

Trường Đại học Khoa học Tự nhiên

Khoa: Toán - Cơ - Tin học

Ngành: Máy tính và Khoa học Thông tin CLC



Nhóm: 3

THIẾT KẾ XE TỰ ĐỘNG CHẠY THEO QUỶ ĐẠO SỬ DỤNG GYROSCOPE

Giảng viên hướng dẫn:

TS. Lê Quang Thảo

Sinh viên thực hiện:

Phạm Hoài Nam - 21000394

Nguyễn Hồng Nguyên - 21001572

Đặng Đình Thắng – 21001586

Đỗ Văn Thuận - 21001590

Dương Minh Tuấn - 21001595

Đoàn Văn Tuấn - 21000523

HÀ NỘI 2023

Mục Lục

Lời cảm ơn.....	3
Danh sách tên viết tắt	4
Danh mục hình vẽ	5
Chương 1: Tổng quan.....	6
1.1 Ngôn ngữ được sử dụng.....	6
1.2 Phần mềm sử dụng.....	6
Chương 2: Giới thiệu sản phẩm.....	7
2.1 Cấu tạo và thông số kỹ thuật.....	7
2.2 Sơ đồ mạch và sơ đồ thuật toán	10
2.3 Kết quả.....	14
Chương 3: Tổng kết.....	15
Tài liệu tham khảo	16
Code	16

Lời cảm ơn

Chúng em xin gửi lời cảm ơn chân thành đến thầy giáo bộ môn hệ thống nhúng, thầy Lê Quang Thảo, với sự tận tâm và sự hướng dẫn tận tình trong suốt thời gian chúng em theo học. Thầy đã luôn dành thời gian và kiến thức quý báu để giúp đỡ chúng em trong quá trình nắm bắt kiến thức về hệ thống nhúng.

Những lời chỉ dẫn, phản hồi, và lời khích lệ từ thầy đã giúp chúng em phát triển kỹ năng và hiểu biết đáng kể. Điều này đã tạo nên một tác động tích cực đến sự phát triển của chúng em trong lĩnh vực hệ thống nhúng.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành đến thầy giáo Lê Quang Thảo với lòng biết ơn sâu sắc.

Danh sách tên viết tắt

Ký hiệu	Tên đầy đủ
IDE	Integrated Development Environment
EEPROM	Electrically Erasable Programmable Read Only Memory
SRAM	Static Random Access Memory
PID	Pulse with Modulation
PWM	Proportional Integral Derivative
LED	Light emitting diode

Danh mục hình vẽ

Hình 1	Phần mềm Arduino IDE
Hình 2	Arduino Nano with ATmega328P MCU
Hình 3	MPU - 6050
Hình 4	L298
Hình 5	Động cơ, bánh xe
Hình 6	Sơ đồ mạch
Hình 7	Sơ đồ thuật toán

Chương 1: Tổng quan

1.1 Ngôn ngữ được sử dụng

Để lập trình và viết code cho thiết bị xe tự động chạy theo quỹ đạo dùng **Gyroscope (ATmega)** nhóm sử dụng **ngôn ngữ C** – một ngôn ngữ phổ biến và dễ sử dụng .

Bên cạnh đó , việc dùng ngôn ngữ **C** trong lập trình nói chung và trong lập trình nhúng nói riêng có các ưu điểm như :

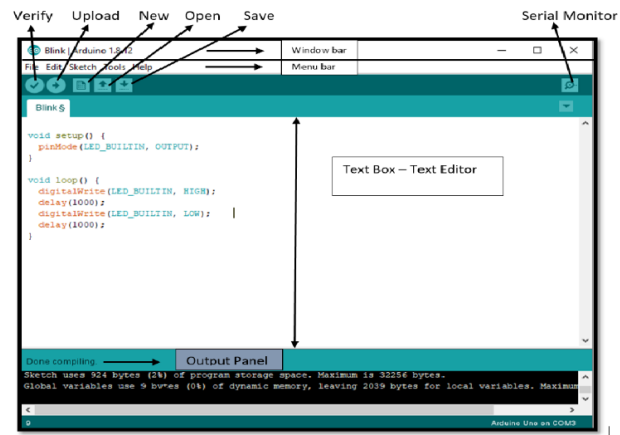
- + Gắn gũi với phần cứng: **C** được phát triển để làm việc trực tiếp với phần cứng.
- + Hiệu suất cao: **C** là ngôn ngữ được biên dịch, cho phép mã hóa được tối ưu hóa máy và thực thi nhanh. Điều này rất quan trọng trong miền nhúng, nơi hệ thống có tài nguyên chế độ và yêu cầu thực thi hiệu suất cao.
- + **C** là một ngôn ngữ đa nền tảng, có thể chạy trên nhiều kiến trúc phần cứng và hệ điều hành khác nhau. Điều này cho phép tái sử dụng mã nguồn và tạo ra các hệ thống nhúng linh hoạt và di động.

