 **BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----🙞🙜🕮🙞🙜-----

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**ĐỀ TÀI: “THIẾT BỊ GIẢI TRÍ VR”**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Giảng viên hướng dẫn:** | **PHẠM VĂN SỰ** | | **Sinh viên thực hiện:** | **VŨ HẢI LONG** | | **Lớp:** | **D15DTMT1** | | **Hệ:** | **ĐẠI HỌC CHÍNH QUY** | |  |
|  |  |
|  |  |
| **Hà Nội - 2019** |  |
|  | |

# LỜI CẢM ƠN

Trong quá trình nghiên cứu và thực hiện đồ án em đã nhận được sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Đầu tiên, em xin được gửi lời cảm ơn đến Ban Giám hiệu Học viện Công nghệ Bưu chính Viễn thông đã tạo cho em môi trường rèn luyện tốt để em có thể học tập và tiếp thu được những kiến thức quý báu trong những năm qua.

Em xin cảm ơn tất cả các thầy cô giáo, đặc biệt là các thầy cô trong khoa Kỹ thuật điện tử 1 đã tận tình chỉ dạy những kiến thức quý báu để em có thể hoàn thành được đồ án cũng như những hành trang cần thiết để em có thể bước trên con đường sự nghiệp sau này.

Em xin được gửi lời cảm ơn đến thầy Phạm Văn Sự đã hướng dẫn em thực hiện đồ án **“Thiết bị giải trí VR”.** Mặc dù công việc rất bận rộn nhưng thầy vẫn luôn dành thời gian hướng dẫn chỉ bảo tận tình để em có thể hoàn thành tốt đồ án.

Mặc dù đã cố gắng hết sức song không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và chỉ bảo tận tình của quý thầy cô và các bạn để em có thể hoàn thành tốt hơn báo cáo này.

Cuối cùng em xin kính chúc các thầy, các cô, gia đình và bạn bè dồi dào sức khỏe và thành công trong sự nghiệp.

# 

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc28177285)

[MỤC LỤC 1](#_Toc28177286)

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT 3](#_Toc28177287)

[DANH MỤC HÌNH ẢNH 4](#_Toc28177288)

[LỜI MỞ ĐẦU 1](#_Toc28177289)

[CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN 1](#_Toc28177290)

[1.1. Công nghệ VR và Ứng dụng 1](#_Toc28177291)

[1.1.1. VR(Virtual reality) là gì? 1](#_Toc28177292)

[1.2 Cấu tạo thành phần và cách hoạt động trong trò chơi điện tử. 4](#_Toc28177293)

[1.2.1 Kính 3D 5](#_Toc28177294)

[1.2.2 Thiết bị sử dụng tương tác môi trường ảo (controler) 5](#_Toc28177295)

[1.2.3 Cách Kính 3D kết hợp thiết bị tương tác ảo 6](#_Toc28177296)

[1.3. Giới thiệu về ý tưởng nghiên cứu và thiết kế sản phẩm 7](#_Toc28177297)

[1.3.1. Những ý tưởng của đề tài 7](#_Toc28177298)

[1.3.2. Những điểm mới của sản phẩm trong đề tài 8](#_Toc28177299)

[1.4. Kết luận chương 8](#_Toc28177300)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI 9](#_Toc28177301)

[2.1 Giới thiệu các thông số về góc : pitch , yaw , roll 9](#_Toc28177302)

[2.2 Phần thu nhận và truyền dữ liệu 12](#_Toc28177303)

[2.2.1 Giới thiệu về cảm biến MPU6050 12](#_Toc28177304)

[2.2.2 Giới thiệu về Arduino, ESP32 và joystick ( nút nhấn đa hướng) 19](#_Toc28177305)

[2.2.3 Các giao tiếp giữa cảm biến với vi điều khiển. 25](#_Toc28177306)

[2.2.4 Một số giao tiếp sử dụng trong đề tài. 25](#_Toc28177307)

[2.3. Giới thiệu hệ thống và cách tương tác với trình quản lý thiết bị trên HĐH Linux ( driver ) 28](#_Toc28177308)

[2.3.1. Giới thiệu chung 28](#_Toc28177309)

[CHƯƠNG III: THIẾT KẾ SẢN PHẨM 33](#_Toc28177310)

[3.1. Sơ đồ khối hệ thống 33](#_Toc28177311)

[3.1.1. Sơ đồ khối thu và truyền dữ liệu cảm biến 34](#_Toc28177312)

[3.1.2. Sơ đồ khối nhận và xử lý dữ liệu cảm biến 34](#_Toc28177313)

[3.2. Thiết kế phần cứng 35](#_Toc28177314)

[3.2.1. Mạch kết hợp Arduino, MPU6050 và ESP32 35](#_Toc28177315)

[3.3. Thiết kế phần mềm 36](#_Toc28177316)

[3.3.1. Các bước thiết lập hệ thống 36](#_Toc28177317)

