



Hình 8.26 Cảm biến nghiêng hai trị

b/ La bàn số

La bàn dùng để đo độ lệch so với cực từ bắc của trái đất, khuyết điểm của la bàn là bị ảnh hưởng của các vật sắt từ ở lân cận. La bàn thường dùng cảm biến Hall cho loại hay Flux gate cho loại chất lượng cao. Độ phân giải của la bàn khoảng 1° đến 2° và giá không đắt, khoảng vài USD với loại chất lượng thấp.



La bàn RB-Ite-71



La bàn HMMC5883L

Hình 8.27 La bàn số

b/ Gyroscope

Gyroscope (Gyros) là thiết bị đo độ lệch so với một phương dùng nguyên tắc con quay. Loại con quay cơ khí sử dụng quán tính của một rotor quay với vận tốc nhanh, duy trì hướng của trục quay nay ít sử dụng, con quay quang học dựa trên sự lệch pha của hai chùm tia laser ngược chiều nhau dùng trong hàng không, quân sự, du hành không gian, con quay áp điện sử dụng trong robot di động hay smartphone đo vận tốc thay đổi của hướng bằng cách đo lực coriolis tác động lên tinh thể áp điện tạo ra điện áp. Nhiều công ty cung cấp con quay áp điện giá rẻ như Futaba GY401, Futaba GY520, HiTec GY 130 sử dụng để ổn định hướng cho các vật bay như trực thăng, quadrotor...



Hình 8.28 Con quay gyroscope

c/ Gia tốc kế (Accelerometer)

Gia tốc kế dùng để đo gia tốc theo một trục, dùng nguyên lý lực tác động lên tinh thể áp điện tỷ lệ gia tốc chuyển động, gia tốc kế có tín hiệu ra kèm theo nhiễu, và trôi theo thời gian do nhiệt độ, cần có bộ lọc nhiễu và bù trôi đi kèm. Thông thường cảm biến đo gia tốc theo ba trục, ví dụ như ADXL335 có thể đo gia tốc đến 3g, ngõ ra điện áp, một số kết hợp thêm la bàn như cảm biến LSM303D.

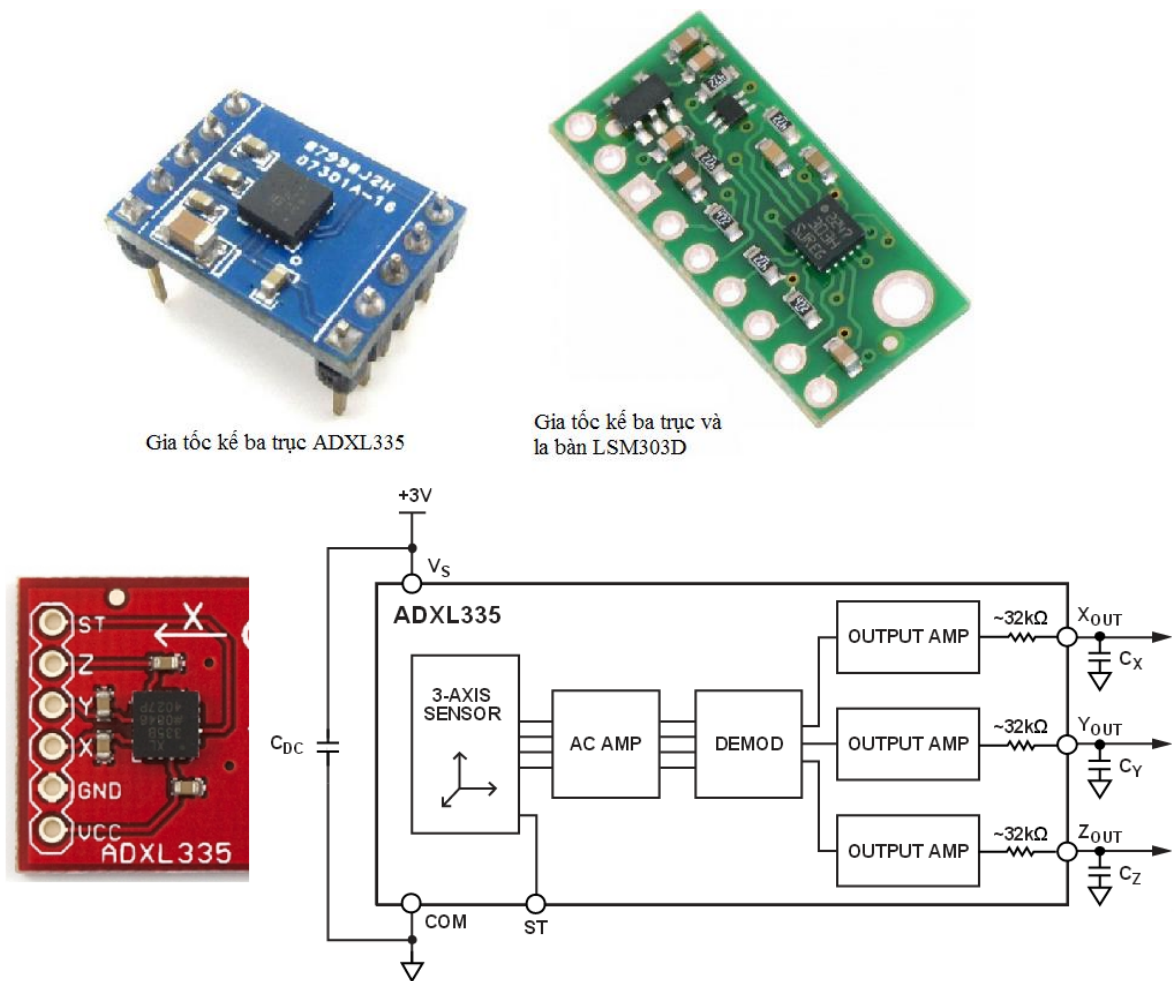
d/ Cảm biến độ cao (Altimeter)

Cảm biến độ cao đo độ cao theo nguyên lý đo áp suất khí quyển theo công thức

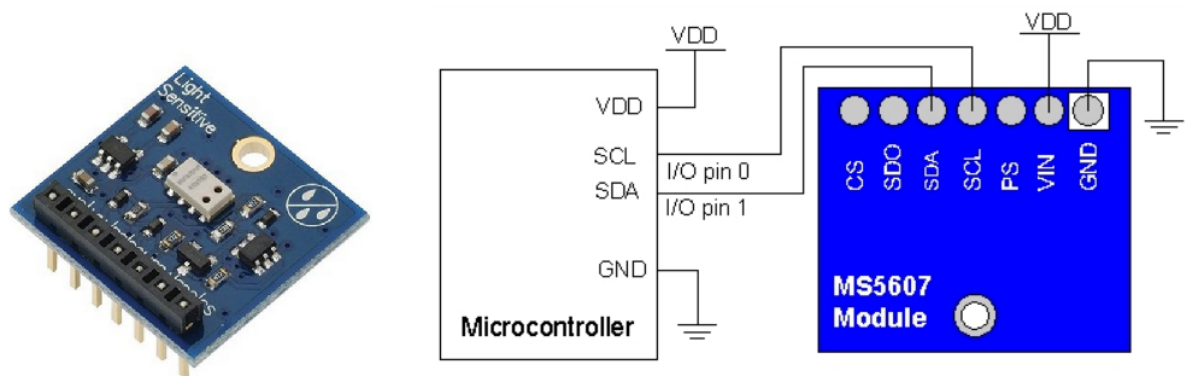
$z = cT \log\left(\frac{P_0}{P}\right)$, c là hằng số, T nhiệt độ tuyệt đối, P_0 áp suất khí quyển mực nước

Robot Di Động

biên, P áp suất khí quyển tại độ cao z . Cách đo này có thể có sai số lớn, các máy bay dùng phương pháp siêu âm hay radar để đo khoảng cách đến mặt đất có độ chính xác cao hơn. Cảm biến Parallax Altimeter Module MS5607 có độ phân giải 20cm, đo độ cao và nhiệt độ, kết quả đo được truyền theo chuẩn I2C đến vi điều khiển.