1.2 Phần mềm sử dụng

Sử dụng phần mềm Arduino IDE

Arduino IDE là một phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino . Đây là một phần mềm Arduino chính thức, giúp cho việc biên dịch mã trở nên dễ dàng.

Arduino IDE sử dụng ngôn ngữ lập trình **C/C++** rất phổ biến trong giới lập trình. Bất kỳ đoạn code nào của **C/C++** thì **Arduino IDE** đều có thể nhận dạng, giúp các lập trình viên thuận tiện trong việc thiết kế chương trình lập cho các bo mạch Arduino.



Hình 1: Phần mềm Arduino IDE

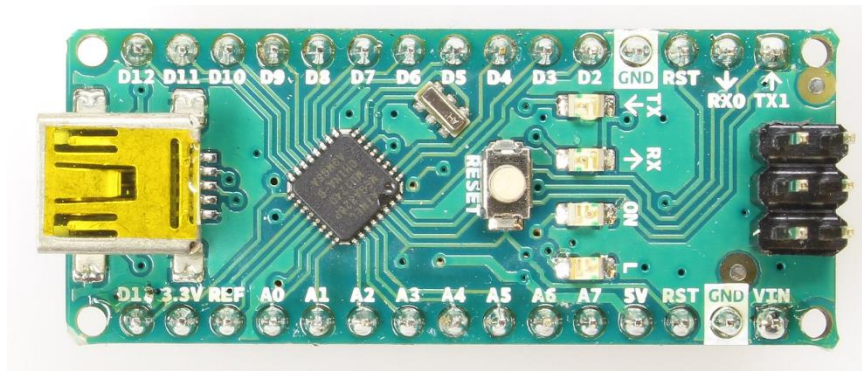
Điều đáng chú ý nhất về Arduino IDE đó là sự hỗ trợ hỗ trợ lập trình tốt cho bo mạch Arduino . Arduino có một module quản lý bo mạch, nơi người dùng có thể chọn bo mạch mà họ muốn làm việc cùng và có thể thay đổi bo mạch thông qua Menu. Quá trình sửa đổi lựa chọn cũng liên tục tự động cập nhật để các dữ liệu có sẵn trong bo mạch và dữ liệu sửa đổi đồng nhất với nhau. Bên cạnh đó, **Arduino IDE** cũng giúp bạn tìm ra lỗi từ code mà bạn biết giúp bạn sửa lỗi kịp thời tránh tình trạng bo mạch Arduino làm việc với code lỗi quá lâu dẫn đến hư hỏng hoặc tốc độ xử lý bị giảm sút.

Chương 2: Giới thiệu sản phẩm

2.1 Cấu tạo và thông số kỹ thuật

1. Arduino nano:

Arduino Nano là một trong những bo mạch phát triển nhỏ gọn và mạnh mẽ của Arduino, được thiết kế để giúp các nhà phát triển dễ dàng tạo ra các dự án điện tử đa dạng. Với vi điều khiển mạnh mẽ từ Microchip ATmega328P hoặc ATmega168 (tùy phiên bản), Arduino Nano cung cấp một nền tảng linh hoạt để phát triển các ứng dụng từ đơn giản đến phức tạp.