[3.3.2. Thiết kế thành phẩn điều khiển trung tâm 39](#_Toc28177318)

[3.3.3. Thiết kế khung dữ liệu truyền 39](#_Toc28177319)

[3.4 Thử nghiệm và đánh giá 40](#_Toc28177320)

[3.4.1 Sản phẩm thử nghiệm 40](#_Toc28177321)

[3.4.2 Đánh giá 40](#_Toc28177322)

[3.5 Kết luận chương 41](#_Toc28177323)

[KẾT LUẬN 41](#_Toc28177324)

[TÀI LIỆU THAM KHẢO 42](#_Toc28177325)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Viết đầy đủ** | **Giải nghĩa** |
| ADC | Analog-to-Digital Converter | Bộ chuyển đổi tương tự sang số |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| DMP | Digital Motion Processor | Khối xử lý chuyển động |
| GPIO | General Purpose Input Output | Cổng kết nối nhập xuất |
| IOT | Internet of Things | Internet kết nối vạn vật |
| I2C | Inter-Integrated Circuit | giao tiếp theo dạng chủ tớ dựa trên 2 dây truyền |
| HĐH | Hệ Điều Hành | Hệ Điều Hành |
| MQTT | Message Queuing Telemetry Transport | Truyền thông tin tuần tự từ xa |
| UART | [Universal Asynchronous Receiver/Transmitter](http://tratu.vietgle.vn/hoc-tieng-anh/tu-dien/lac-viet/all/Universal+Asynchronous+Receiver-slash-2-Transmitter+-+UART.html) | Truyền thông nối tiếp không đồng bộ |
| USB | Universal Serial Bus | Chuẩn kết nối USB |
| VĐK | Vi Điều Khiển | Vi Điều Khiển |
| VR | Virtual Reality | Thực tế ảo |
| 3D | 3 Dimensional | Không gian 3 chiều |

# 

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 Ứng dụng AR 3](#_Toc27677278)

[Hình 1. 2 Ứng dụng của VR trong đào tạo 3](#_Toc27677279)

[Hình 1. 3 Ứng dụng VR trong đào tạo 4](#_Toc27677280)

[Hình 1. 4 Ứng dụng VR trong đào tạo lái xe 5](#_Toc27677281)

[Hình 1. 5 Mô hình hoạt động cơ bản của hệ thống VR 5](#_Toc27677282)

[Hình 1. 6 Kính 3D 6](#_Toc27677283)

[Hình 1. 7 thiết bị điện tử Wii 7](#_Toc27677284)

[Hình 1. 8 Thiết bị giải trí tích hợp cảm biến chuyển động 7](#_Toc27677285)

[Hình 1. 9 Kiến trúc hoạt động cơ bản của hệ thống VR 8](#_Toc27677286)

[Hình 1. 10 Sản phẩm Nintendo switch 8](#_Toc27677287)

[Hình 2. 1 Mô tả ba thông số về góc 10](#_Toc27677302)

[Hình 2. 2 Vector Quaternion trong không gian 3 chiều xyz 11](#_Toc27677303)

[Hình 2. 3 Mã nguồn triển khai theo công thức. 12](#_Toc27677304)

[Hình 2. 4 Mô hình hệ thống đối phần thu dữ liệu 13](#_Toc27677305)

[Hình 2. 5 Hình ảnh MPU6050 13](#_Toc27677306)

[Hình 2. 6 Hiệu ứng Coriolis 15](#_Toc27677307)

[Hình 2. 7 Hiệu ứng Coriolis lên vật Mass 15](#_Toc27677308)

[Hình 2. 8 Cấu trúc của một cảm biến gyroscope 16](#_Toc27677309)

[Hình 2. 9 Thông số các chế độ của MPU6050 16](#_Toc27677310)

[Hình 2. 10 Nguyên lý vật lý được sử dụng trong đo gia tốc. 17](#_Toc27677311)

[Hình 2. 11 Cấu trúc của một cảm biến đo góc nghiêng 18](#_Toc27677312)

[Hình 2. 12 Thông số các chế độ của cảm biến đo góc nghiêng trên MPU6050 18](#_Toc27677313)

[Hình 2. 13 Tổng quan về các khối trong MPU6050 19](#_Toc27677314)

[Hình 2. 14 Hình ảnh kit Arduino 20](#_Toc27677315)

[Hình 2. 15 Cơ chế hoạt động của Arduino 21](#_Toc27677316)

[Hình 2. 16 Kit ESP32 21](#_Toc27677317)

[Hình 2. 17 Mô hình hoạt động của MQTT trong sản phẩm. 23](#_Toc27677318)

[Hình 2. 18 Hình ảnh Joystick 24](#_Toc27677319)

[Hình 2. 19 Mạch chuyển đổi logic 2 chiều 3.3v-5v 25](#_Toc27677320)

[Hình 2. 20 Mô hình kết nối I2C 27](#_Toc27677321)