Hình 8.29 Gia tốc kế



Hình 8.30 Cảm biến độ cao

e/ IMU (Inertial Measurement Unit)

IMU (đơn vị đo quán tính) là cảm biến kết hợp gyro và gia tốc kế, đo gia tốc tuyến tính ba hướng và vận tốc quay theo ba hướng, tổng cộng sáu bậc tự do, từ gia

tốc sẽ tích phân để cho vận tốc và vị trí cũng như góc lệch, đôi khi IMU có cảm biến từ kế đo cường độ từ trường. Giống như gia tốc kế, IMU có độ trôi theo thời gian và cần có thuật toán xử lý như lọc Kalman để bù trừ. MPU6050 là loại cảm biến giá rẻ có thể đo gia tốc đến 16g và vận tốc quay 2000⁰/s, sử dụng ADC 16 bit và truyền dữ liệu nối tiếp chuẩn I2C, SPI, điện áp nguồn là 3V và dòng tiêu thụ 4mA. Công nghệ MEMS (Micro Electro- Mechanical System) giúp giảm kích thước (4x4x0.9mm) và giá thành cảm biến (10USD), trên bo mạch còn có bộ xử lý chuyển động số DMP hỗ trợ xử lý chuyển động ba chiều và nhận dạng cử chỉ.



Hình 8.31 Cảm biến IMU

Tín hiệu ra từ cảm biến gyros và IMU bị nhiễu và trôi theo thời gian, do đó cần dùng bộ lọc để khắc phục, thông thường ta dùng lọc Kalman. Những IMU công nghiệp có giá từ 1000USD trở lên độ chính xác cao

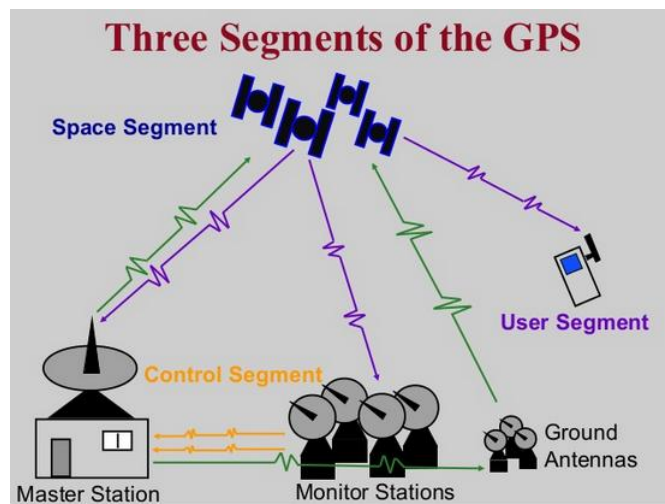
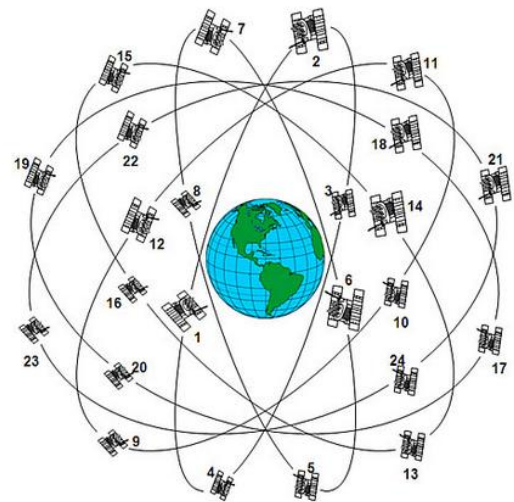
f/ Hệ thống Định vị Toàn cầu GNSS

GNSS (Global Navigation Satellite Service) là hệ thống xác định vị trí dùng vệ tinh, sử dụng phương pháp đo khoảng cách đến 3 hay 4 vệ tinh có vị trí xác định trên quỹ đạo để xác định kinh độ, vĩ độ và độ cao của một vị trí. GPS (Global Positioning System) là hệ thống vệ tinh định vị toàn cầu của Bộ Quốc Phòng Mỹ. Hệ GPS gồm 24 vệ tinh chính và 7 vệ tinh phụ, quay trên quỹ đạo gần tròn độ cao 20.200 km với chu kỳ 2 lần/ngày, mỗi quỹ đạo gồm 4 vệ tinh và có 6 quỹ đạo. Mỗi điểm trên trái đất nhìn thấy hơn 4 vệ tinh. Châu Âu có hệ thống GALILEO, Nga có hệ thống GLONASS (Global Navigation Satellite System), Trung quốc đang triển khai hệ thống Beidou (Bắc đầu).

Hệ thống GPS gồm ba đoạn: đoạn không gian là các vệ tinh, đoạn điều khiển là các trạm mặt đất và đoạn người dùng là các máy thu GPS. Các vệ tinh được giám sát bởi các trạm mặt đất đặt rải rác khắp thế giới gồm có 12 trạm anten mặt đất, 16 trạm quan sát/điều khiển và hai trạm điều khiển chính, Trạm anten nhận tín hiệu vệ tinh gửi đến trạm điều khiển và gửi lệnh lên vệ tinh, trạm quan sát thu thập số liệu khí quyển, tín hiệu từ vệ tinh và gửi về trạm điều khiển, trạm điều khiển nhận thông tin vệ tinh từ

Robot Di Động

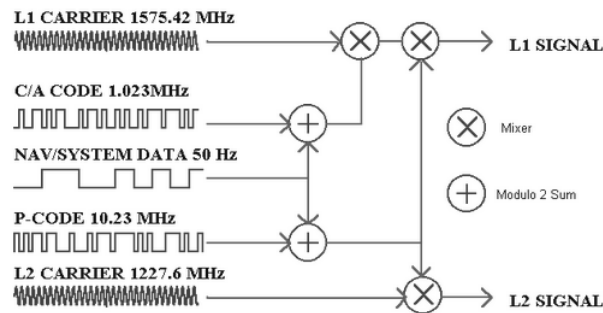
trạm quan sát, tính toán quỹ đạo chính xác của vệ tinh, sức khỏe của vệ tinh và gửi dữ liệu lên vệ tinh, điều chỉnh vị trí vệ tinh khi cần thiết.



Hình 8.32 Hệ thống GPS

Các vệ tinh gửi tín hiệu xuống trái đất có mang thông tin mã hoá và vị trí vệ tinh, máy thu GPS nhận tín hiệu từ bốn vệ tinh, so sánh với thời điểm nhận, từ đó tính khoảng cách đến các vệ tinh, vị trí máy thu là giao điểm bốn hình cầu có tâm là bốn vệ tinh và bán kính là các khoảng cách, từ đó tính ra tọa độ ba chiều của điểm thu. Các đồng hồ trên vệ tinh là đồng hồ nguyên tử đồng bộ với nhau, sai số 1s trong 300 triệu năm, một trung tâm quản lý mặt đất giám sát hoạt động các vệ tinh này. Vệ tinh phát tín hiệu số ở tần số sóng mang L1 (1,575.42 MHz) và L2 (1,227.60 MHz). Bản tin từ vệ tinh gồm thông tin khoảng cách (Ranging code) là một đoạn mã PRN (Pseudo Random Noise) và thông tin về vệ tinh (Navigation data). Thông tin khoảng cách được mã hoá dưới hai dạng C/A (Coarse/Acquisition) dùng cho dân sự và P (Precise) dùng cho quân sự. Thông tin C/A có vận tốc 1,023Mbit/s gồm 1023 bit trong thời gian 1ms và lặp lại, khác nhau cho mỗi vệ tinh. Thông tin vệ tinh (navigation message) bao gồm thời điểm truyền đoạn mã trên vệ tinh và vị trí của vệ tinh, truyền đi với vận tốc 50 bit/s và có 25 frame, mỗi frame dài 1500 bit và bản tin truyền đi hết 12.5 phút. Hai thông tin này được exclusive or (cộng modulo 2) với nhau

và điều chế sóng mang L1. Thông tin P có vận tốc 10,23Mbit/s và chiều dài là 6.1871×10^{12} được cộng modulo 2 với thông tin vệ tinh và điều chế hai sóng mang L1 L2.



