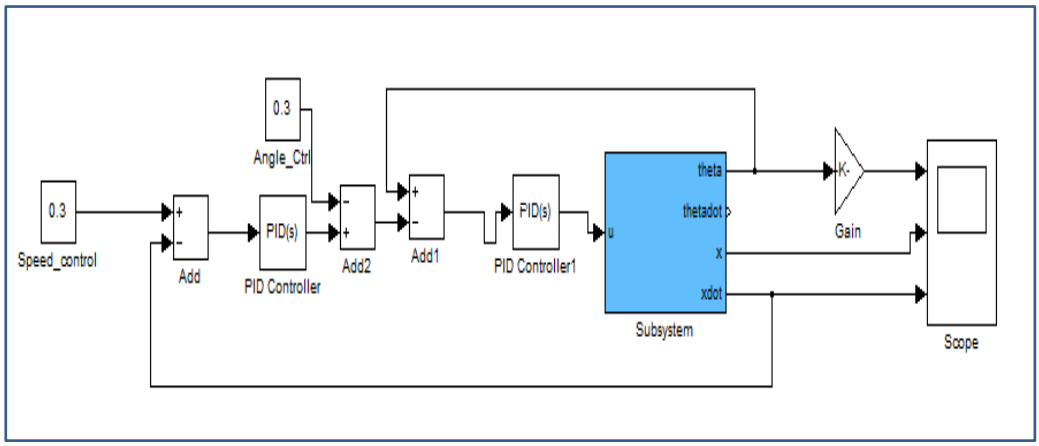
3.1.2. Bộ điều khiển PID cascade

Khi ta dung 1 bộ PID điều khiển góc, để xe vẫn đủ khả năng cân bằng nhưng vấn đề đặt ra là xe dao động di chuyển tới lui. Vì vậy để đạt chất lượng tốt hơn ta cần thêm một bộ điều khiển PID để đảm bảo vận tốc xe bằng 0.



Hình 3. 1 Sơ đồ simulink bộ điều khiển PID cascade

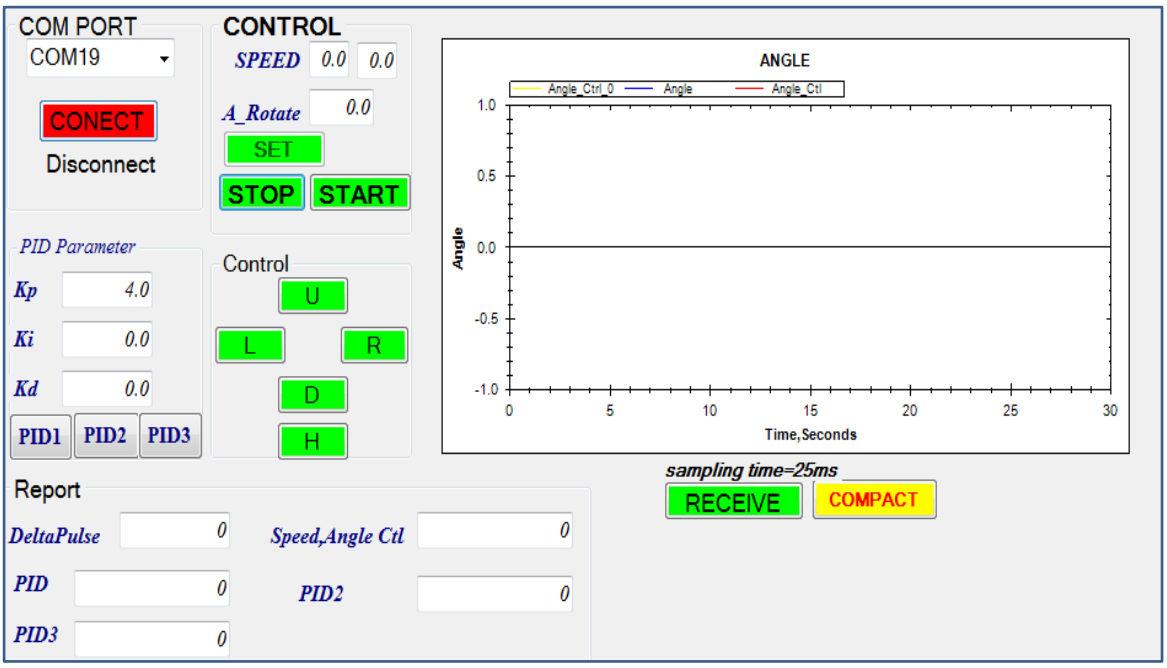
Ta có thể thấy ngõ ra của bộ PID vận tốc sẽ là ngõ vào của bộ PID góc.