[Hình 2. 21 Giao diện kết nối UART 28](#_Toc27677322)

[Hình 2. 22 Kiến trúc của hệ thống Linux 29](#_Toc27677323)

[Hình 2. 23 Các file trong thư mục /dev 31](#_Toc27677324)

[Hình 2. 24 Mô hình chi tiết hoạt động của MQTT trong sản phẩm 32](#_Toc27677325)

# LỜI MỞ ĐẦU

Trong quá trình phát triển của con người, những cuộc cách mạng về công nghệ đóng một vai trò rất quan trọng, chúng làm thay đổi từng ngày từng giờ cuộc sống của con người, theo hướng hiện đại hơn. Đi đôi với quá trình phát triển của con người, những thay đổi do chính tác động của con người trong tự nhiên, trong môi trường sống cũng sự phát triển như vũ bão của công nghệ và Internet, người tiêu dùng ngày nay đang thay đổi cả về nhận thức, suy nghĩ lẫn hành vi.  Các báo cáo bán lẻ gần đây cho biết hành vi tiêu dùng của người Việt Nam đang thay đổi nhanh chóng.

Đặc biệt là áp dụng các công nghệ của các ngành điện tử, công nghệ thông tin và truyền thông vào trong thực tiễn cuộc sống con người. Thay vì tới các trung tâm thương mại để mua sắm thuần túy, người dùng tới để trải nghiệm dịch vụ, công nghệ mới, điều này đang dần trở thành trào lưu . Chính những điều này đã tạo nên cho em ý tưởng để thiết kế một thiết bị có thể tương tác thật hơn đối với trải nghiệm người dùng . Do đó, em mong muốn thực hiện đề tài “Thiết bị giải trí VR” để qua đó nắm bắt và làm chủ được các công nghệ đã đề cập ở trên.

Nội dung đồ án bao gồm 3 chương:

**Chương 1: Nghiên cứu tổng quan.**

**Chương 2: Cơ sở lý thuyết của đề tài.**

**Chương 3: Thiết kế sản phầm.**

Em xin chân thành cảm ơn thầy Phạm Văn Sự cùng toàn thể các thầy cô khoa Kỹ thuật điện tử I đã tạo điều kiện và giúp đỡ em trong quá trình thực hiện đồ án này.

Dưới đây em xin trình bày chi tiết các phần trong nội dung của đồ án.

# 

# CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN

Chương này sẽ giúp người đọc có được một cái nhìn tổng quát về ý tưởng nghiên cứu đề tài. Các nội dung sẽ được trình bày trong chương này:

* **Công nghệ VR và Ứng dụng.**
* **Cấu tạo thành phần và cách hoạt động.**
* **Giới thiệu về ý tưởng nghiên cứu và thiết kế sản phẩm.**

Dưới đây em xin trình bày các nội dung cụ thể của chương.

## Công nghệ VR và Ứng dụng

### VR(Virtual reality) là gì?

Công nghệ VR đang dần phổ biến và được áp dụng trong nhiều lĩnh vực. Vậy VR là gì và xu hướng này nghĩa như thế nào?

VR (Virtual Reality) hay được gọi là Thực tế ảo, công nghệ này sẽ đưa bạn vào một thế giới hoàn toàn ảo do máy tính tạo ra. Nó chuyển bạn từ môi trường với những đồ vật có thật ở xung quanh sang một môi trường ảo, nơi mà bạn thực sự trở thành một phần của nó và tương tác với nó theo những cách khác nhau. Bên cạnh việc tạo ra cho người dùng các trải nghiệm về hình ảnh ảo, công nghệ VR còn tương tác với người qua những giác quan khác như thính giác và xúc giác.

Công nghệ thực tế ảo đã có từ những năm 1990. Lúc đó, người người nhà nhà trên khắp thế giới nói về việc làm ra những sản phẩm VR lớn nhưng hầu hết đều thất bại vì nhiều lý do: sức xử lý của máy tính chưa đủ mạnh, cộng đồng chưa nhiều, chi phí đắt đỏ, trải nghiệm chưa tốt (đây là lý do lớn nhất).

Đặc tính của thực tế ảo đó là sự hòa nhập. Thuật ngữ này mô tả cảm giác của bạn khi được đưa vào thế giới VR: bạn cảm thấy thế giới đó có thật không, bạn có thấy được hết những đối tượng trong đó hay không, bạn có cảm thấy như mình đang sống trong một không gian hoàn toàn mới hay không. Sự hòa nhập này một phần đến từ việc kính thực tế ảo sẽ bao phủ hết tầm nhìn của mắt nên bạn sẽ không thấy gì ngoài đời thật cả .

VR có tính gần gũi cao với game về mặt tính chất, thường dùng trong công việc mô phỏng (simulation) hay giải trí.