Hình 8.33 Tín hiệu GPS

Máy thu GPS có đồng hồ đồng bộ với đồng hồ vệ tinh, so mã PRN của nó với mã PRN nhận được, khi có trùng khớp thì nó xác định được độ lệch thời gian từ đó suy ra khoảng cách đến vệ tinh, 1 bit mã C/A là $1\mu s$ tương ứng với khoảng đường đi là 300m, nếu sai lệch bit là $1/100 \mu s$ thì sai lệch khoảng cách là 3m

Tuỳ theo máy thu GPS, sai số định vị từ vài mét đến vài chục mét ở ngoài trời. Hệ thống GPS có sai số do các yếu tố sau:

- Đồng hồ máy thu GPS không chính xác bằng đồng hồ nguyên tử
- Vận tốc truyền tín hiệu vô tuyến thay đổi do không khí và các tầng khí quyển
- Truyền tín hiệu từ vệ tinh đến máy thu không theo đường thẳng do khúc xạ trong khí quyển và phản xạ từ các vật trên mặt đất.
- Vị trí vệ tinh lúc thu, nếu các vệ tinh tản ra cách xa nhau thì chính xác hơn.
- Cường độ tín hiệu thu yếu, nhất là trong nhà, máy thu bị che chắn

Độ chính xác định vị được cải thiện nhờ các cải tiến sau:

- DGPS Differential GPS, dùng tín hiệu từ vệ tinh và tín hiệu từ trạm cố định để sửa sai vị trí.
- Thu hai sóng mang để khử sai số do khí quyển.
- Assisted-GPS (A-GPS) thường được dùng cho các thiết bị cầm tay, Smart Phone. Ngoài việc định vị GPS, smartphone sử dụng A-GPS còn kết nối với máy chủ thông qua mạng 3G, Wi-Fi để nhận tín hiệu phát ra từ các trạm phát sóng của nhà mạng. Nhờ thế mà thiết bị khắc phục được sai số từ tín hiệu vệ tinh khi truyền xuống vùng đô thị có nhiều tán cây, cao ốc.
- GPS-RTK (Real Time Kinematic) Hệ thống GPS động học thời gian thực.

Các máy thu GPS dùng GPS Chipset, có loại dùng được cho nhiều hệ thống GNSS, dữ liệu tính toán được truyền nối tiếp theo chuẩn NMEA-0183 ví dụ như \$GPGGA,111636.932, 2447.0949, N, 12100.5223, E, 1,11,0.8,118.2, M,,, 0000*02 <CR> <LF>

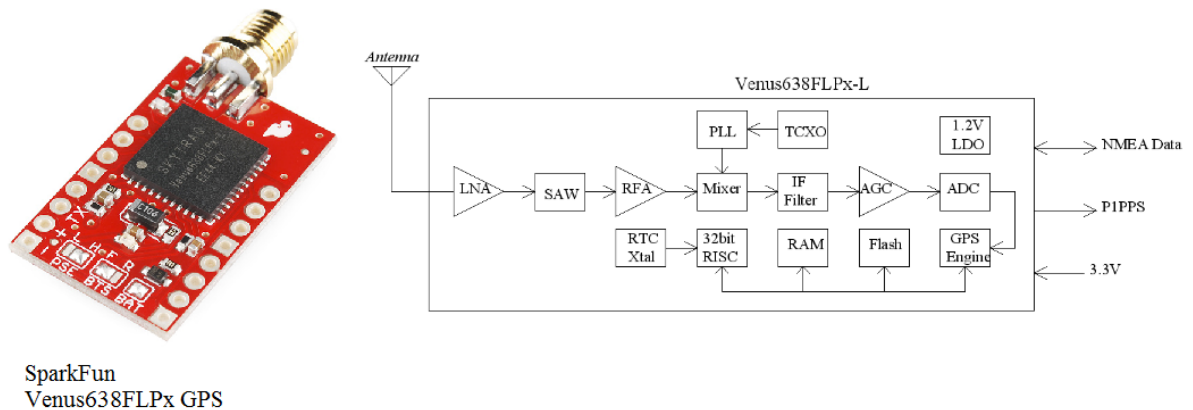
Giao thức trên có nghĩa như sau:

\$GPGGA nghĩa là định vị dùng GPS bản tin GGA, sau đó truyền giờ UTC, vĩ độ, kinh độ, chất lượng tín hiệu 1..8, số vệ tinh dùng, 1..12, hệ số HDOP 00.0 ..99.9, cao độ tính bằng mét, ID của trạm DGPS, checksum *xx, cuối cùng là mã ASCII <CR> <LF>

Robot Di Động

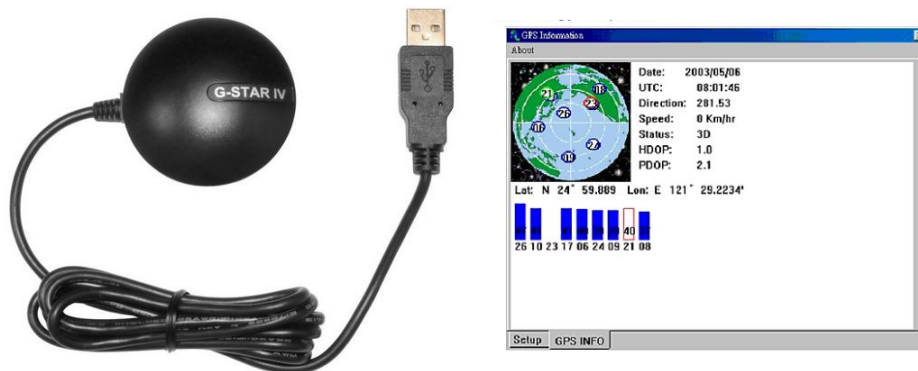
Giờ được tính theo giờ quốc tế UTC Universal Coordinated Time đi qua kinh tuyến Greenwich có dạng hhmmss.sss, 111636.932 có nghĩa giờ quốc tế là 11 giờ 16 phút 36 giây 932 phần ngàn giây, giờ VN là UTC+7.

Vĩ độ (latitude) kinh độ (longitude) có dạng dddmm.mmmm, N/S chỉ bán cầu Bắc Nam, E/W chỉ phía Đông/ Tây kinh tuyến Greenwich, ví dụ 2447.0949, N, 12100.5223, E có nghĩa vĩ độ Bắc 24°47,0949 phút, kinh độ Đông 121° 0,5223 phút, một độ là 60 phút.



SparkFun
Venus638FLPx GPS

Hình 8.34 Chip thu GPS



BU-353-S4 GPS Receiver

Hình 8.35 Thiết bị thu GPS dùng với máy tính



Thiết bị thu GPS cầm tay



Ứng dụng GPS trên smartphone

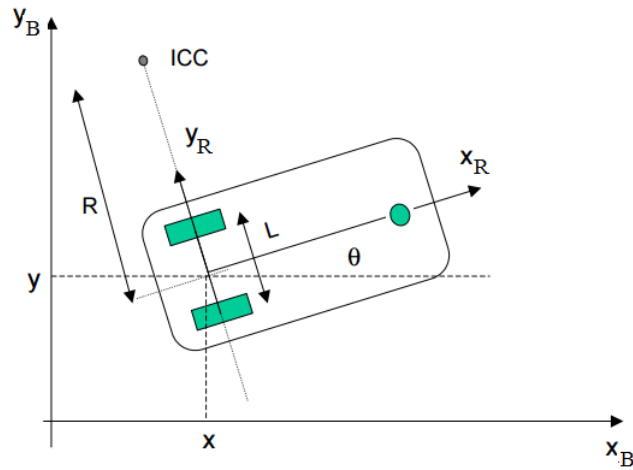
Hình 8.36 Thiết bị thu GPS cầm tay và Ứng dụng GPS trên điện thoại

8.3 ĐỘNG HỌC VÀ ĐỘNG LỰC HỌC

Động học robot khảo sát vị trí và hướng robot theo góc quay các động cơ điều khiển, tùy theo loại robot phương trình động học sẽ khác nhau, trong mục này chúng ta chỉ khảo sát động học một số loại robot đơn giản.. Động lực học robot di động biểu diễn quan hệ giữa điện áp động cơ và vị trí robot, có liên quan đến lực, moment, khối lượng và moment quán tính hệ nhiều vật.