Mô tả cách hoạt động:

- Đầu tiên khi tôi đặt một vận tốc điều khiển, xe sẽ đọc vận tốc về thông qua các giá trị đọc về từ 2 encoder từ đó tính ra sai số. Qua bộ PID thứ nhất ta sẽ được góc đặt cần thiết ứng với tốc độ đó.
- Sau khi có được góc đặt, học viên sẽ tiến hành đọc góc từ cảm biến Gy521 và tính toán để được sai số. Qua bộ PID góc thứ 2, đầu ra sẽ là các xung PWM và qua các mạch lái sẽ cho ra điện áp để điều khiển tốc độ quay của hai bánh xe và giữ xe cân bằng.

Phương pháp điều chỉnh hệ số PID :

- Phương pháp điều chỉnh thủ công: Đầu tiên tôi sẽ chọn Kp trước, quan sát mức độ đáp ứng. Sau khi thấy đủ lực để xe có thể đáp ứng với sai số góc thì lúc đó ta tăng dần KD lên. Nếu có hiện tượng steady error xảy ra thì ta càn tăng thêm KI.
- Phương pháp truyền từ C#: Tôi xây dựng một formload trên C# để truyền các giá trị Kp, Ki, KD thông qua giao tiếp serial.



Hình 3. 2 Hình formload Uart hiệu chỉnh thông số PID

1.1 Robot cân bằng

Tổng quan các vấn đề cần giải quyết

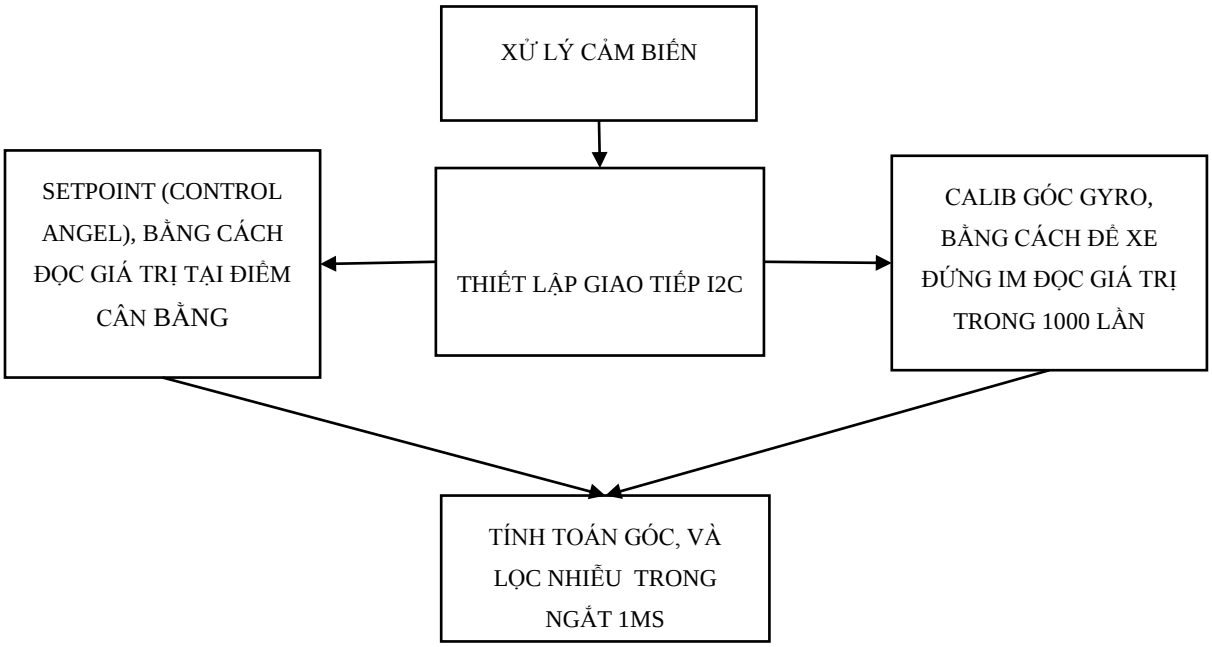
Vấn đề cần giải quyết khi đọc góc từ cảm biến :

- Các giá trị đọc về có nhiễu nên chúng ta phải tiến hành calib và dùng bộ lọc để đọc được góc chính xác.

