

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



DỒ ÁN THIẾT KẾ LUẬN LÝ(CO3091)

KHÓA HỌC: DỒ ÁN THIẾT KẾ LUẬN LÝ

**DỒ ÁN: SỬ DỤNG CẢM BIẾN IMU
ĐỂ ĐIỀU KHIỂN XE THÔNG 4 BÁNH**

Giảng viên hướng dẫn: Tran Thanh Binh

Sinh viên thực hiện: Lâm Minh Vinh - 2120082
To Diu Quang - 2014251

Ngày nộp báo cáo: 16th December 2023



Contents

1 Mục tiêu và ý tưởng đồ án	2
1.1 Mục tiêu	2
1.2 Ý tưởng thực hiện	2
2 Đặc tả phần cứng đề xuất	2
2.1 Arduino Nano	2
2.2 NRF24L01 + RF Module	3
2.3 L298N Motor Driver	5
2.4 ADXL335 Module	6
2.5 TT Gear Motor	6
2.6 18650 li-ion battery	7
3 Cảm biến gia tốc IMU	8
3.1 Cấu hình chân cảm	8
3.2 Nguyên lý hoạt động	8
4 Giao thức sử dụng	10
4.1 Khái niệm giao thức RF	10
4.2 Gói tin truyền dẫn	11
4.3 Kiểm soát gói tin	11
5 Thiết kế và hiện thực	13
5.1 Thiết kế hệ thống	13
5.2 Transmitter	13
5.3 Receiver	14
5.4 Hiện thực phần cứng	15
5.4.1 Transmitter	15
5.4.2 Receiver	15
5.4.3 Sản phẩm hoàn chỉnh	16
6 Mã nguồn và demo hiện thực	16
List of Figure	17
7 References	18



1 Mục tiêu và ý tưởng đồ án

1.1 Mục tiêu

Trong đồ án này, báo cáo này trình bày việc sử dụng Cảm biến Đơn vị Đo Trục Trọng lực và Góc Độ (IMU) để điều khiển một xe thông minh với 4 bánh. IMU được tích hợp để theo dõi chuyển động và hướng của xe, và thông tin này được sử dụng để điều khiển động cơ mô hình lái xe. Nhóm tập trung vào việc phát triển một giao thức điều khiển hiệu quả, giúp xe di chuyển một cách ổn định và linh hoạt. Ngoài ra, đồ án cũng bao gồm việc triển khai và thử nghiệm các chức năng điều khiển sử dụng IMU trên xe, đồng thời đề xuất hướng phát triển tiềm năng cho các ứng dụng tương lai của công nghệ này trong lĩnh vực xe tự động và robot di động.

1.2 Ý tưởng thực hiện

Ý tưởng cho đồ án sử dụng IMU để điều khiển xe thông minh với 4 bánh có thể được mô tả như sau:

- Mục Tiêu Chính: Phát triển một hệ thống xe tự động có khả năng di chuyển ổn định và linh hoạt trong môi trường đa dạng.
- Cảm Biến IMU:Sử dụng IMU để đo lường gia tốc và góc độ của xe. IMU sẽ cung cấp thông tin chính để theo dõi tình trạng chuyển động của xe.
- Giao thức Điều Khiển:Phát triển một giao thức điều khiển thông minh dựa trên dữ liệu từ IMU. Giao thức này sẽ xử lý thông tin đo được và truyền tải về tốc độ, hướng, và chuyển động của xe.

2 Đặc tả phần cứng đề xuất

2.1 Arduino Nano

Arduino Nano với vi mạch USB to UART CH340 là một phiên bản của bo mạch Arduino Nano sử dụng vi mạch CH340 để thực hiện chức năng chuyển đổi USB sang UART (Universal Asynchronous Receiver-Transmitter). Điều này giúp Arduino Nano kết nối dễ dàng với máy tính thông qua cổng USB để lập trình và tương tác.

Thông số kỹ thuật:

- Kích thước PCB: 18mm x 45mm
- Trọng lượng: 7g
- IC chính: ATmega328P-AU
- IC nạp và giao tiếp UART: CH340
- IC ổn áp tuyến tính LM1117 tạo nên điện áp 5V
- Điện áp cấp: 5VDC cổng USB hoặc 6-9VDC chân Raw
- Số chân Digital: 14 chân, trong đó có 6 chân PWM
- Số chân Analog: 8 chân (hơn Arduino Uno 2 chân)

- Clock Speed: 16Mhz
- Bộ nhớ Flash: 32KB (Bộ tải khởi động 2KB).