Hình 2: Arduino Nano with ATmega328P MCU

Thông số kỹ thuật:

Vi điều khiển: Microchip ATmega328P

Điện áp hoạt động: 5 volts

Điện áp đầu vào: Từ 5 đến 20 volts

Chân I/O kỹ thuật số: 14 (6 trong số đó có khả năng đầu ra PWM tùy chọn)

Chân vào analog: 8

Dòng điện tối đa cho mỗi chân I/O: 40 mA

Dòng điện cho chân 3.3V: 50 mA

Bộ nhớ Flash: 32 KB, trong đó 2 KB được sử dụng cho bootloader

SRAM: 2 KB

EEPROM: 1 KB

Tốc độ xung nhịp: 16 MHz

Chiều dài: 45 mm

Chiều rộng: 18 mm

Trọng lượng: 7 g

USB: Mini-USB Type-B

Đặc tính:

Arduino Nano có kích thước nhỏ gọn với chiều dài chỉ 45mm và chiều rộng 18mm, làm cho nó rất thích hợp cho các dự án có yêu cầu không gian hạn chế. Bên cạnh đó, nó có các chân I/O kỹ thuật số và analog, hỗ trợ PWM và nhiều tính năng khác, giúp bạn kết nối và điều khiển nhiều loại cảm biến, thiết bị và module ngoại vi khác nhau.

Với bộ nhớ Flash lớn, tốc độ xung nhịp 16 MHz và khả năng kết nối thông qua cổng USB Mini-USB Type-B, Arduino Nano là một công cụ mạnh mẽ để bạn tiến hành phát triển và thử nghiệm các ứng dụng điện tử một cách thuận tiện và sáng tạo.

2. MPU6050

MPU-6050 là một cảm biến gia tốc và gia tốc góc (gyroscope) kết hợp trong một module, thường được sử dụng để đo chuyển động và góc xoay trong các ứng dụng điện tử và robotics. MPU-6050 được phát triển với mục tiêu cung cấp thông tin về vận tốc và góc xoay trong thời gian thực.

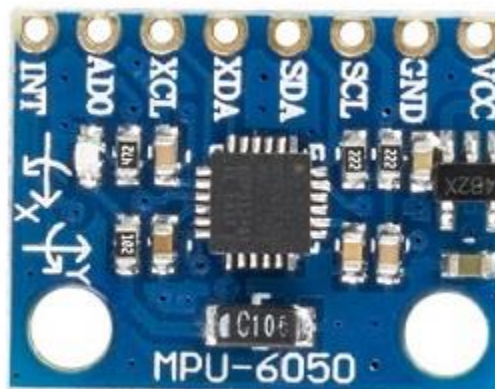
Thông số kỹ thuật:

Đầu ra kỹ thuật số I2C của dữ liệu MotionFusion 6 hoặc 9 trục trong định dạng ma trận quay, số tứ giác (quaternion), góc Euler hoặc dữ liệu thô.

Điện áp đầu vào: 3 - 5V.

Có các lựa chọn Solder Jumpers trên các chân CLK, FSYNC và AD0.

Cảm biến tốc độ góc (gyro) 3 trục với độ nhạy lên đến 131 LSBs/dps và phạm vi tỉ lệ đầy đủ từ ± 250 ,



Hình 3: MPU - 6050

± 500 , ± 1000 và ± 2000 dps.

Cảm biến gia tốc 3 trục với phạm vi tỉ lệ đầy đủ có thể lập trình từ $\pm 2g$, $\pm 4g$, $\pm 8g$ và $\pm 16g$.

Bộ xử lý chuyển động số hóa (DMP™)

Các thuật toán nhúng cho việc hiệu chỉnh thời gian thực và hiệu chuẩn la bàn. Không cần sự can thiệp của người dùng.

Cảm biến nhiệt độ với đầu ra kỹ thuật số.

Kích thước: 20 * 15 * 1.6mm.

Đặc điểm chính của MPU-6050:

Cảm biến Gia tốc (Accelerometer): Có thể đo gia tốc tuyến tính trên các trục XYZ, cho phép xác định thay đổi vị trí hoặc phát hiện chuyển động.

Cảm biến Gia tốc góc (Gyroscope): Cung cấp thông tin về tốc độ góc xoay của module xung quanh các trục XYZ.

Giao tiếp: MPU-6050 thường được giao tiếp với các vi điều khiển hoặc bo mạch phát triển thông qua giao tiếp I2C (Inter-Integrated Circuit), giúp dễ dàng kết nối và truy cập dữ liệu từ module.

Độ chính xác: MPU-6050 được thiết kế với độ chính xác cao, cho phép đo đạc các biến đổi vận tốc và góc xoay với độ chính xác tương đối cao.

Ứng dụng: MPU-6050 thường được sử dụng trong các ứng dụng như robot tự hành, điều khiển chuyển động của quadcopter, thu thập dữ liệu chuyển động trong thể thao hoặc theo dõi các dự án liên quan đến chuyển động và góc xoay.