8.3.1 Động học xe ba bánh

Robot ba bánh rất thông dụng vì điều khiển đơn giản với hai bánh sau điều khiển độc lập và bánh trước tự do



Hình 8.37

Gọi ω_w vận tốc quay bánh xe, r bán kính, vận tốc di chuyển của bánh xe là $v = \omega_w r$, Nếu hai bánh xe quay cùng chiều cùng vận tốc, xe di chuyển theo đường thẳng, nếu hai bánh xe quay không cùng vận tốc, xe di chuyển theo đường cong có bán kính cong là R theo trục giữa của xe, gọi v vận tốc dài của xe, ω vận tốc góc của xe, ta có

$$\omega = \frac{v_r}{R + \frac{L}{2}} = \frac{v_l}{R - \frac{L}{2}},$$

$$\omega = \frac{v_r - v_l}{L}, R = \frac{L}{2} \frac{v_l + v_r}{v_l - v_r},$$

$$v = \omega R = \frac{1}{2}(v_l + v_r)$$

Vận tốc xe theo hai trục và vận tốc góc là

$$\dot{x} = v_x = v \cos \theta,$$

$$\dot{y} = v_y = v \sin \theta,$$

$$\dot{\theta} = \omega$$

Biểu thị theo vận tốc góc bánh xe

$$\dot{x} = \frac{r}{2}(\omega_{wl} + \omega_{wr}) \cos \theta,$$

$$\begin{aligned}\dot{y} &= \frac{r}{2}(\omega_{wl} + \omega_{wr})\sin\theta, \\ \dot{\theta} &= \frac{r}{L}(\omega_{wr} - \omega_{wl}) \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} &= \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_{wl} \\ \omega_{wr} \end{bmatrix}\end{aligned}$$

Tích phân vận tốc ta được vị trí của xe

$$\begin{aligned}\theta(t) &= \theta_o + \frac{r}{L} \int_0^t (\omega_{wr}(\tau) - \omega_{wl}(\tau)) d\tau, \\ x(t) &= x_o + \frac{r}{2} \int_0^t (\omega_{wr}(\tau) + \omega_{wl}(\tau)) \cos\theta(\tau) d\tau, \\ y(t) &= y_o + \frac{r}{2} \int_0^t (\omega_{wr}(\tau) + \omega_{wl}(\tau)) \sin\theta(\tau) d\tau,\end{aligned}$$

Thay tích phân bằng phép cộng, ta đếm số xung từ encoder trong khoảng thời gian T_s , quãng đường di chuyển của mỗi bánh là $d_k = (\text{số xung}(k) - \text{số xung}(k-1)) * 2\pi * r / N$, N là số xung encoder/vòng

$$\begin{aligned}\theta_k &= \theta_{k-1} + \frac{d_{kr} - d_{kl}}{L}, \\ x_k &= x_{k-1} + \frac{d_{kr} + d_{kl}}{2} \cos\theta_k, \\ y_k &= y_{k-1} + \frac{d_{kr} + d_{kl}}{2} \sin\theta_k,\end{aligned}$$

Do sai số của tích phân số, độ trượt trên mặt đường và lỗi lồm của mặt đường nên vị trí và hướng xe thực tế khác với biểu thức trên và sai số sẽ tích lũy dần.

Phương trình động lực học của xe suy từ phương trình Lagrange

$$L = K = \frac{1}{2}M(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}J_l\omega_{wl}^2 + \frac{1}{2}J_r\omega_{wr}^2,$$

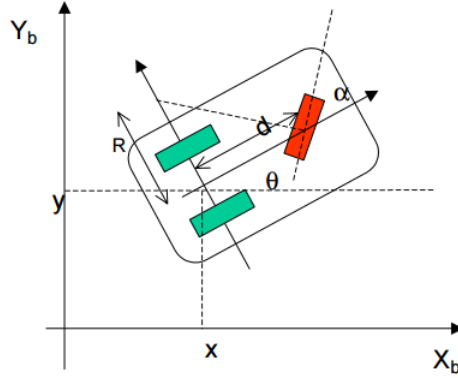
M : khối lượng, I : moment quán tính xe, J_l , J_r : moment quán tính rotor động cơ và bánh xe.

Suy ra phương trình động lực học

$$D \begin{bmatrix} \dot{\omega}_{wr} \\ \dot{\omega}_{wl} \end{bmatrix} + F(\omega_{wr}, \omega_{wl}) = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} = \tau$$

D : ma trận quán tính robot, τ : moment quay hai bánh xe, $F(\omega_{wr}, \omega_{wl})$: các ma sát. Nhìn chung khó tính được biểu thức chính xác của phương trình.

Trường hợp bánh xe trước dẫn động còn hai bánh xe sau tự do, quỹ đạo của xe xác định bởi góc lái α và vận tốc ω_w bánh xe, gọi d khoảng cách trục bánh lái và trục bánh sau, ω vận tốc góc xe, R bán kính cong, r bán kính bánh lái



Hình 8.38

$$R = d \tan\left(\frac{\pi}{2} - \alpha\right),$$

$$\omega = \frac{\omega_w r}{\sqrt{d^2 + R^2}} = \frac{\omega_w r \sin \alpha}{d},$$

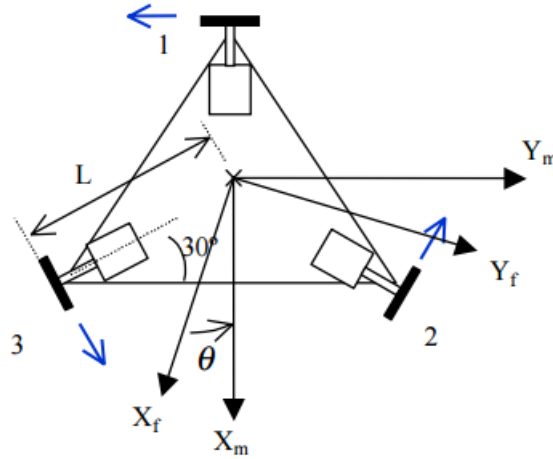
$$\dot{x} = r\omega_w \cos \alpha \cos \theta,$$

$$\dot{y} = r\omega_w \cos \alpha \sin \theta,$$

$$\dot{\theta} = \omega$$

Nếu xe ba bánh dùng bánh xe đa hướng thì ta có thể di chuyển xe theo mọi hướng và quay xe tại chỗ bằng cách điều khiển vận tốc ba bánh, giả sử các bánh xe bố trí theo hình tam giác đều

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{r}{\sqrt{3}} & \frac{r}{\sqrt{3}} \\ -\frac{2r}{3} & \frac{r}{3} & \frac{r}{3} \\ \frac{r}{3L} & \frac{r}{3L} & \frac{r}{3L} \end{bmatrix} \begin{bmatrix} \omega_{w1} \\ \omega_{w2} \\ \omega_{w3} \end{bmatrix}$$

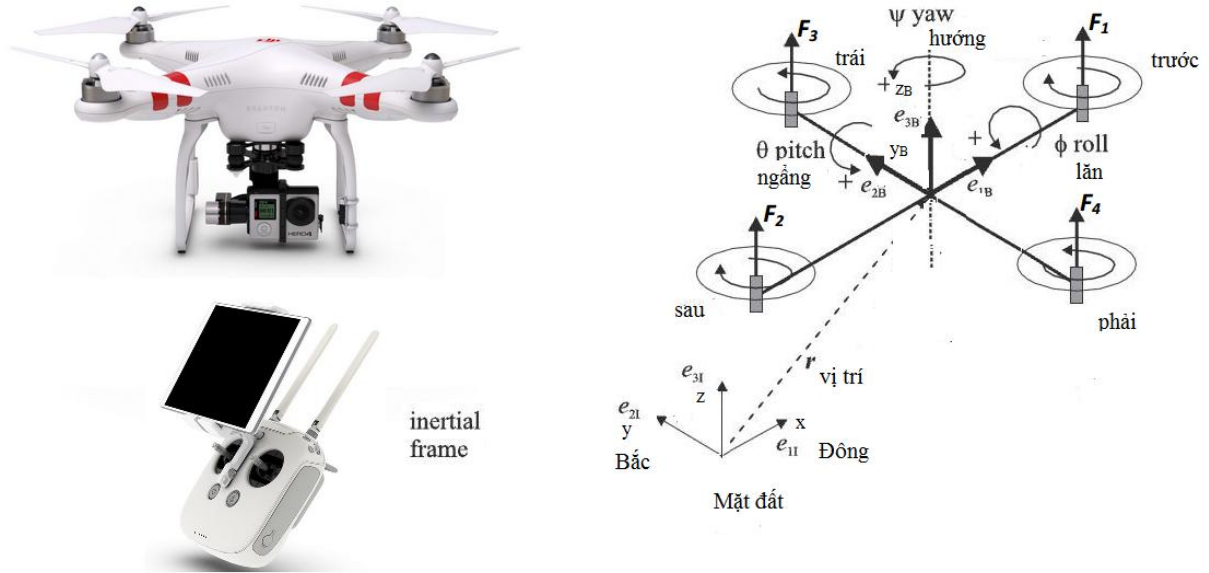


Hình 8.39

8.3.2 Động học QuadRotor

Quad Rotor là thiết bị bay có bốn chong chóng, nhỏ gọn thường dùng để quay phim từ trên cao, có bốn cánh quạt điều khiển bởi bốn động cơ, có khả năng cất cánh và hạ cánh thẳng đứng, lơ lửng trên không, thay đổi vị trí và hướng trong không gian bằng cách điều khiển chiều và vận tốc động cơ, Bốn động cơ chia thành hai cặp trước sau và trái phải, khi quay cánh quạt tạo lực nâng và moment xoay quanh trục, do đó cặp trước sau và trái phải phải quay ngược chiều để tránh quadrotor tự quay quanh trục của nó. Vị trí của quadrotor là vector $[x; y; z]$ tọa độ trọng tâm của nó còn hướng xác định bởi vector Θ gồm ba góc Euler ϕ (roll góc lặn) θ (pitch góc ngả) ψ (yaw góc hướng) là góc quay lần lượt qua ba trục x, y, z của hệ trục tham chiếu cố định.