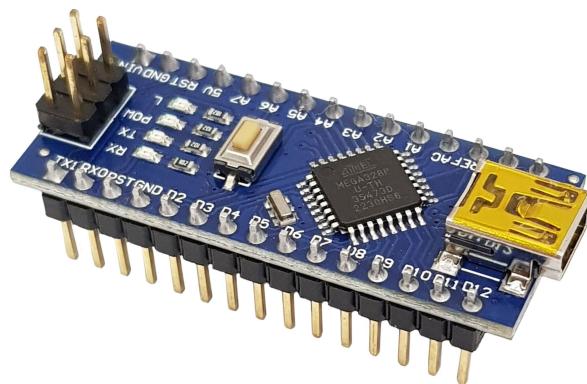


Figure 1: Arduino nano

2.2 NRF24L01 + RF Module

RF module NRF24L01+ là một mô-đun truyền thông không dây sử dụng công nghệ radio frequency (RF) ở tần số 2.4 GHz. Đây là một mô-đun rất phổ biến trong cộng đồng điện tử và IoT (Internet of Things) vì khả năng truyền thông không dây và linh hoạt trong việc kết nối các thiết bị.

Thông số:

- Kích thước: 15.24mm x 29.21mm
- Điện áp hoạt động: 1.9V - 3.6V
- Giao thức truyền: SPI (Serial Peripheral Interface)
- Tần số: 2.4GHz (dải tần số 2.4Ghz - 2.525GHz)
- Tốc độ truyền (Data Rate): 250kbps tới 2Mbps
- Công suất truyền (Transmit Power): 0 dBm (1 mW), -6 dBm, -12 dBm, -18 dBm
- Dòng tiêu thụ: Khoảng 11.3 mA khi truyền dữ liệu
- Độ nhạy thu (Receiver Sensitivity): -94 dBm at 250 kbps, -85 dBm at 2 Mbps, -82 dBm at 1 Mbps
- Dòng tiêu thụ: Khoảng 13.5 mA khi nhận dữ liệu, Chế độ chờ tiêu thụ ít hơn 1 μ A

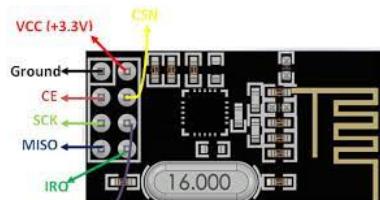


Figure 2: PIN of NRF24L01



Figure 3: RF NANO

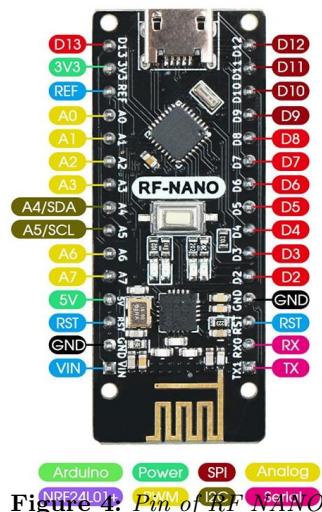


Figure 4: Pin of RF NANO

2.3 L298N Motor Driver

L298 Motor Driver là một bộ khuếch đại và điều khiển motor đôi (Dual H-Bridge Motor Driver). Bộ điều khiển này cung cấp khả năng điều khiển hướng và tốc độ của động cơ DC.

Điều khiển motor đôi (Dual H-Bridge Motor Driver) là một loại mạch điều khiển được sử dụng để điều khiển động cơ đối với hai chiều cảng (động cơ đôi)

Một dual H-bridge motor driver thường bao gồm hai cầu H độc lập (H-bridge), mỗi cái được thiết kế để điều khiển một động cơ. Mỗi cầu H bao gồm bốn công tắc có thể bật/tắt (transistor hoặc MOSFET), sắp xếp theo dạng hình chữ "H". Các công tắc này cho phép nguồn điện được kết nối với động cơ theo cả hai chiều, cho phép điều khiển hướng quay và tốc độ của động cơ.