Vấn đề trong xử lý giải thuật điều khiển PID cascade và thời gian xuất xung:

- Tính toán thời gian đọc mẫu về và thời gian phát xung sao cho hợp lý. Thường thì điều khiển thời gian phát xung sẽ lớn hơn thời gian đọc mẫu về. Thời gian phát xung và lấy mẫu không nên quá nhanh vì vi xử lý sẽ không còn thời gian xử lý những tác vụ khác, nhưng nếu thời gian quá chậm thì chất lượng của hệ thống không được đảm bảo. Ở đây học viên chọn thời gian lấy mẫu 1ms và thời gian phát xung 10ms.

3.2.1. Xử lý cảm biến đo góc



Hình 3. 3 Sơ đồ giải thuật xử lý cảm biến

Thiết lập chân giao tiếp I2C và khởi động cảm biến MPU6050 để đọc giá trị.

Tại điểm cân bằng của xe tiến hành đọc góc và đặt cố định là setpoint (Control angle).

Calib góc: để xe đứng yên đọc giá trị gyro lúc đó và dùng đó là gyroCalib (do accel không trôi nhiều nên không tiến hành calib).

Sau khi kết nối I2C và calib góc ở bước trên ta tiến hành tính toán góc

- Thực hiện ngắt 1ms vào đọc giá trị cảm biến, ta tiến hành theo trục x.
- Ta thực hiện đọc 1 lần giá trị từ cảm biến nhưng đọc thêm 9 giá trị trước đó gần nhất và chia ra lấy trung bình (có tác dụng như một bộ lọc trung bình giúp chuẩn xác hơn). Dùng buffer vòng.

⇨ Ưu điểm và nhược điểm như sau

Nhược điểm: Do vận tốc thay đổi đột ngột, nhưng do tính trung bình với 9 giá trị trước đó nên khi xe thay đổi vận tốc liên tục thì ít nhất sau 10ms (10 vòng lập) mới có giá trị chính xác. Sử dụng thêm kết hợp khi sử dụng bộ lọc bù.

Ưu điểm: Khi có bị nhiễu giao động thì nó sẻ giảm nhỏ lại.

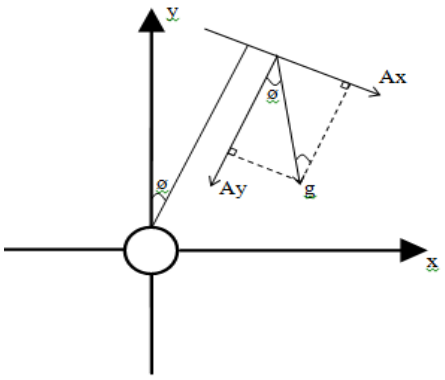
Giá trị AccelX được tính toàn bằng hàm arctan2 và không cần trừ đi góc calib.

Giá trị Gyrox mỗi lần đọc được sẽ tiến hành trừ đi gyro calib để chống trôi.

Từ giá trị accel và gyro ta dùng bộ lọc bù hoặc kalman để kết hợp suy ra góc hiện tại chính xác.