Hình 1. Ứng dụng AR

**Giáo Dục 4.0 – thực tế ảo trong giáo dục:** ứng dụng thực tế ảo trong đào tạo,dạy nghề, huấn luyện với Giáo dục (AR education) là một đột phá về phương pháp giảng dạy, cách thức tiếp cận mới cho người học và người dạy, thay vì học chay, lý thuyết xuông sẽ chuyển sang học thực hành,trải nghiệm thực tế qua mô phỏng 3D, các phòng LAB, nên các bài học, kiến thức sẽ thực tế, chi tiết, nhanh hiểu, dễ nhớ thu hút người học, hiệu quả cao gấp nhiều lần so với cách cũ. Hãy tưởng tượng, làm thế nào những bài học sẽ thú vị hơn, sinh động, hiệu quả



Hình 1. Ứng dụng của VR trong đào tạo

Giáo viên sử dụng công nghệ AR để giải thích một số sự kiện lịch sử, các môn khoa học, sinh học. Việc biến bất kỳ sự kiện, kiến thức thụ động, nhàm chán trở thành sinh động, thú vị hơn hay thấy chúng hoạt động chi tiết thế nào… Trường học đang được số hóa và có thể sử dụng máy tính, ứng dụng di động cho các bài học của mình. Hãy suy nghĩ về những gì họ có thể làm với công nghệ VR.

Sinh viên có thể đi lang thang xung quanh bên trong một tế bào người hoặc nhìn thấy, tương tác với một thiên hà 3D trên màn hình điện thoại thông minh hoặc máy tính bảng của mình. Ứng dụng thực tế ảo vào giáo dục có thể nói là một trong những nội dung đang chú ý và đúng chủ trương chính sách và yêu cầu thực tế của xã hội hiện nay là đổi mới phương pháp giáo dục bằng các công nghệ mới nhất để đáp ứng hướng phát triển của giáo dục 4.0 tại việt nam trong thời gian tới

Ưu điểm và hiệu quả của thực tế ảo trong đào tạo

* Thực tế ảo giúp người học trải nghiệm thực tế, tương tác, dễ nhớ,hiệu quả.
* Học sinh có thế nhìn thấy các phần tử cấu tạo, nhập, tách thế nào.
* Khi học lịch sử sẽ trở nên sinh động hơn, dễ nhớ hơn …
* Môn khoa học sẽ nhìn thấy vũ trụ hiện ra trước mắt, có thể cảm nhận ,tương tác được.

**Ứng dụng thực tế ảo trong dạy nghề, huấn luyện công nhân**

* Giúp người học thực hành dễ dàng, thao tác nhiều lần, tiết kiệm kinh phí.
* Học nghề sửa chữa có thể thực hành luôn trên máy tính, điện thoại, không cần máy móc thật,tiết kiệm
* Khi tháo lắp máy móc không cần nhiều đến máy thật, mà hiệu quả và chính xác hơn nhiều, không bị che khuất.
* các chi tiết , từng bộ phận đã được số hóa 3D và dùng công nghệ thực tế ảo để họ có thể thực hành mọi lúc mọi nơi. Nhiều lần mà không tốn nhiều kinh phí, không phụ thuộc vào điều kiện thực địa để thực hành.



Hình 1. Ứng dụng VR trong đào tạo

**Ứng dụng trong học, cấp chứng chỉ giao thông :**Người dùng trực tiếp lái xe chip thi thử trong sa bàn với đầy đủ địa hình, độ khó và được chấm điểm như bài thi lái xe ngoài đời thực.

Ứng dụng VR tiên phong này là lựa chọn tốt nhưng ai cần thêm thời gian thực hành trước khi thi lấy bằng lái xe ô tô.

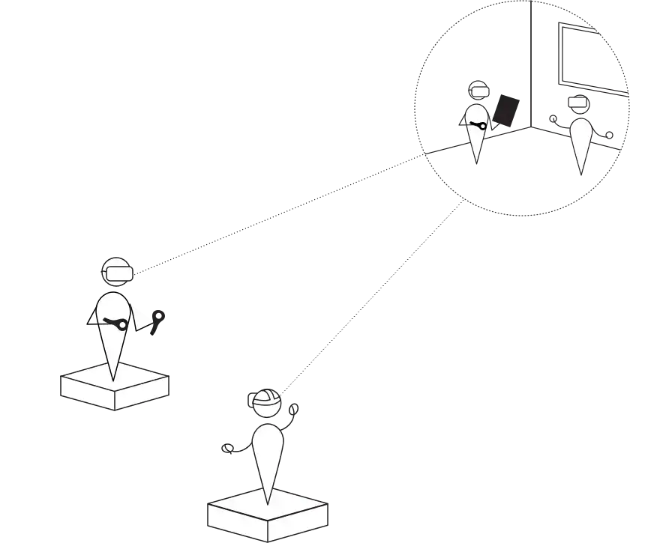


Hình 1. Ứng dụng VR trong đào tạo lái xe

## 1.2 Cấu tạo thành phần và cách hoạt động trong trò chơi điện tử.

Một **thiết bị giải trí có sử dụng công nghệ VR** sẽ bao gồm 2 thành phần chính là:

1. Kính 3D để đưa người sử dụng vào không gian ảo qua thị giác
2. Thiết bị điều khiển để tương tác trong không gian ảo đó



Hình 1. Mô hình hoạt động cơ bản của hệ thống VR

### Kính 3D



Hình 1. Kính 3D

Chiếc Kính 3D này sẽ làm nhiệm vụ đưa người sử dụng vào không gian ảo thông qua thị giác , cùng với sự kết hợp các cảm biến trên nó như cảm biến gia tốc , vận tốc góc để xác định hướng người dùng di chuyển mắt.

Các loại kính này sẽ hỗ trợ các ứng dụng trên Smartphone có hỗ trợ chế độ 3D như youtube , các ứng dụng game thực tế ,...

Trong phạm vi đề tài của mình em sẽ không đi sâu và sử dụng thiết bị này.

### 1.2.2 Thiết bị sử dụng tương tác môi trường ảo (controler)

Thiết bị này được trang bị một số cảm biến để đo hướng di chuyển và một số phím nhấn để tương tác . Thiết bị này cho người dùng cảm giác thực tế hơn so với các thiết bị điều khiển cổ điển như trước đây

Hầu hết các thiết bị tương tác ảo sẽ được tích hợp 2 loại cảm biến chính là :

* Cảm biến gia tốc 3 trục dùng để do tốc độ di chuyển trong không gian của thiết bị.
* Cảm biến con quay hồi chuyển 3 trục để đo góc nghiêng của thiết bị

Em sẽ trình bày chi tiết về các loại cảm biến này trong phần II.



Hình 1. thiết bị điện tử Wii



Hình 1. Thiết bị giải trí tích hợp cảm biến chuyển động

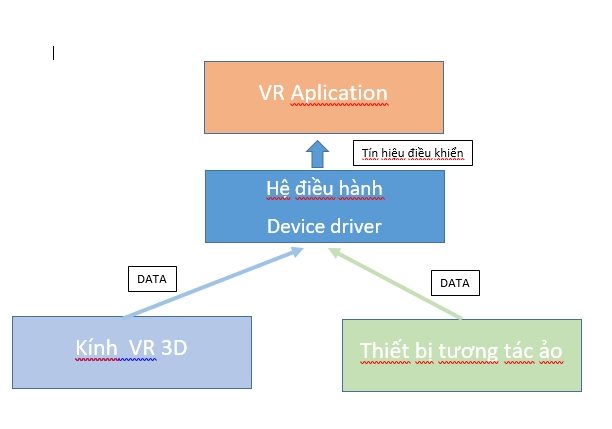
### Cách Kính 3D kết hợp thiết bị tương tác ảo

Một nền tảng OS chạy trên bộ xử lý trung tâm ( thường là Android) sẽ nhận dữ liệu thu thập được từ Kính 3D , thiết bị tương tác để quyết định hành vi xử lý tiếp theo lên trên phía VR application

Kính 3D sẽ thu xác định hướng nhìn của người sử dụng để có thể gửi thông tin về cho trung tâm xử lý . Đối với thiết bị tương tác trong trường hợp đơn giản người ta có thể chỉ cần đến xác định di chuyển 2D .

Việc VR application có thể hoạt động được thì cần có sự tương thích cả Kính 3D và thiết bị tương tác ảo. VR application sẽ đăng ký nhận thông tin từ hệ thống đối với hai loại thiết bị trên (trong hệ điều hành Linux thì chúng ta có thể biết đến như lớp device driver tạo device file , các ứng dụng sẽ nhận được thông tin tự hệ thống thông qua các file ảo này)

Trong phạm vi nghiên cứu đề tài của mình em sẽ đề cập chi tiết đến việc xây dựng và cách hoạt động của thiết bị tương tác ảo tương tác lên trên với một game application .



Hình 1. Kiến trúc hoạt động cơ bản của hệ thống VR

## 1.3. Giới thiệu về ý tưởng nghiên cứu và thiết kế sản phẩm

### 1.3.1. Những ý tưởng của đề tài

Dựa vào ý tưởng về các sản phẩm thực tế đang kinh doanh trên thị trường đối với loại sản phẩm là game controller như : Nintendo Switch, Wii, các thiết thị hỗ trợ điều khiển trong môi trường ảo VR,..