- Kích thước: 3,4cm x 4,3cm x 2,7cm
- Chip điều khiển: Cầu đôi H L298N
- Điện áp cung cấp động cơ (Tối đa): 46V
- Dòng cung cấp động cơ (Tối đa): 2A
- Điện áp logic: 5V
- Điện áp trình điều khiển: 5-35V
- Trình điều khiển hiện tại: 2A
- Dòng điện logic: 0-36mA
- Công suất tối đa (W): 25W
- Nguồn cung cấp đầu ra được điều chỉnh +5V trên bo mạch (cung cấp cho bảng điều khiển, tức là Arduino)

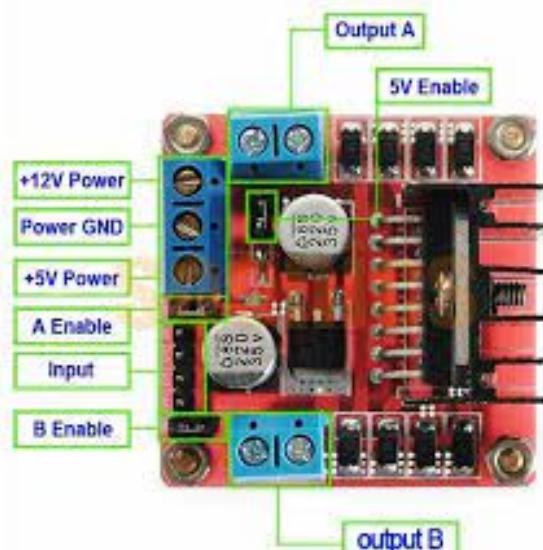


Figure 5: L298N Motor Driver

2.4 ADXL335 Module

Cảm Biến GY-61 Analog Accelerometer ADXL335 được sử dụng để đo gia tốc hướng (Accelerometer) hoặc độ rung động (tilt) theo 3 trục x, y, z và trả ra giá trị điện áp Analog tương ứng trên 3 chân của cảm biến nên có thể dễ dàng quan sát bằng máy đo hoặc đọc bằng các chân Analog của Vi điều khiển.

- Dải đo gia tốc: ± 3 g (góc độ) - Nghĩa là cảm biến có thể đo gia tốc từ -3 g đến $+3$ g theo các trục X, Y và Z
- Độ nhạy: 300 mV/g - Nghĩa là mỗi 1 g gia tốc sẽ tạo ra một biến đổi điện áp tương ứng là 300 mV
- Điện áp: $1.8V$ - $3.6V$
- Dòng tiêu thụ: Khoảng 320 μA khi cảm biến hoạt động
- Giao tiếp: Analog Output - Dữ liệu đầu ra ở dạng tín hiệu analog



Figure 6: ADXL335

2.5 TT Gear Motor

- Điện áp hoạt động : $3-9VDC$
- Dòng điện tiêu thụ: $110-140mA$
- Tỉ số truyền 1:48:
 125 vòng/ 1 phút tại $3VDC$.
 208 vòng/ 1 phút tại $5VDC$.
Moment: $0.5KG.CM$



Figure 7: TT Gear Motor

2.6 18650 li-ion battery

- Kích thước: 18x65mm
- Kiểu pin: 18650 Li-ion rechargeable battery.
- Điện áp trung bình 3.7VDC, sạc đầy 4.2VDC.
- Dung lượng: 1200mAh.
- Dòng xả tối đa liên tục: Max 2.4A (2C x 1200mAh).
- Nộp trớ trung bình: khoảng 40mΩ
- Dòng pin này có thể sạc lại



Figure 8: Pin 18650



3 Cảm biến gia tốc IMU

3.1 Cấu hình chân cảm

Như đã đặc tả sơ bộ ở phần trên, Gia tốc kế ADXL 335 là thiết bị đo gia tốc của vật thể bất kì. Đo giá trị gia tốc có dạng tín hiệu đầu vào analog, theo ba chiều X, Y và Z. Là thiết bị có độ nhiễu thấp và ít tiêu thụ điện năng.

Gia tốc kế ADXL 335 có năm chân được sử dụng cho các chức năng khác nhau. Cấu hình chân được hiển thị trong bảng dưới đây và các kết nối dây với bộ điều khiển bất kì theo bảng này.

chân cảm	chức năng
1	Là chân VCC được sử dụng để cấp nguồn cho gia tốc kế ADXL 335. Được kết nối với nguồn điện DC 3,3V
2	Là chân nối đất được sử dụng để cấp mass cho gia tốc kế ADXL 335. Được kết nối với mass nguồn.
3	Là chân X đầu vào analog của kích thước trực X. Chân này cấp tín hiệu đầu vào analog cho bộ điều khiển được đo bởi gia tốc kế ADXL 335
4	Là chân Y đầu vào analog của kích thước trực Y. Chân này cấp tín hiệu đầu vào analog cho bộ điều khiển được đo bởi gia tốc kế ADXL 335.
5	Là chân Z đầu vào analog của kích thước trực Z. Chân này cấp tín hiệu đầu vào analog cho bộ điều khiển được đo bởi gia tốc kế ADXL 335.