1.1.1.1 Cách tính góc theo accel và gyro theo trục

- **Tính theo accel**



Hình 3. 4 Hình minh họa cách tính giá trị góc từ giá trị đọc về từ cảm biến góc nghiêng

<u>Giải thích hình 5.2 một cách dễ hiểu hơn:</u>

Ở đây ta đang xoay cảm biến quanh trục z. Tức là trên cảm biến sẽ có 3 trục x,y,z ta giữ yên trục z và xoay cảm biến lúc này thì trục x và trục y thay đổi (để đơn giản hình dung ta có thể dùng 3 ngón tay cái, trỏ và giữa và xoay quanh 1 ngón)

Lúc này góc xoay của xe sẽ được tính toán như sau:

$Ax = g * \sin\theta$

$Ay = g * \cos\theta$

⇨ $\tan\theta = ax/ay \Rightarrow \theta = \arctan(Ax/ay)$

Tương tự như vậy ta sẽ tính được góc nghiêng tùy theo trục quay phụ thuộc vào cảm biến đặt trên xe theo hướng nào

$\theta x = \arctan(ay/az)$

Lưu ý: trong công thức code ta hay dùng arctan2 vì arctan2 mở rộng góc quay lớn hơn arctan

- **Tính theo góc theo gyro, cảm biến vận tốc**

$$\theta(t) = \int_{t1}^{t2} G(t)dt$$

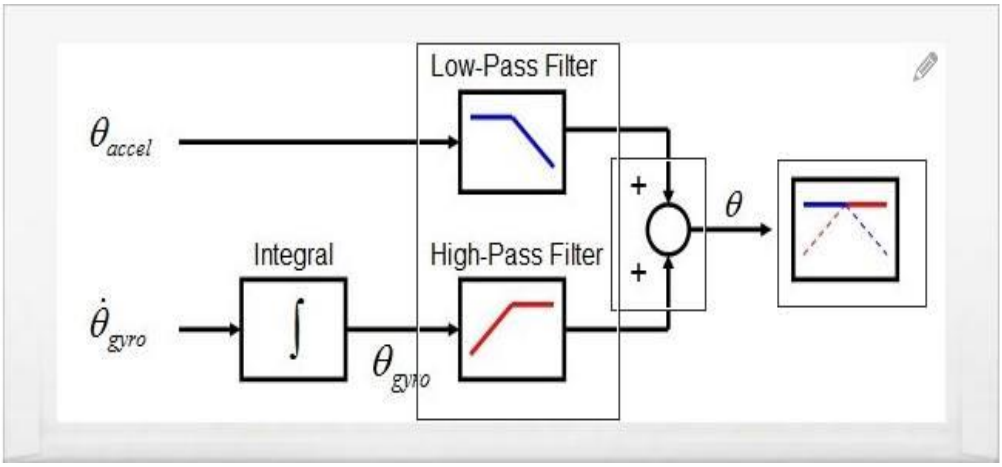
⇨ Tuy nhiên cảm biến gyro hay bị trôi nên ta sẽ tiến hành trừ đi giá trị calib góc.

Ta thực hiện calib bằng cách để cảm biến ở vị trí trục z 1g và y 0g (đơn giản hơn tuy có thể hơi thiếu chính xác thì có thể để cảm biến nằm thẳng theo mặt phẳng ngang xOy). Đọc giá trị gyroX 1000 lần và chia trung bình.Đặt giá trị này chính là giá trị góc calib

Tính được giá trị calib, thì sau khi đọc được giá trị hiện tại của gyro ta trừ đi calib sẽ được giá trị ra chính xác hơn , ít bị trôi.

Sau khi tính được các giá trị accel và gyro, ta sẽ đưa qua bộ lọc để lọc nhiễu và kết hợp hai giá trị này để được giá trị góc chính xác.

1.1.1.2 Bộ lọc bù



Hình 3. 5 Bộ lọc bù

Ta có thể hiểu một cách đơn giản bộ lọc bù bao gồm sự kết hợp giữa bộ lọc thông cao và lọc thông thấp.

Bộ lọc này thiết kế để hợp nhất giá trị từ accel và gyro để suy ra được giá trị góc nghiêng tốt nhất theo công thức.

Angle = A*(angle+ gyro*dt) + (1-A)*x_acc.

Theo hình ta sẽ thấy A ở đây là 0.98, vậy tại sao lại chọn như vậy:

Ở đây ta chọn thời gian dt =1ms thì ta có công thức tính thời gian liên tục của bộ lọc thông thấp và thông cao:

$$T = \frac{a*dt}{1-a} = \frac{0.98*0.001s}{0.02} = 0.049s$$

Ở đây T có thể được hiểu như là ranh giới giữa sự tin tưởng của gyro và accel, khi thời gian nhỏ hơn 0.049s thi gyro đáng tin hơn và được lọc, lớn hơn 0.049s thì accel được lọc.