Tính khả năng kết hợp cảm biến gia tốc và gia tốc góc trong một module nhỏ gọn và dễ sử dụng, MPU-6050 đã trở thành một phần quan trọng trong việc phát triển các ứng dụng đo đạc và kiểm soát chuyển động trong nhiều lĩnh vực công nghiệp và dự án cá nhân.

3. L298N

L298N là module điều khiển động cơ trong các xe DC và động cơ bước. Module có một IC điều khiển động cơ L298 và một bộ điều chỉnh điện áp 5V 78M05. Module L298N có thể điều khiển tối đa 4 động cơ DC hoặc 2 động cơ DC với khả năng điều khiển hướng và tốc độ

Đặc điểm và thông số kỹ thuật module L298N

Module điều khiển: 2A L298N

Chip điều khiển: Cặp H-Bridge L298N

Điện áp cấp cho động cơ (Tối đa): 46V

Dòng điện cấp động cơ (tối đa): 2A

Điện áp logic: 5V

Điện áp hoạt động của IC: 5-35V

Dòng điện hoạt động IC: 2A

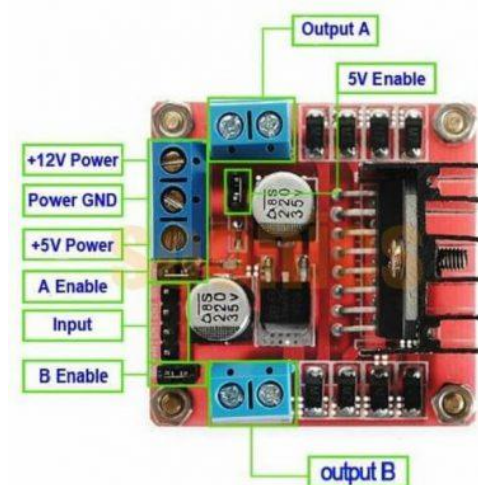
Dòng logic: 0-36mA

Công suất tối đa (W): 25W

Cảm biến dòng điện cho mỗi động cơ

Có tản nhiệt cho hiệu suất tốt hơn

Có đèn báo LED bật nguồn



Hình 4: L298N

4. Động cơ, bánh xe

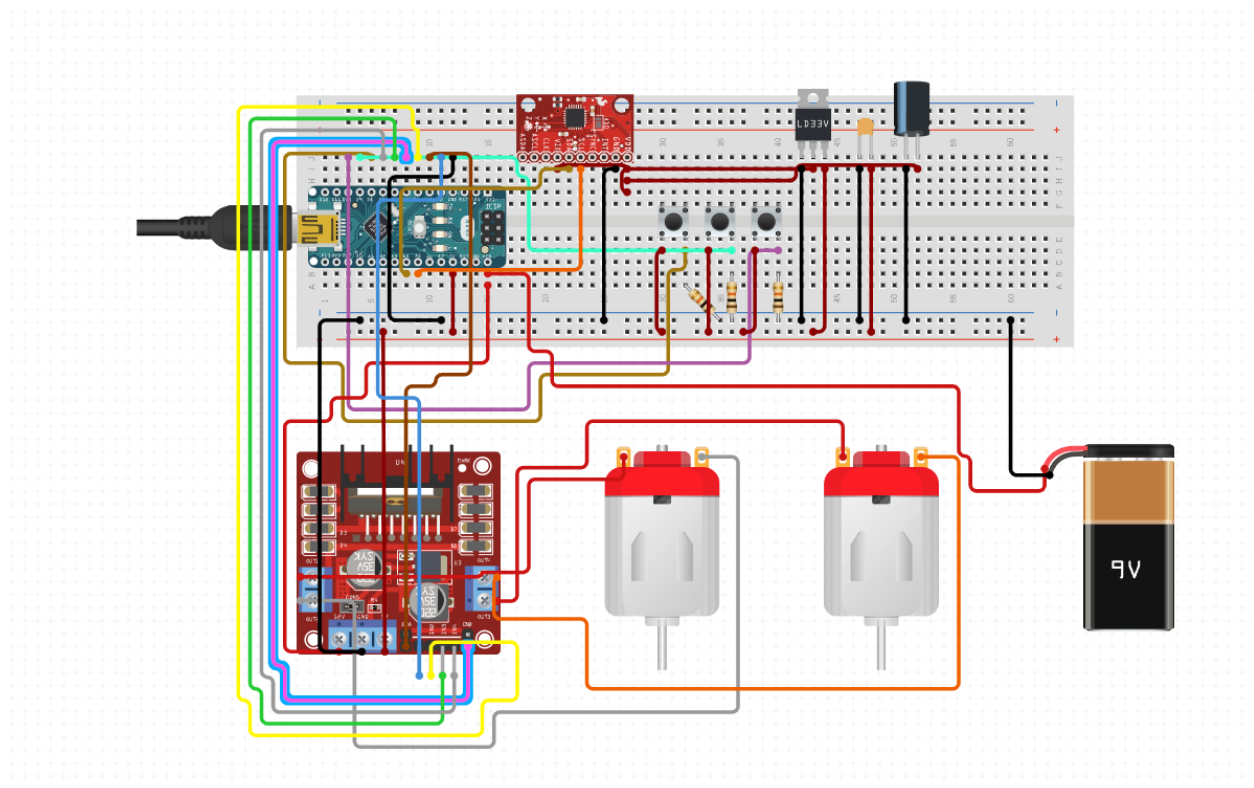
Thông số kỹ thuật

- 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 70mm: 26m/1p) 208 Vòng/ 1 Phút (5V) (Với bánh 70mm: 44m/1p)
- Dòng không tải: 70mA (250mA MAX)