3.2 Nguyên lý hoạt động

Hiện nay trên thị trường có nhiều loại gia tốc kế khác nhau được sử dụng cho các mục đích khác nhau. Một số hoạt động theo nguyên lý MEMS (Micro electro mechanical sensor).

Có một mass nhỏ được chạm khắc vào bề mặt silicon và sau đó tích hợp vào một bảng mạch nhỏ. Khi lực tác dụng lên mass này sẽ di chuyển, do đó tạo ra gia tốc theo định luật chuyển động thứ hai của newton $F = ma$ được cảm biến.

Tương tự, nếu nói về gia tốc kế analog thì chúng hoạt động dựa trên hai nguyên lý cảm biến điện dung và cảm biến điện áp. Cả hai đều có những ưu và nhược điểm khác nhau. Gia tốc kế ADXL335 là một gia tốc kế analog do đó hoạt động theo nguyên lý cảm biến điện dung.

Trong gia tốc kế cảm biến điện dung, Khi nó được di chuyển theo một hướng bất kỳ thì điện dung của nó sẽ thay đổi. Khi điện dung thay đổi thì các điện áp analog sẽ thay đổi và được đo bởi bộ điều khiển giao tiếp với cảm biến.

Để có thu được tín hiệu từ gia tốc kế ADXL 335, cần giao tiếp với bộ điều khiển. Có thể giao

tiếp với bất kỳ loại bộ điều khiển nào nhưng ở đây sẽ đề cập đến cách giao tiếp bảng mạch Arduino với gia tốc kế ADXL 335. Hãy thực hiện kết nối dây theo sơ đồ hình 2 dưới đây. Theo

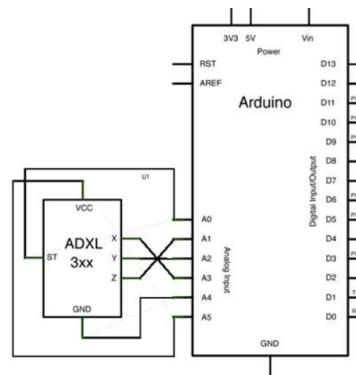


Figure 9: Ví dụ chân cắm về gia tốc kế cho mạch chính

sơ đồ hình trên, các chân X, Y và Z của gia tốc kế được kết nối với các chân A1, A2 và A3 của bảng mạch Arduino. Đây là các chân đầu vào analog cấp tín hiệu đầu vào analog cho bảng mạch Arduino.

Tương tự, các chân Vcc và Gnd của gia tốc kế được kết nối với các chân A5 và A4 của bảng mạch Arduino. Sau khi thực hiện tất cả các kết nối dây, viết một chương trình logic với thư viện Arduino và tải lên bảng mạch Arduino qua phần mềm Arduino IDE. Sau đó, đọc giá trị ba trục từ cảm biến bằng bảng mạch Arduino.

4 Giao thức sử dụng

4.1 Khái niệm giao thức RF

nRF24L01+ module sử dụng và truyền nhận dữ liệu trên một tần số cụ thể được biết đến là một kênh. Đối với hai hoặc nhiều module để giao tiếp với nhau, chúng phải ở trên cùng một kênh. Kênh này có thể có bất kỳ tần số nào trong dải ISM 2,4 GHz, hoặc cụ thể hơn, bất kỳ tần số nào từ 2,400 đến 2,525 GHz (2400 đến 2525 MHz).

Mỗi kênh chiếm ít hơn 1 MHz băng thông. Điều này mang lại cho chúng ta 125 kênh khả dụng với khoảng cách 1 MHz. Điều này có nghĩa là nRF24L01+ có thể hoạt động trên 125 kênh khác nhau, cho phép xây dựng một mạng lưới gồm 125 modem độc lập vận hành tại một địa điểm.