Tùy hệ thống nhưng học viên thường thấy hay chọn (0.96 0.04 hoặc 0.98 0.02).

1.1.1.3Bộ lọc kalman

Một số định nghĩa cơ bản thường gặp khi nghiên cứu bộ lọc Kalman:

- Varian: phương sai "trung bình của bình phương khoảng cách của mỗi điểm tới dữ liệu trung bình"
- Covarian: Hiệp phương sai: sự biến thiên cùng nhau của 2 biến ngẫu nhiên
- Giá trị kì vọng E: là giá trị nhân với xác suất tương ứng ra số chính xác

Có thể hiểu đây là một loạt các phép đo theo thời gian để dự đoán chính xác hơn phép đo một mình.

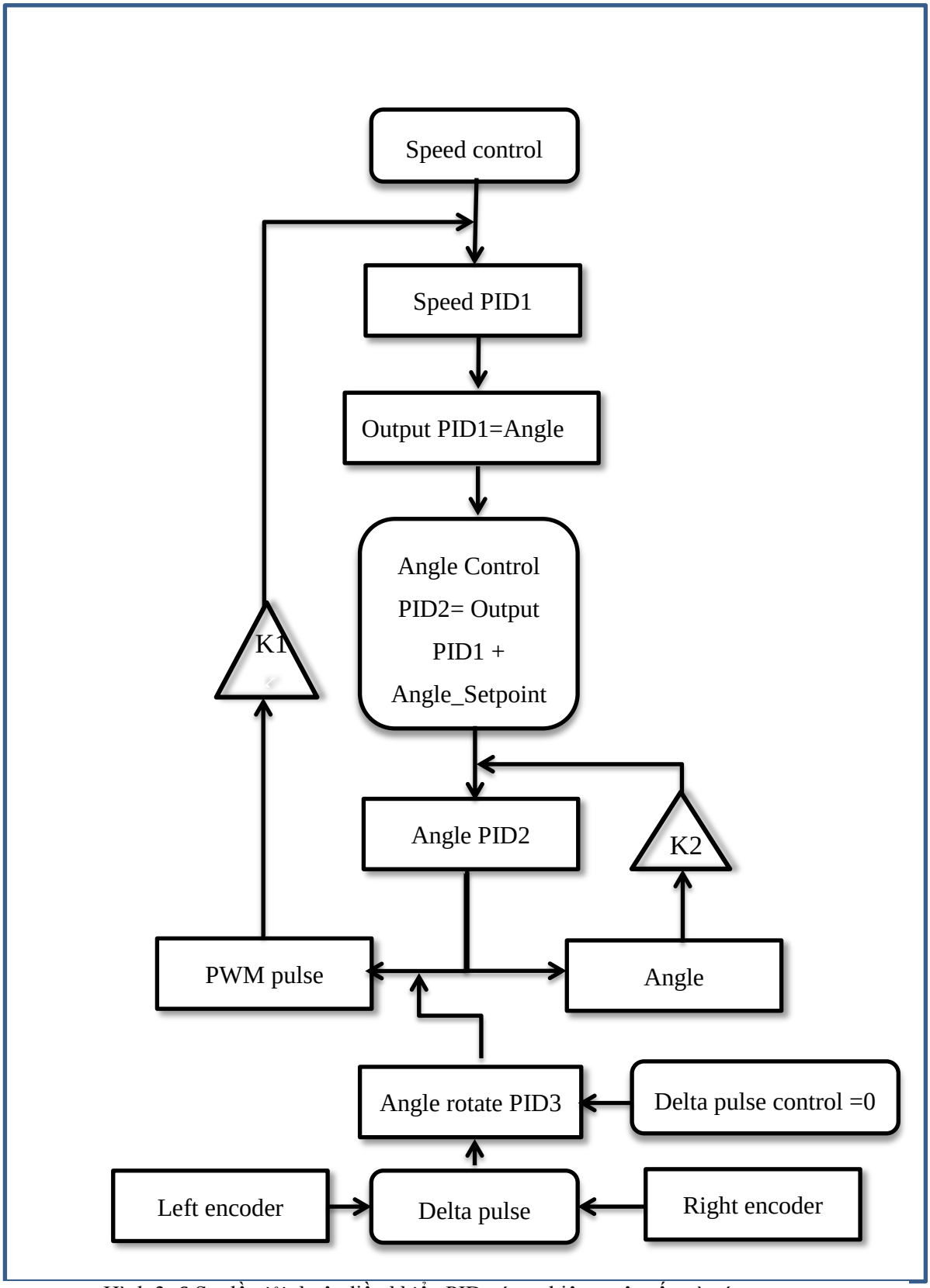
Khắc phục sự nhiễu do gia tốc accelerometer và sự trôi góc của gyro.