Muốn thay đổi độ cao ta tăng hay giảm cùng một lượng cho vận tốc bốn động cơ, muốn xoay quanh trục đứng (góc hướng, yaw) ta tăng/giảm vận tốc cặp trước sau và giảm /tăng vận tốc cặp trái phải, thay đổi góc ngả (pitch) ta tăng/giảm vận tốc động cơ trước và giảm/ tăng vận tốc động cơ sau, thay đổi góc lặn (roll) ta tăng/giảm vận tốc động cơ trái và giảm/ tăng vận tốc động cơ phải.



Hình 8.40

Sử dụng hệ tham chiếu cố định (x y z) và hệ trục qua trọng tâm gắn với robot ($x_B y_B z_B$). Ma trận quay hệ toạ độ robot r so với hệ tham chiếu f là (xem 2.15)

$$R_f^r = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

Gọi $\mathbf{v}^r = [u \ v \ \omega]^T$, $\boldsymbol{\omega}^r = [p \ q \ r]^T$ vận tốc dài và vận tốc góc trong hệ toạ độ robot, $\mathbf{v}^f = [\dot{x} \ \dot{y} \ \dot{z}]^T$, $\dot{\boldsymbol{\Theta}}^f = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ là các đạo hàm vị trí và góc Euler trong hệ toạ độ tham chiếu

$$\mathbf{v}^f = R_f^r \mathbf{v}^r$$

$$\dot{\boldsymbol{\Theta}}^f = \mathbf{T} \boldsymbol{\omega}^r, \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}$$

Lực nâng của cánh quạt tỷ lệ với bình phương vận tốc quay của cánh, gọi $\omega_1, \omega_2, \omega_3, \omega_4$ lần lượt là vận tốc quay của cánh quạt, lực nâng là $f_i = b\omega_i^2$, b là hệ số. Lực nâng là $u_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$, các moment gây ra chuyển động roll, pitch, yaw lần lượt là

$$u_2 = bl(\omega_3^2 - \omega_4^2), u_3 = bl(\omega_2^2 - \omega_1^2), u_4 = d(\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2)$$

l là khoảng cách từ trọng tâm robot đến trục cánh quạt, d là hệ số moment quay

Ta công nhận không chứng minh phương trình động lực học quad rotor (xem Modelling, Identification and Control of a Quadrotor Helicopter, Tommaso Bresciani, 2008)

$$\ddot{x} = (s\psi s\phi + c\psi s\theta c\phi) \frac{u_1}{m},$$

$$\ddot{y} = (-c\psi s\phi + s\psi s\theta c\phi) \frac{u_1}{m},$$

$$\ddot{z} = (-g + c\theta c\phi) \frac{u_1}{m},$$

$$\dot{p} = \frac{I_y - I_z}{I_x} qr - \frac{J_{TP}}{I_x} q\omega + \frac{u_2}{I_x},$$

$$\dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{J_{TP}}{I_y} p\omega + \frac{u_3}{I_y},$$

$$\dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{u_4}{I_z},$$

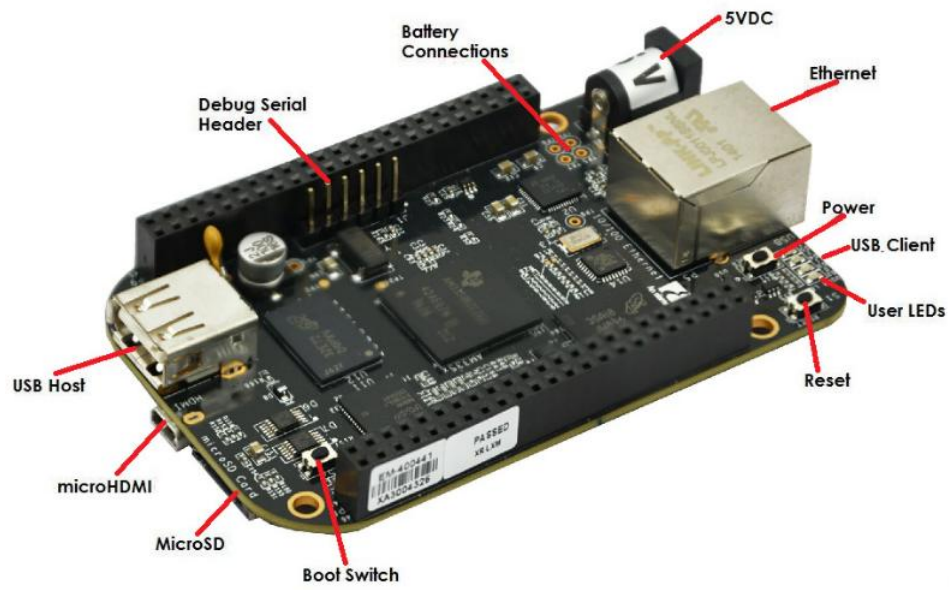
m khối lượng robot, I_x, I_y, I_z các moment quán tính robot theo ba trục, $\omega = \omega_3 + \omega_4 - \omega_1 - \omega_2$, J_{TP} là moment quán tính động cơ.

8.4 ĐIỀU KHIỂN

Bộ điều khiển cho robot di động thường dùng vi điều khiển được trang bị ngoại vi có khả năng giao tiếp truyền thông nối tiếp như USB, UART, Ethernet, WIFI, có khả năng phát xung điều rộng PWM điều khiển động cơ. Các loại vi điều khiển thông dụng thuộc họ ARM, PIC, AVR.

Công nghệ vi mạch đã phát triển các loại máy tính nhỏ có tính năng mạnh chạy trên hệ điều hành Android, Linux rất tiện sử dụng cho robot di động. ví dụ như Beaglebone Black là một bo mạch kích thước bằng thẻ tín dụng, vi xử lý ARM CORTEX A8 1GHz, RAM 512MB, Flash ROM 4GB, micro HDMI port, thẻ nhớ ngoài, USB, Ethernet. Một loại máy tính khác là Raspberry Pi3, CPU ARM Cortex-A53 Quadcore 1.2GHz 64-bit, RAM 1GB, USB, Ethernet, Wifi 802.11n, Bluetooth 4.1.

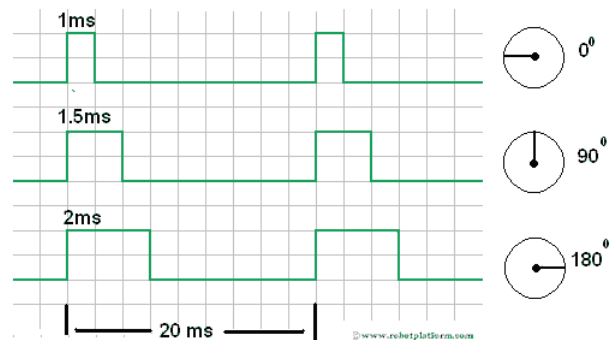
Động cơ dùng cho robot di động để tạo chuyển động thường là động cơ PMDC, BLDC và động cơ RC servo. Động cơ RC (Radio Control) servo thường dùng trong các hệ điều khiển từ xa dùng sóng vô tuyến, là động cơ một chiều ghép với bánh răng giảm tốc kéo một chiết áp, điện áp trên chiết áp tỷ lệ góc quay động cơ, dùng làm tín hiệu hồi tiếp cho mạch kín điều khiển vị trí, tín hiệu đặt vị trí gửi từ bộ điều khiển từ xa tới dưới dạng xung điều rộng có chu kỳ 20ms, được chuyển thành điện áp đặt cho bộ điều khiển và làm quay động cơ, bề rộng xung thường là từ 1ms đến 2ms.