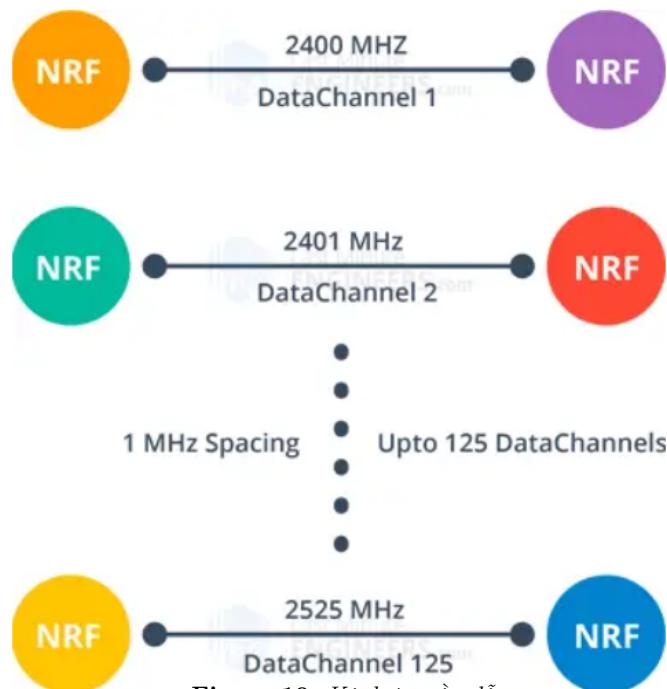


Figure 10: Kênh truyền dẫn

Ở tốc độ truyền dữ liệu không khí 250kbps và 1Mbps, mỗi kênh chiếm ít hơn 1 MHz băng thông, vì vậy có một khoảng trống 1 MHz giữa hai kênh. Tuy nhiên, đối với tốc độ truyền dữ liệu không khí 2Mbps, yêu cầu 2MHz băng thông (lớn hơn độ phân giải của cài đặt tần số kênh RF). Do đó, ở chế độ 2 Mbps, giữ một khoảng trống 2MHz giữa hai kênh để đảm bảo các kênh không chồng lấn và giảm hiện tượng cross-talk.

Tần số kênh RF của kênh bạn chọn được tính bằng công thức sau:

$$\text{Freq} = 2400 + \text{ChannelSelect}$$

4.2 Gói tin truyền dẫn

Cấu trúc gốc của Shockburst chỉ bao gồm các trường Preamble, Address, Payload và Cyclic Redundancy Check (CRC). Bằng cách giới thiệu trường Packet Control Field (PCF), Enhanced Shockburst đã thêm chức năng cho giao tiếp tiên tiến hơn.



Figure 11: Cấu trúc gói tin

- Nó hỗ trợ các payload có độ dài biến với một bộ chỉ định độ dài payload, cho phép payload có độ dài từ 1 đến 32 byte.
- Mỗi gói tin đã gửi được gán một Packet ID, giúp bộ nhận xác định xem tin nhắn có phải là mới hay đã được gửi lại.
- Mỗi tin nhắn chứa một trường yêu cầu bộ nhận gửi một ACK.

4.3 Kiểm soát gói tin

Truyền tải với một ACK:

Đây là một ví dụ về kịch bản tốt nhất. Trong trường hợp này, bộ truyền bắt đầu giao tiếp bằng cách gửi một gói dữ liệu đến bộ nhận. Sau khi gửi gói tin, bộ truyền đợi khoảng 130 µs để xác nhận (ACK) đến. Bộ nhận gửi ACK sau khi nhận gói tin thành công. Khi bộ truyền nhận được ACK, quá trình truyền tải kết thúc.

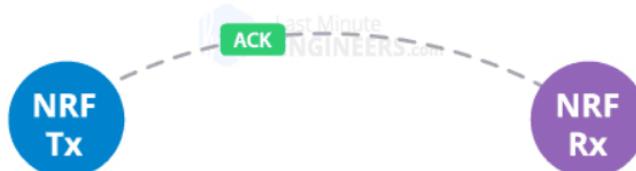


Figure 12: Kiểm soát truyền tải

Truyền tải với một gói dữ liệu bị mất:

Đây là một kịch bản tiêu cực trong đó cần phải thực hiện việc gửi lại do mất gói tin. Bộ truyền đợi xác nhận sau khi gửi gói tin.

Nếu bộ truyền không nhận được nó trong thời gian trễ gửi lại tự động (ARD), gói tin sẽ được gửi lại. Khi bộ nhận nhận được gói tin được gửi lại, nó sẽ gửi ACK, quá trình truyền tải kết thúc.