Lưu ý: thời gian ngắn thì gyro ảnh hưởng, thời gian dài sẽ là gia tốc accel.

Ta thực hiện bộ lọc kalman qua 7 bước.

3.2.2. Bộ điều khiển PID

- Tiếp theo ta sử dụng bộ PID số 1 để đưa sai số góc so với điểm đặt cân bằng về bằng 0.
- Tiếp theo nếu xe vẫn giao động quanh điểm cân bằng ta sẽ thực hiện bộ PID số 2, bộ PID vận tốc với vận tốc đặt là 0, để xe không di chuyển tại điểm cân bằng. Ngõ ra của bộ PID sẽ là góc, để đưa vào bộ PID số 1.
- Tuy nhiên phần cơ khí có thể làm hai bánh xe quay không đều nhau nên ta sử dụng thêm một bộ PID số 3 để điều khiển tốc độ hai bánh bằng nhau. Số xung sẽ được cộng vào ở bánh chậm hơn và trừ đi ở bánh nhanh hơn. Ngõ ra sẽ là xung PWM được công dồn với hai bộ PID đầu tiên.



Hình 3. 6 Sơ đồ giải thuật điều khiển PID góc nghiêgn, vận tốc và góc xoay

1.2 Robot di chuyển

Tương tự như robot thăng bằng nhưng lúc này vận tốc đặt >0 nếu muốn xe đi tới, vận tốc đặt <0 nếu muốn xe đi lùi.

Khi robot di chuyển quẹo trái thì vận tốc đặt >0, và deltapulse = pulse phải – pulstrai > 0.