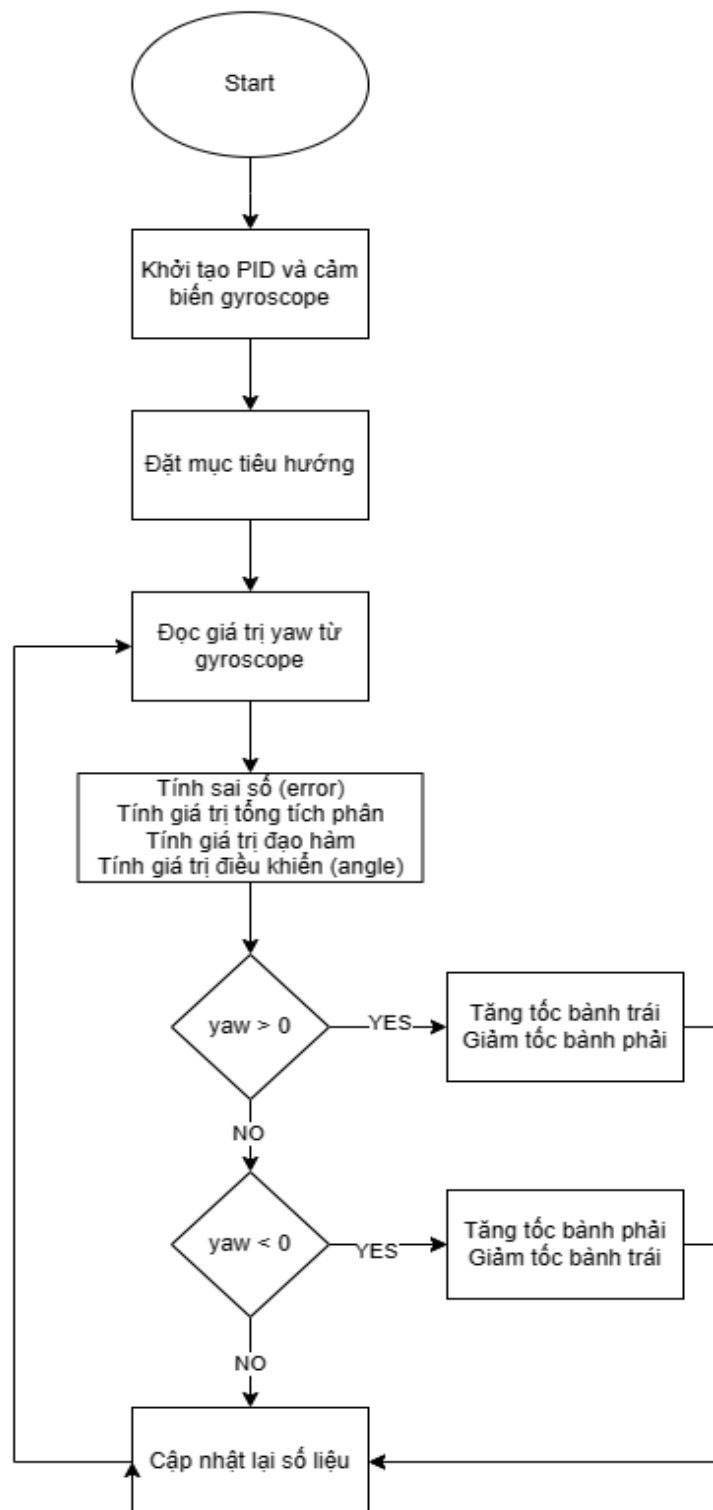


Hình 5: Động cơ, bánh xe

2.2 Sơ đồ mạch và sơ đồ thuật toán



Hình 6: Sơ đồ mạch



Hình 7: Sơ đồ thuật toán

Tính dữ liệu sai số

```
for (int x = 0; x < 1000; x++) {
    Wire.beginTransaction(MPU);
    Wire.write(0x43);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 6, true);
    GyroX = (Wire.read() << 8 | Wire.read())/131.0;
    GyroY = (Wire.read() << 8 | Wire.read())/131.0;
    GyroZ = (Wire.read() << 8 | Wire.read())/131.0;

    froll += GyroX;
    fpitch += GyroY;
    fyaw += GyroZ;
}

froll /= 1000;
fpitch /= 1000;
fyaw /= 1000;
```

```
Serial.println("Probe fillter done!!!!");
```

Vấn đề mà người dùng MPU6050 nào cũng gặp phải là hiện tượng bị trôi góc mặc dù để MPU6050 đứng yên. Vì thế nên chúng ta cần phải tính thông số bị trôi để có thể tính được góc xoay có độ chính xác cao.

Arduino yêu cầu MPU6050 gửi 6 bytes dữ liệu.

Mỗi giá trị GyroX, GyroY, GyroZ là 16 bits bao gồm 8 bits cao và 8 bits thấp được gộp lại.

Gyro chia 131.0 thì sẽ thành degree/s.

Ở đây tính phần nhiễu trong 1000 lần đo.

Tính góc lệch từ MPU

```
previousTime = millis();
Wire.beginTransaction(MPU);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(MPU, 6, true);
GyroX = (Wire.read() << 8 | Wire.read())/131.0;
GyroY = (Wire.read() << 8 | Wire.read())/131.0;
GyroZ = (Wire.read() << 8 | Wire.read())/131.0;

roll += (GyroX - froll) * dt;
pitch += (GyroY - fpitch) * dt;