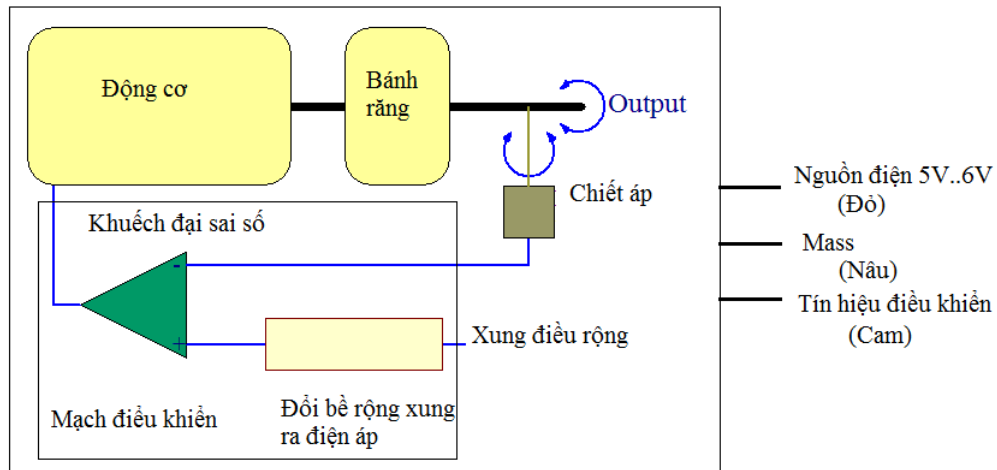


Hình 8.41 BeagleBone Black computer



Hình 8.42 Raspberry Pi3 computer





Hình 8.43 Động cơ RC servo

Ví dụ: Chương trình con dùng PIC PC.1 điều khiển RC servo

```
#include <16F877A.h>
#fuses NOWDT,PUT,XT,NOPROTECT
#use delay(clock=16000000)
long dem;
int value;//bề rộng xung,0..200
#INT_TIMER0
void RC_servo()
{
    output_high(PIN_C0);
    dem++;          //biến đếm tăng thêm 1 mỗi 10us
    if ((dem>=value)|| (dem>=200))output_low(PIN_C0); //tắt cổng
    if (dem >= 2000) //xung 50Hz
    {
        dem=0;    // reset biến đếm
        output_high(PIN_C0);//bật cổng
    }
}
```



```

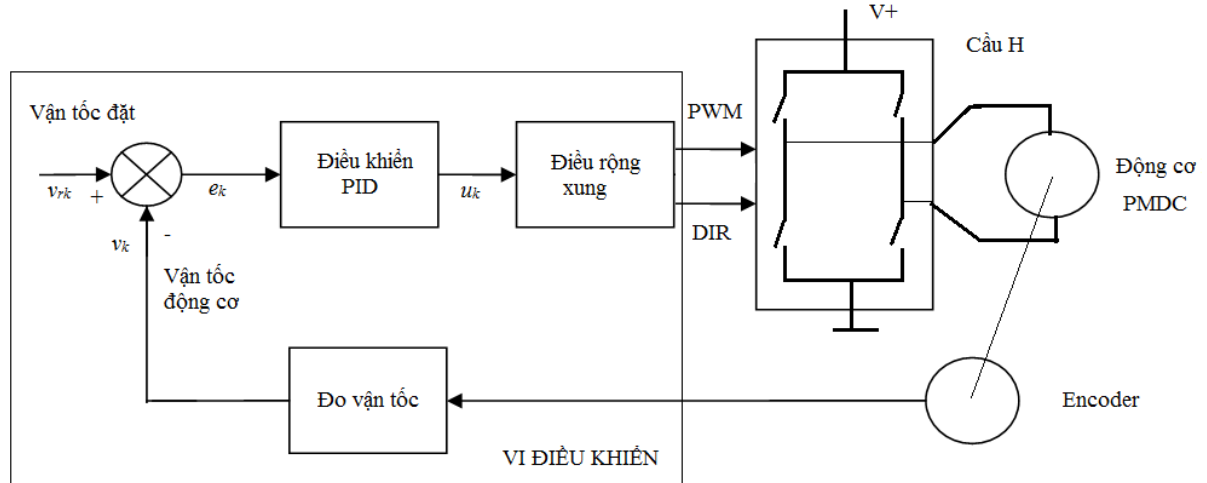
    }
void main()
{
    set_tris_c(0x00); //set các chân port C là chân xuất
    output_c(0);
    enable_interrupts(int_timer0); //cho phép ngắt timer0
    enable_interrupts(global);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4); //chọn tần số xung nhịp
    // cho timer0
    set_timer0(245); //cài timer0 tràn mỗi 10us
    while(true){ value=100;}
}

```

Gần đây đã có loại RC servo số, tần số xung gửi đến động cơ là 300Hz thay vì 50Hz, động cơ sử dụng là loại PMDC, một số sử dụng động cơ BLDC.

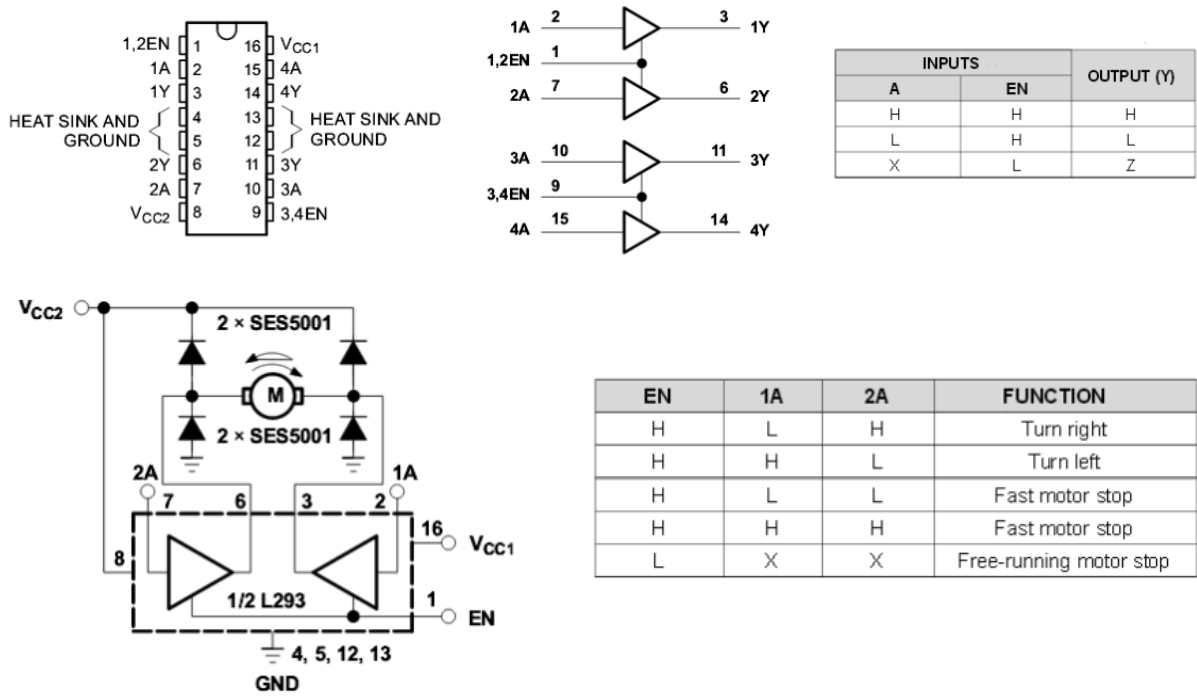
Một loại RC servo khác điều khiển vận tốc động cơ thay vì vị trí, như vậy động cơ sẽ quay liên tục với vận tốc phụ thuộc độ rộng xung.

Khi không dùng động cơ RC Servo ta phải thiết kế mạch vòng kín điều khiển động cơ, thông thường dùng điều khiển PID thực hiện nhờ vi điều khiển, tín hiệu ra PID đưa vào mạch công suất dùng cầu H điều rộng xung để điều khiển động cơ, tùy theo tín hiệu hồi tiếp ta có thể điều khiển vị trí hay vận tốc động cơ.