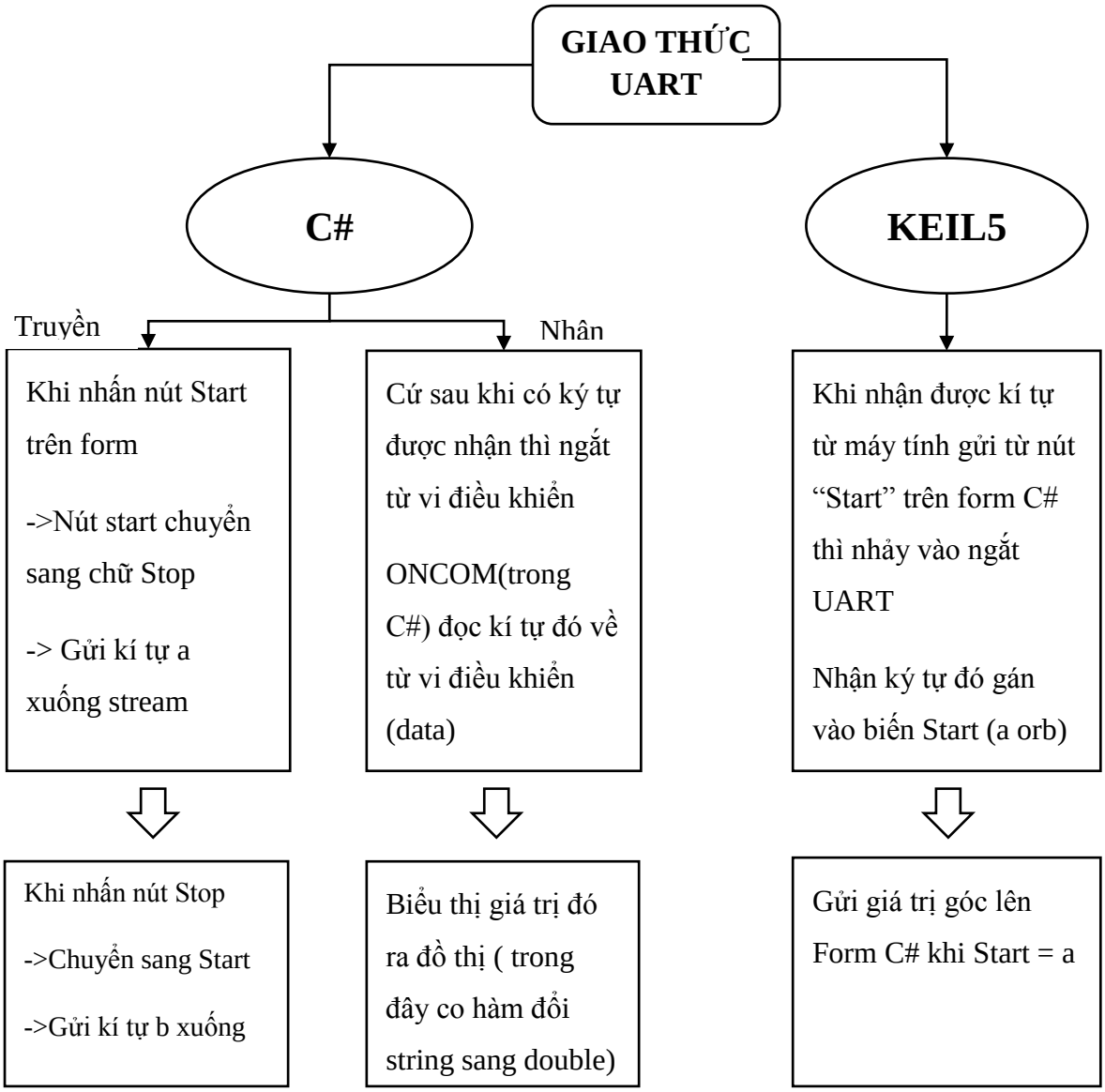
Khi robot di chuyển quẹo trái thì vận tốc đặt >0, và deltapulse < 0.

1.3 Robot xoay

Tương tự robot cân bằng nhưng:

- Khi robot xoay trái thì deltapulse <0, vận tốc đặt =0
- Khi robot xoay phải thì deltapulse >0, vận tốc đặt =0

GIao thức UART:

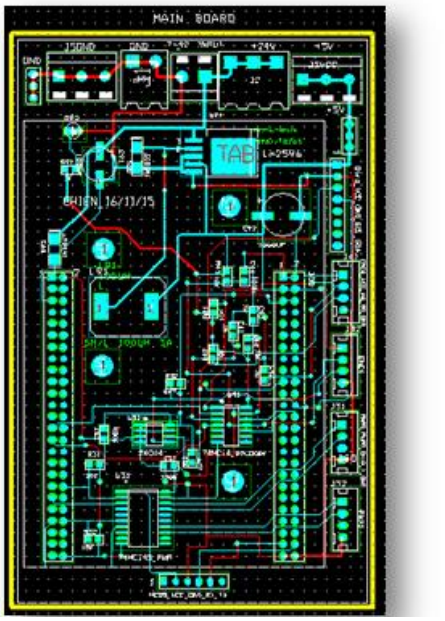


Hình 4. 1 Giao thức truyền UART giữa máy tính và vi điều khiển

<u>Yêu cầu đạt được sau khi đọc góc:</u> góc đọc về có chính xác không, đáp ứng nhanh không, khi thay đổi góc lớn có bị trôi nhiều không, góc lớn nghiêng về đều về 2 bên có đọc về chính xác không , lúc đó ta sẽ biết là làm có chính xác không.

HardWare:

- Board ngoai vi:



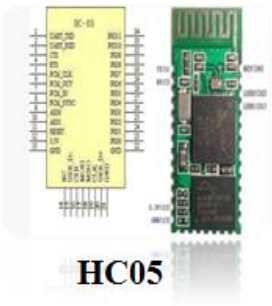
Encoder

Driver MB3B

Sensor Gy521

Module Bluetooth HC05

Micro controller



HC05

RF-HC11