yaw += (GyroZ - fyaw) * dt;
Serial.print("Z = "); Serial.println((int)yaw);
currentTime = millis();
```

```
elapsedTime = currentTime - previousTime;  
dt = elapsedTime / 1000;
```

Như đã trình bày ở trên: Arduino yêu cầu MPU6050 gửi 6 bytes dữ liệu. Sau khi gửi hoàn tất, sẽ tiến hành đọc dữ liệu

Mỗi giá trị GyroX, GyroY, GyroZ là 16 bits bao gồm 8 bits cao và 8 bits thấp.

Gyro chia 131.0 thì sẽ thành degree/s. Sau đó trừ đi phần nhiễu, cộng dồn nhân với dt sẽ ra góc lệch hiện tại với phương ban đầu (do góc lệch biến đổi theo thời gian nên cần 2 biến previousTime và currentTime để tính).

Dùng PID để giữ cho xe đi thẳng

```
void pid() {  
    error = target - yaw*3.141592654/180; // proportional  
    integral = integral + error; //integral  
    derivative = error - last_error; //derivative  
  
    angle = (error * Kp) + (integral * Ki) + (derivative * Kd);  
    Serial.print("angle = "); Serial.println(angle);  
  
    forward(mtrSpd, mtrSpd);  
  
    if (yaw > 0) {  
        forward(mtrSpd+abs(angle), mtrSpd-abs(angle));  
    } else if (yaw < 0) {  
        forward(mtrSpd-abs(angle), mtrSpd+abs(angle));  
    }  
    last_error = error;  
}
```

Với các hệ số dựa trên tính toán:

```
float Kp = 100;  
float Ki = 10;  
float Kd = 70;
```

Ở đây target = 0, chính là hướng xe đi thẳng ban đầu. Những giá trị Kp, Ki, Kd này là trong quá trình thử nghiệm tìm ra để cho xe chạy ổn định nhất.