Hình 1. Sản phẩm Nintendo switch

Từ những thành công là lợi ích của sản phẩm mang lại cùng với những kiến thức em đã tích luy được trong quá trình học tập tại trường và thời gian đi thực tập nên em đã quyết định chọn và phát triển lên thành đề tài **“Thiết bị giải trí VR”.**

### 1.3.2. Những điểm mới của sản phẩm trong đề tài

Trong sản phầm của em đã có một số những điểm mới như sau :

Thiết kế đơn giản , giá rẻ nhưng vẫn đầy đủ tính năng để điều khiển một ứng dụng có liên kết sử dụng “chuột , bàn phím trên máy tính”

Có thể tùy biến dễ dàng các chức năng điều khiển trên thiết bị ( sản phẩm cho phép thay đổi các phím tắt để điều khiển ứng dụng)

Vì là ứng dụng trên máy tính ( yêu cầu sử dụng hệ điều hành Linux) nên nhiều ứng dụng giải trí khá đa dạng và không phải trả phí .

Sản phẩm hiện tại có thể phát triển để dễ dàng biến thành một con chuột trên không tương tác ảo đơn giản trên máy tính

## 1.4. Kết luận chương

Như vậy ở **Chương I** em đã trình bày được khái quát về ý tưởng, tổng quan công nghệ VR, các thành phần cấu tạo và chức năng của chúng để tiền đề cho việc xây dựng ý tưởng đề tài.

# 

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI

Chương này sẽ trình bày cơ sở lý thuyết và những kiến thức cơ bản về các công nghệ được sử dụng cho việc thiết kế sản phẩm.

Chương này gồm các vấn đề sau:

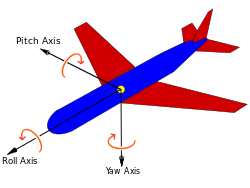
* **Giới thiệu các thông số về góc : pitch , yaw , roll**
* **Các thiết bị điều khiển và thu nhận dữ liệu.**
* **Giới thiệu hệ thống quản lý thiết bị trên hệ điều hành Linux.**
* **Cách sử dụng giao tiếp giữa hệ thống và ứng dụng trong thiết kế sản phẩm .**

Dưới đây em xin trình bày các nội dung cụ thể của chương:

## 2.1 Giới thiệu các thông số về góc : pitch , yaw , roll

Các thông số quan trọng mà em sẽ dung tới trong sản phầm đồ án lần này gồm :

Pitch , Yaw , Roll.



Hình 2. Mô tả ba thông số về góc

Ở đây cảm biến sẽ được coi là đặt trung tâm máy bay . Yaw đại diện cho góc xoay quanh trục thẳng đứng đối với mặt đất . Pitch là góc nghiêng thân máy bay so với trục là cánh . Roll là góc nghiêng khi coi thân máy bay làm trục và khi đó cánh máy bay sẽ là góc nghiêng . Việc tính toán cả 3 tham số trên em sẽ dựa trên dữ liệu từ cảm biến và áp dụng công thức toán học đã được chứng minh

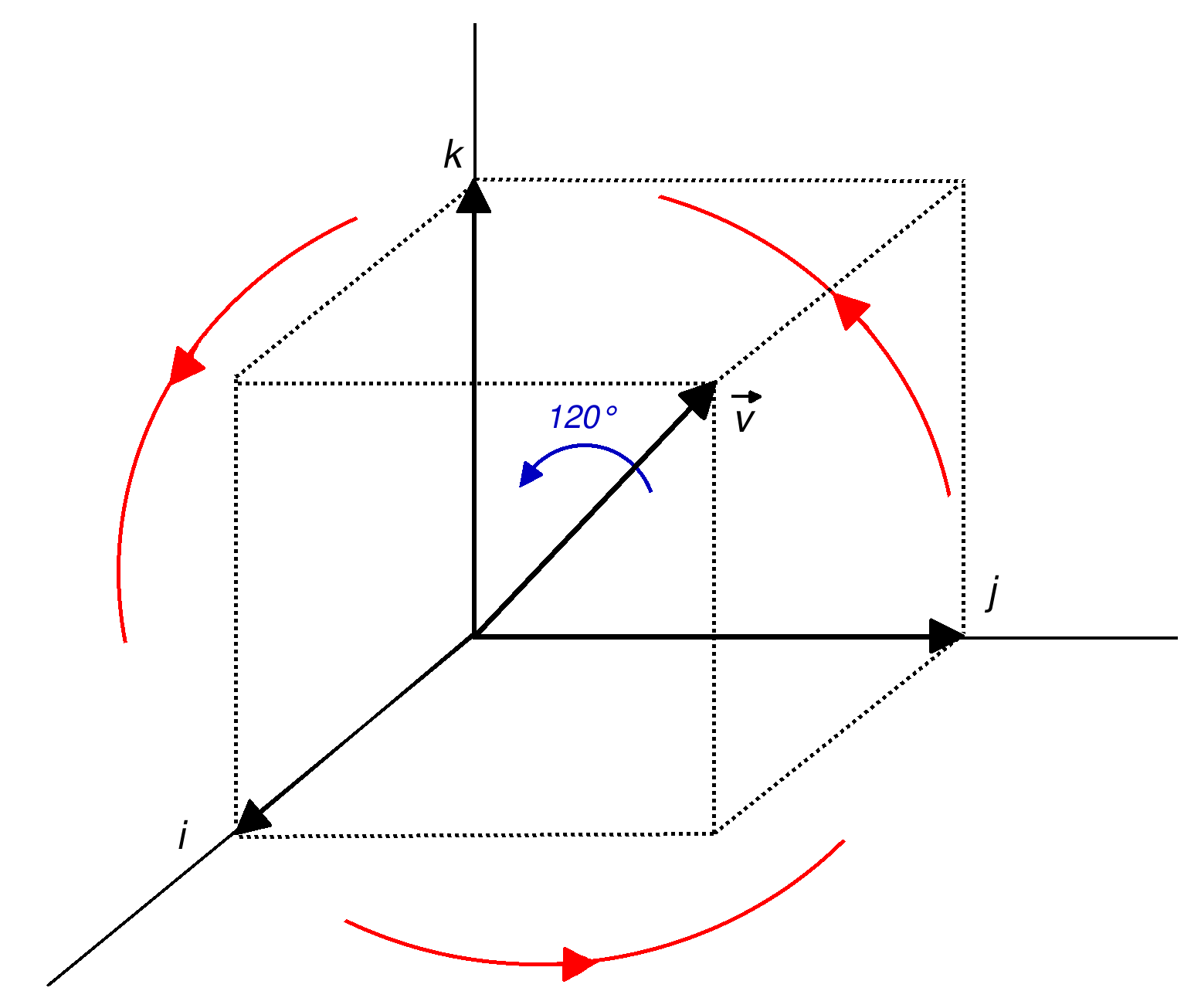
Trong đề tài của em sẽ sử dụng các thông số này để xác định được hướng di chuyển của thiết bị . Việc tính toán 3 tham số trên sẽ dựa trên hệ Quaterion để biểu diễn góc quay của vật