Truyền tải với với điều kiện ACK bị mất:

Đây là một tình huống tiêu cực khác trong đó cần thực hiện việc gửi lại do mất ACK. Vì bộ truyền không nhận được ACK, nó tin rằng gói tin đã bị mất (mặc dù bộ nhận đã nhận gói tin trong lần thử đầu tiên).

Do đó, bộ truyền gửi lại gói tin sau khi hết thời gian chờ tự động (Auto-Retransmit-Delay). Khi bộ nhận nhận được một gói tin có cùng ID như trước đó, nó sẽ từ chối nó và gửi lại ACK. Khi bộ truyền nhận được ACK, quá trình truyền tải kết thúc.

Trong bối cảnh sử dụng cảm biến IMU truyền tải dữ liệu thông qua sóng radio RF, quá trình truyền tải sẽ gửi gói tin liên tục trong khoảng 10ms, nên sẽ không xét tới việc ACK bị mất hay không.

5 Thiết kế và hiện thực

5.1 Thiết kế hệ thống

Để tổng quát hóa phần hiện thực, từ sơ đồ finite state machine sẽ thể hiện các hành vi của hệ thống: Để làm rõ hơn về các mã nguồn chức năng của từng bên phát(transmitter) và bên

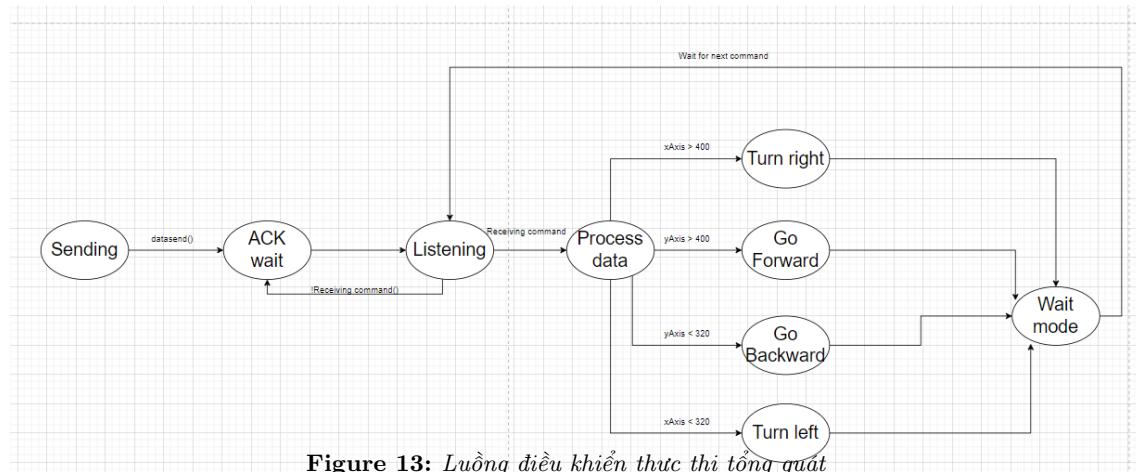


Figure 13: Luồng điều khiển thực thi tổng quát

nhận(receiver), sẽ được đặc tả bởi sơ đồ flowchart ở các phần bên dưới:

5.2 Transmitter

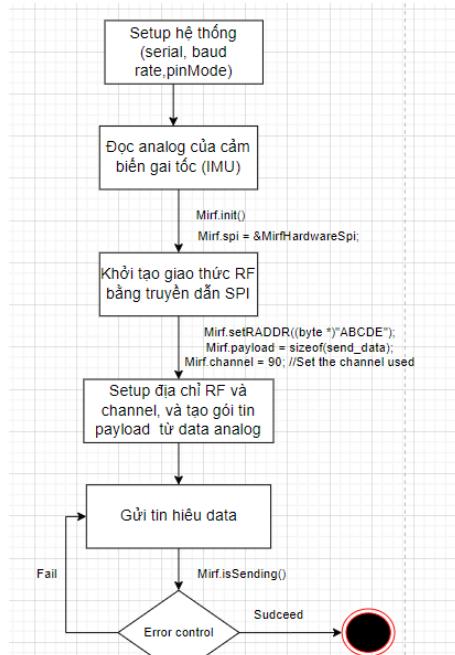


Figure 14: Luồng điều khiển thực thi bên gửi

5.3 Receiver

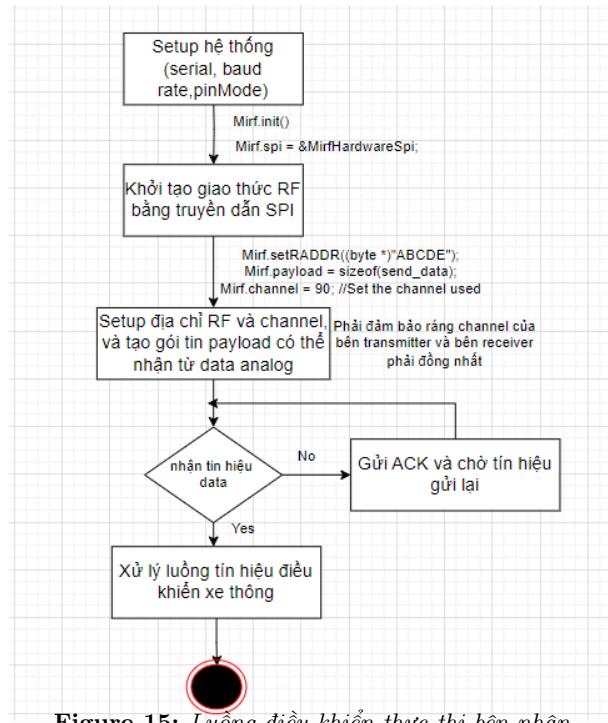
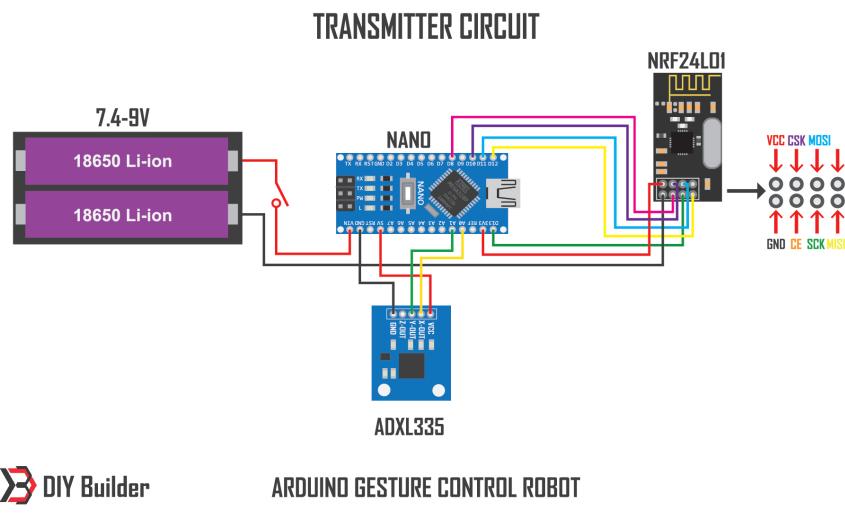


Figure 15: Luồng điều khiển thực thi bên nhận

5.4 Hiện thực phần cứng

Từ các thông số ở mục phần cứng đề xuất, nhóm thực hiện cơ khí và lắp ráp dựa trên mô hình xe thông theo sơ đồ như sau:

5.4.1 Transmitter

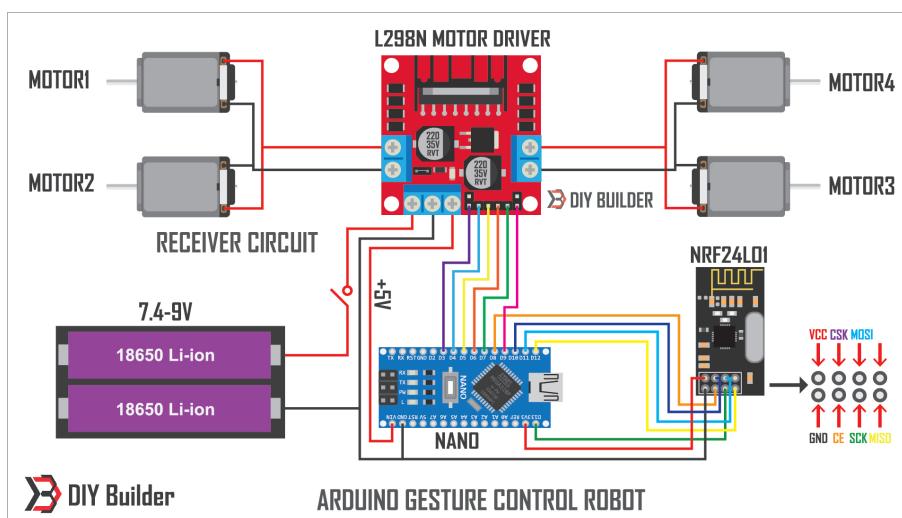


DIY Builder

ARDUINO GESTURE CONTROL ROBOT

Figure 16: Mô hình phần cứng bên gửi

5.4.2 Receiver



DIY Builder

ARDUINO GESTURE CONTROL ROBOT

Figure 17: Mô hình phần cứng bên nhận

5.4.3 Sản phẩm hoàn chỉnh

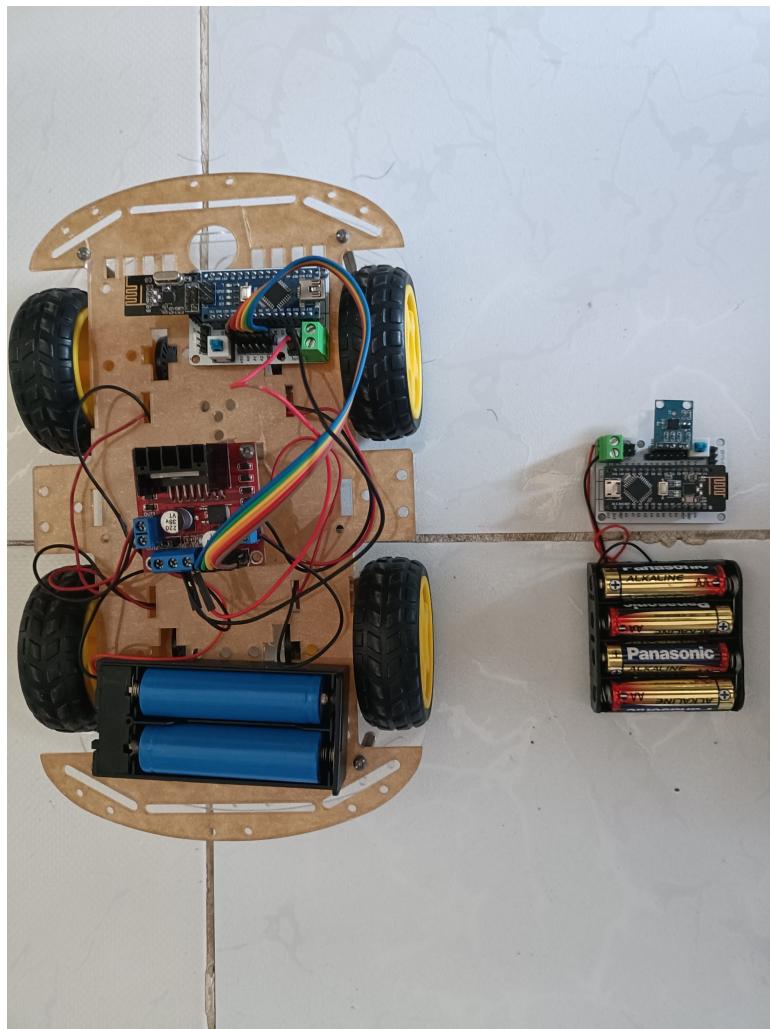


Figure 18: Hình ảnh xe và bộ điều khiển

6 Mã nguồn và demo hiện thực

Mã nguồn thực hiện của nhóm được trình bày trong repo link github dưới đây:
https://github.com/quangto188/DoAnTKLL_231.git



List of Figures

1	Arduino nano	3
2	PIN of NRF24L01	4
3	RF NANO	4
4	Pin of RF NANO	4
5	L298N Motor Driver	5
6	ADXL335	6
7	TT Gear Motor	7
8	Pin 18650	7
9	Ví dụ chân cắm về giá tốc kế cho mạch chính	9
10	Kênh truyền dẫn	10
11	Cấu trúc gói tin	11
12	Kiểm soát truyền tải	11
13	Luồng điều khiển thực thi tổng quát	13
14	Luồng điều khiển thực thi bên gửi	13
15	Luồng điều khiển thực thi bên nhận	14
16	Mô hình phần cứng bên gửi	15
17	Mô hình phần cứng bên nhận	15
18	Hình ảnh xe và bộ điều khiển	16



7 References

- [1] [How To Make DIY Arduino Gesture Control Robot At Home](#)
- [2] [Nơi tham khảo và mua các phân cứng đề xuất](#)
- [3] [How nRF24L01+ Wireless Module Works and Interface with Arduino](#)
- [4] [What is an Accelerometer and Interfacing the ADXL335 accelerometer with Arduino](#)