Hình 8.44 Điều khiển vận tốc động cơ PMDC

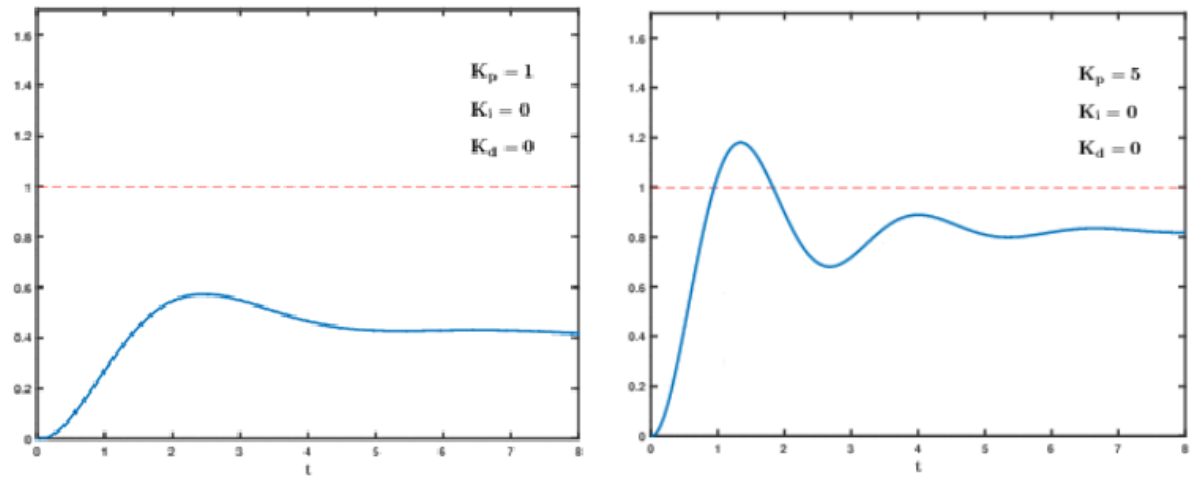
Mạch cầu H thường dùng loại IC công suất, ví dụ L293 có khả năng tải dòng đến 1A với điện áp nguồn 4.5V..36V, điều khiển được hai động cơ.

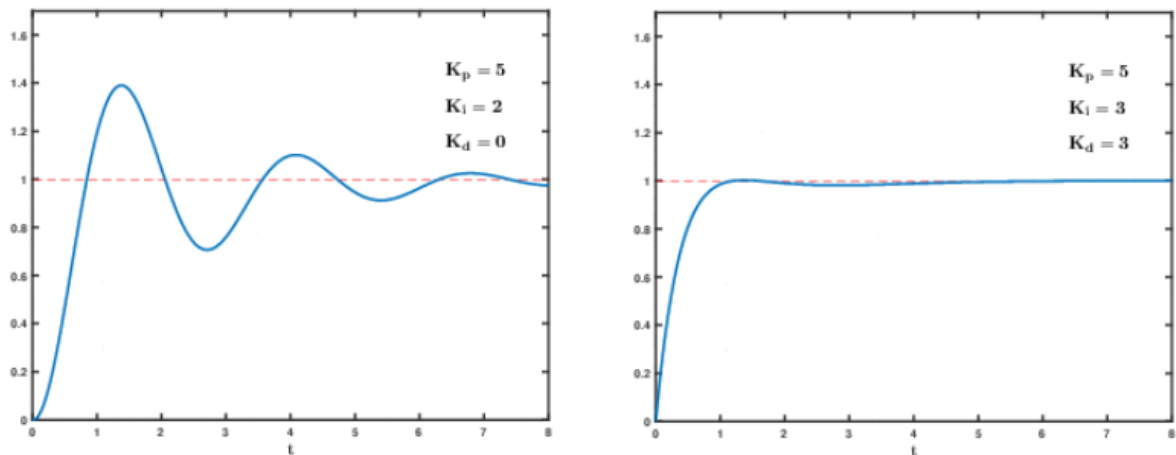


Hình 8.45 Cầu H điều khiển động cơ

Thuật toán điều khiển PID đã được khảo sát trong môn học Cơ sở điều khiển tự động, rất thông dụng vì đơn giản và hiệu quả, triệt tiêu sai số xác lập với tín hiệu đặt và với nhiễu, tuy nhiên việc tính chọn thông số PID để đạt chất lượng tốt không phải là đơn giản vì ta không biết được chính xác mô hình toán của động cơ và xe robot.

Hình 8. 44





Hình 8.46 Ảnh hưởng thông số lên chất lượng điều khiển

Thuật toán PID số được thực hiện theo chu kỳ lấy mẫu T_s thường là vài chục ms theo công thức sau, gọi u_k , e_k lần lượt là tín hiệu ra PID và sai lệch giữa vận tốc đặt v_{rk} và vận tốc hồi tiếp v_k ở bước thứ k :

$$u_k = K_P e_k + \frac{T_s}{T_I} \frac{e_k + e_{k-1}}{2} + \frac{T_D}{T_s} (e_k - e_{k-1})$$

$$u_k = K_P e_k + K_I \frac{e_k + e_{k-1}}{2} + K_D (e_k - e_{k-1})$$

$$u_k = u_{k-1} + K_P (e_k - e_{k-1}) + K_I \frac{e_k + e_{k-1}}{2} + K_D (e_k - 2e_{k-1} + e_{k-2})$$

$$K_I = \frac{T_s}{T_I}, K_D = \frac{T_D}{T_s}$$

Các bước tìm thông số thực nghiệm được thực hiện như sau:

- Cho K_I , K_D bằng 0, chạy động cơ ở chế độ định mức,
- Tăng dần K_P cho đến khi có dao động, ghi nhận giá trị này
- Chỉnh K_P bằng phân nửa giá trị trên,
- Tăng K_D , quan sát đáp ứng khi cho vận tốc đặt thay đổi 5% , chọn giá trị có đáp ứng tốt nhất (không vọt lố, đáp ứng nhanh)
- Tăng dần K_I cho đến khi bắt đầu dao động, chọn K_I là $\frac{1}{2}$ hay $\frac{1}{3}$ giá trị này.

Đoạn code cho PID có thể viết như sau:

```
static int r_old=0, e_old=0, e_old2=0;
...
e = v_des - v_act;
r_mot = r_old + Kp*(e-e_old) + Ki*(e+e_old)/2 + Kd*(e - 2*e_old + e_old2);
r_mot = min(r_mot, rsat); /* limit output */
r_mot = max(r_mot, -rsat); /* limit output */
```

```
r_old = r_mot;
e_old2 = e_old;
e_old = e;
```

Đoạn code sau viết cho PIC 16F877 điều khiển vận tốc hai chiều, encoder 200xung/vòng, tần số lấy mẫu 100Hz, chu kỳ PWM 1ms, xung đếm RB0, chiều quay RB1, điều khiển chiều quay dùng RC6, RC7, tín hiệu PWM là RC2, để nhận biết chiều quay động cơ, xung A của encoder đưa vào RB0 gây ngắt cạnh xuống, còn xung B đưa vào RB1, nếu khi ngắt mà RB1 là 1 thì quay thuận tăng số xung đếm Pulse lên còn ngược lại ta giảm Pulse, vận tốc là $(Pulse_k - Pulse_{k-1})/T_s$.

PIC 16F877 có hai chân ra điều rộng xung ở RC1/CCP2 và RC2/CCP1, chu kỳ xung xác định bởi Timer 2, độ rộng xung có độ phân giải 10 bit, từ 1 đến 1023

- Cài đặt chu kỳ phát xung

```
setup_timer_2(PreScaler, PR2, PostScaler);
```

PreScaler: hệ số 1,4,16; T2_DIV_BY_1, T2_DIV_BY_4, T2_DIV_BY_16

Chu kỳ $= (PR2+1) \cdot 4 \cdot T_{OSC} \cdot PreScaler$.

PostScaler: số lần Timer2 tràn thì ngắt.