**Quaternion ( vector biểu diễn góc quay trong 3D)**

Quarternion được sử dụng trong phép quay không gian. Quaternion được định nghĩa như là một số phức có ba thành phần ảo:

 Các thành phần ảo i,j ,k có thể được coi như ba vector đơn vị của trục tọa độ x, y, z. Và một quaternion có thể được viết lại như sau:

Khi đó vector v sẽ đại diện cho hướng và w đại điện cho độ lớn của sự quay hay sự thay đổi góc nghiêng của vật so với 3 trục tọa độ x, y, z



Hình 2. Vector Quaternion trong không gian 3 chiều xyz

Trong phạm vi nghiên cứu của đề tài em sẽ không đi sâu vào tìm hiểu chi tiết về mặt giải thích và phân tích toán học đối với hệ Quaternion . Mà em sẽ dựa trên những công thức tính toán đã được chứng minh và đánh giá qua thực nghiệm đưa ra để áp dụng vào đề tài của mình .

Từ hệ quaternion nhiều nhà toán học đã đưa ra các công thức để tính góc nghiêng của 1 vật một cách tương đối chính xác tiêu biểu như Euler – Người đã đưa ra công thức tính , chuyển đổi các hệ số từ hệ Quaternion sang các thông số pitch , roll , yaw .

Việc tính toán thông số yaw áp dụng công thức Euler tỏ ra ổn định và được sử dụng nhiều , tuy nhiên pitch , roll thì khá mất ổn định . Do vậy việc tính pitch , roll em sẽ sử dụng cách tính khác dựa trên một nhà sản xuất chip NXP cung cấp. Vì công thức mà NXP đưa ra có liên quan đến trọng lực nên bước đầu tiên em sẽ phải tính được giá trị trọng lực đối 3 trục là gx , gy , gz

Trong đó

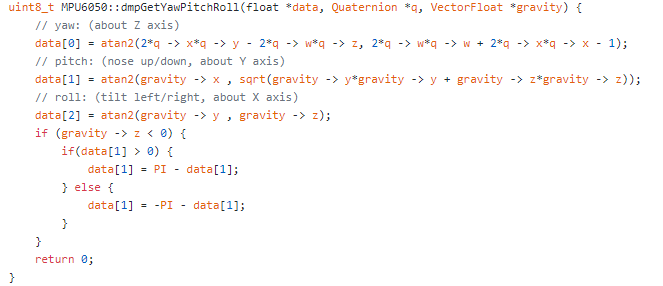
q{i}acc chính là các tham số cua hệ Quaternion lấy được từ cảm biến

gx , gy , gz là giá trị của vector trọng lực tương ứng theo 3 trục x, y,z

Tổng kết lại các công thức em sẽ áp dụng đối với dữ liệu từ cảm biến để xác định pitch , roll và yaw sẽ là :

) , ,

Việc áp dụng công thức này trong code sẽ thể hiện ở hàm :



Hình 2. Mã nguồn triển khai theo công thức.