Sử dụng thuật toán PID để tính angle để thay đổi vận tốc 2 bánh xe giữ cho xe không đi lệch khỏi quỹ đạo.

2.3 Kết quả

Sau khi hoàn thành mô hình và phần code, nhóm đã thực hiện video kiểm tra và phân tích thành phẩm.

Link video:

<https://drive.google.com/drive/folders/1liut8skcd5rNXvzuRWwnb3gWBJVIVlc?fbclid=IwAR0v2oNO6zsu360W1XWe2eORPQKJO1xbP2sHdEKq2ecx3Agn9GWUEzRdPZs>

Sau quá trình nghiên cứu và thiết kế, nhóm thu được sản phẩm xe điều khiển với các chức năng và đặc điểm như sau:

- Kết quả sản phẩm của thiết kế xe tự động chạy theo quỹ đạo sử dụng gyroscope là một chiếc xe có khả năng tự động điều hướng và di chuyển theo một quỹ đạo cụ thể dựa trên thông tin từ gyroscope.
- Xe gồm 2 động cơ hoạt động độc lập, hỗ trợ xe thực hiện các năng đi tiến, lùi, quay trái, quay phải, vòng trái, vòng phải, điều chỉnh tốc độ.
- Sử dụng thuật toán PID để giữ cho xe đi thẳng.
- Khi bấm nút 1 thì xe đi hình vuông rẽ bên phải quay lại
- Khi bấm nút 2 thì xe đi hình vuông rẽ bên trái quay lại
- Khi bấm nút 3 là đi thẳng quay lại

Chương 3: Tổng kết

Trong bài viết này, chúng tôi đã trình bày về thiết kế một mô hình xe tự động chạy theo quỹ đạo sử dụng gyroscope. Mục tiêu của chúng tôi là xây dựng một hệ thống xe tự động có khả năng nhận biết và duy trì quỹ đạo di chuyển một cách chính xác bằng cách sử dụng cảm biến gyroscope.

Qua quá trình nghiên cứu và thiết kế, chúng tôi đã nhận thấy rằng sử dụng gyroscope là một phương pháp hiệu quả để đo và điều khiển tốc độ góc của xe. Gyroscope cho phép chúng tôi đo lường chính xác các thay đổi về hướng di chuyển của xe và sử dụng thông tin này để điều chỉnh hướng di chuyển của xe.

Mô hình xe tự động chạy theo quỹ đạo sử dụng gyroscope có nhiều ứng dụng tiềm năng. Nó có thể được áp dụng trong các lĩnh vực như xe tự động điều khiển trong công nghiệp, xe tự động trong giao thông vận tải, hoặc thậm chí trong các ứng dụng giải trí và giáo dục.

Sau khi hoàn thiện báo cáo, nhóm nhận thấy một số hạn chế của mô hình xe tự động sau:

- Xe vẫn chưa chạy được đến theo tọa độ cho trước
- Nhạy cảm với nhiễu: Gyroscope có thể nhạy cảm với nhiễu và rung động, điều này có thể ảnh hưởng đến sự chính xác của thông tin mà nó cung cấp.
- Khả năng hỏng hóc: Gyroscope là một phần cơ học và có thể bị hỏng hóc do va chạm hoặc sự mài mòn, đòi hỏi bảo dưỡng thường xuyên.

Tài liệu tham khảo

<https://chipn24.com/modul-1298-dieu-khien-dong-co-cau-h-id256.html>

https://en.wikipedia.org/wiki/Arduino_Nano?fbclid=IwAR3-qVSqXJ7oKVwSX5XLUDeHuFw1CKPp4Iuz5qXDRxSNJB5MfhvvYkBf-HA

https://www.elecrow.com/wiki/index.php?title=MPU-6050&fbclid=IwAR0fzV_ntIO0GSjg3tFaU99OVGwbur8S_fEIjDLL9qfHt_dnyPMU7XMWBCo

https://www.researchgate.net/figure/MPU6050-module-a-sensor-module-b-working-axis-details-and-c_fig1_370797001

<http://arduino.vn/result/5473-robot-hai-banh-tu-can-bang-2-wheel-self-balancing-robot>

<https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/>

Code

<https://drive.google.com/drive/folders/1wyWB9UXz6IMX4cqhAAWaawaHeuLaSywf?usp=sharing>