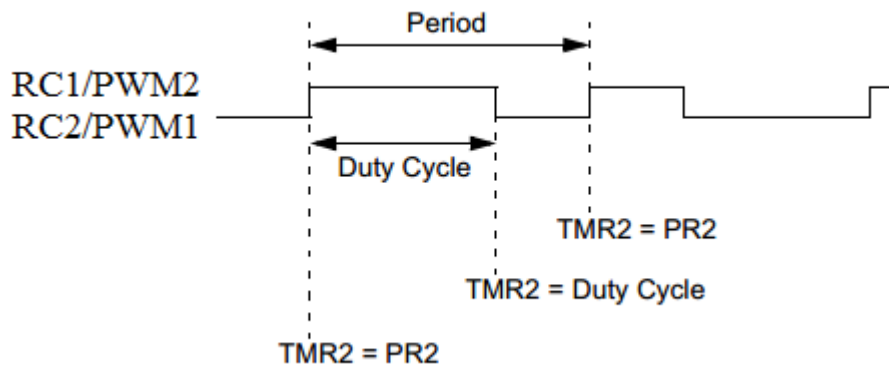
Ví dụ setup_timer_2(T2_DIV_BY_16, 255, 1) chu kỳ phát xung là $T_{PWM} = T_{OSC} \cdot 16384$, nếu tần số dao động là 16MHz thì chu kỳ phát xung là 1ms

- Cài đặt chế độ PWM

```
setup_ccpx(CCP_PWM);
```

- Đặt bề rộng xung

set_pwmxduty(Control); bề rộng xung là $duty = Control \cdot T_{PWM} / 1024$

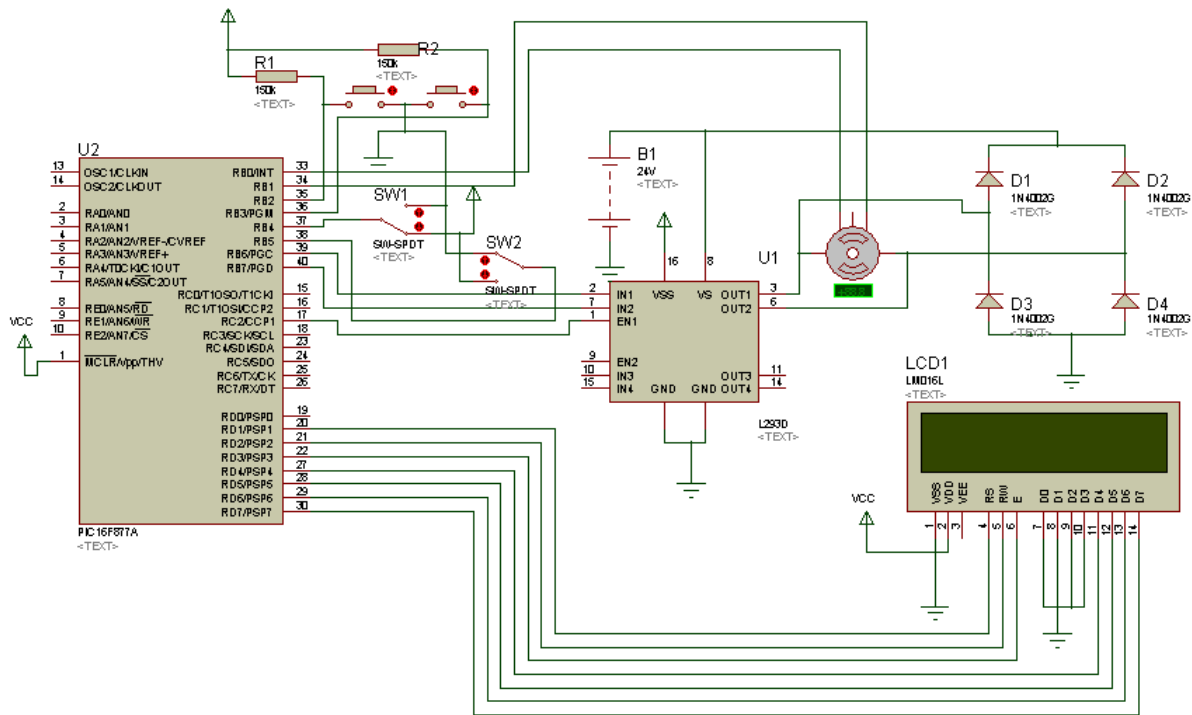


- Cài đặt chu kỳ lấy mẫu

Chu kỳ lấy mẫu dùng ngắt Timer 1

setup_timer_1(T1_INTERNAL|T1_DIV_BY_2): dùng dao động bên trong có chu kỳ $4T_{osc}$ sau đó nhân 2, với $f_{osc} = 16\text{MHz}$, chu kỳ xung nhịp vào timer 1 là $T_{ck} = 0.5\mu\text{s}$

set_timer1(value): muốn Timer 1 tràn sau thời gian T_s gây ngắt ta nạp vào số $value = 65536 - T_s / T_{ck}$



```
#include <16F877A.h>
#include <stdlib.h>
#include <math.h>

#fuses NOWDT,PUT,XT,NOPROTECT
#use delay (clock=16000000)

#define LCD_RS_PIN      PIN_D1
#define LCD_RW_PIN      PIN_D2
#define LCD_ENABLE_PIN  PIN_D3
#define LCD_DATA4       PIN_D4
#define LCD_DATA5       PIN_D5
#define LCD_DATA6       PIN_D6
#define LCD_DATA7       PIN_D7
#include <lcd.c>

float Sampling_time = 0.01;      //thoi gian lay mau (ms)
int16 Encoder_Resolution=200;// s? xung/vòng
int k=0,STOP;

signed int32 Pulse=0,pre_Pulse=0,pulse0=0;
float feedback,Err , Err1=0, Err2=0 ;
float MeasSpeed;
signed int32 Speed;
float Kp=250, Kd=20, Ki=50;
signed int32 Speed1=300,Speedp=0;//van toc can dieu khien vong/phut
float Controlled;
```

```

float Control;
void Motor_Control_PID(float vantocdat);
#int_EXT
void Count_isr(void)
{
    if(input(PIN_B1))Pulse++;///??m xung encoder chi?u thu?n
    else Pulse--; //Chi?u ngh?ch
}
#int_RB
void STOP_isr(void)
{
    if(!input(PIN_B5))
    {
        STOP=1;
        output_low(PIN_B7);output_low(PIN_B6);
    }
    ELSE STOP=0;
}
#int_TIMER1
void CONTROL_isr(void)
{
    {
        Set_timer1(45536);///10ms, sampling period
        k++;
        if(k==100)
        {speedp=pulse-pulse0;pulse=0;pulse0=0;
         pre_Pulse=0; k=0;
        }
        Motor_Control_PID(Controlled);
    }
}

void Motor_Control_PID(float vantocdat)
{
    feedback=(Pulse-pre_Pulse)/ Sampling_time;///tinh van toc
    pre_Pulse=Pulse;
    Err=vantocdat -feedback; //tinh error
    Control=Kp*Err+Ki*(Err+Err1)/2+Kd*(Err-Err1);
    Err1=Err;
    if (STOP==0)
    {
        if (Control>0){output_low(PIN_B7);output_high(PIN_B6);}
        else if (Control<0 ){output_high(PIN_B7);output_low(PIN_B6);}
        else {output_low(PIN_B7);output_low(PIN_B6);}
    }
    else {output_low(PIN_B7);output_low(PIN_B6);CONTROL=0;}
    control=abs(control);
}

```

```

    if (control>1023) {control=1023;}
    set_pwm1_duty(Control); //gan duty cycle cho CCP1 update PWM
}

void main()
{
    lcd_init();
    Delay_ms(100);
    set_tris_c(0x00);
    set_tris_b(0x3F);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_2); //Xung nh?p 0.5us
    setup_timer_2(T2_DIV_BY_16,255,1); //Chu k? PWM 1ms
    setup_ccp1(CCP_PWM); //Ch? ?? PWM
    enable_interrupts(INT_EXT); //ng?t ngoài RB0
    ext_int_edge( H_TO_L ); // Sets up EXT
    enable_interrupts(INT_RB);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);
    while(true)
    {
        if (!input(PIN_B2)) {SPEED1=speed1-50;}
        if (!input(PIN_B3)) {SPEED1=speed1+50;}
        if(input(PIN_B4)) {SPEED=SPEED1;}else {SPEED=-SPEED1;}
        Controlled= Speed*Encoder_Resolution/60;
        lcd_gotoxy(1,1);
        printf(lcd_putc,"SET SPEED: %Ld", SPEED);
        if(input(PIN_B5))
        {
            STOP=0;
            if (Speed>0){output_low(PIN_B7);output_high(PIN_B6);}
            else {output_high(PIN_B7);output_low(PIN_B6);}
        }
        else {STOP=1;output_low(PIN_B7);output_low(PIN_B6);}
        lcd_gotoxy(1,2);
        MeasSpeed=speedp*60.0/Encoder_Resolution;
        printf(lcd_putc,"SPEED      :%f",MeasSpeed) ;
        Delay_ms(1000);
    }
}

```