Ở đây hàm này sẽ có nhiệm vụ tính toán các giá trị pitch ,roll, yaw từ độ lớn của vector quaternion và gravity. Riêng giá trị pitch người ta đã xác định đo được khi áp dụng công thức trên phải nằm trong khoảng [- , π] . do vậy ta sẽ phải kiểm tra và đưa giá trị pitch( data[1]) về khoảng [- , π]

Vậy là em đã trình bày cơ bản về các công thức tính góc được áp dụng trong sản phẩm . từ các thông số về vector quaternion em sẽ tính ra được pitch , roll và raw . ở phần tiếp theo em sẽ trình bày cách để có được dữ liệu về vector quaternion.

**Các Thiết bị điều khiển và thiết bị thu nhận dữ liệu**

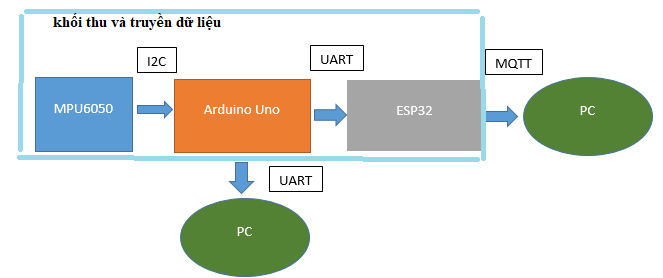
Trong phần này em sẽ trình bày về 2 loại thiết bị được sử dụng trong sản phẩm của mình đó là :

Phần thu nhận và truyền dữ liệu : Cảm biến MPU6050, Arduino , ESP32

Phần điều khiển : Lớp ứng dụng trên hệ điều hành Linux

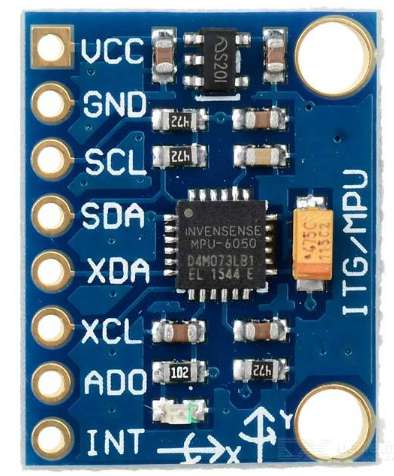
## 2.2 Phần thu nhận và truyền dữ liệu

Để có được kết quả về góc của vật trong hệ xyz thì em sẽ sử dụng cảm biến MPU6050 để đo được vector quaternion và Arduino thu nhận qua giao tiếp I2C để xử lý sau đó trả về cho máy tính qua kết nối UART hoặc MQTT ( với ESP32) . Cụ thể mô hình của khối này sẽ như sau :



Hình 2. Mô hình hệ thống đối phần thu dữ liệu

### 2.2.1 Giới thiệu về cảm biến MPU6050



Hình 2. Hình ảnh MPU6050

MPU-6050 là cảm biến của hãng InvenSense. MPU-6050 là một trong những giải pháp cảm biến chuyển động đầu tiên trên thế giới có tới 6 (mở rộng tới 9) trục cảm biến tích hợp trong 1 chip duy nhất.

MPU-6050 sử dụng công nghệ độc quyền MotionFusion của InvenSense có thể chạy trên các thiết bị di động, tay điều khiển.

MPU-6050 được trang bị :

+ con quay hồi chuyển 3 trục (3-axis MEMS gyroscope)

+ cảm biến gia tốc 3 chiều (3-axis MEMS accelerometer)

Ngoài ra, MPU-6050 còn có 1 đơn vị tăng tốc phần cứng chuyên xử lý tín hiệu (Digital Motion Processor - DMP) do cảm biến thu thập và thực hiện các tính toán cần thiết. Điều này giúp giảm bớt đáng kể phần xử lý tính toán của vi điều khiển, cải thiện tốc độ xử lý và cho ra phản hồi nhanh hơn. Đây chính là 1 điểm khác biệt đáng kể của MPU-6050 so với các cảm biến gia tốc và gyro khác.

MPU-6050 có thể kết hợp với cảm biến từ trường (xác định được phương hướng Đông –Tây-Nam-Bắc dựa trên từ trường trái đât) để tạo thành bộ cảm biến 9 góc đầy đủ thông qua giao tiếp I2C

Các cảm biến bên trong MPU-6050 sử dụng bộ chuyển đổi tương tự - số (Anolog to Digital Converter - ADC) 16-bit cho ra kết quả chi tiết về góc quay, tọa độ... Với 16-bit chúng ta sẽ có 2^16 = 65536 giá trị cho 1 cảm biến.

Tùy thuộc vào yêu cầu , cảm biến MPU-6050 có thể hoạt động ở chế độ tốc độ xử lý cao hoặc chế độ đo góc quay chính xác (chậm hơn). MPU-6050 có khả năng đo ở phạm vi:

+ con quay hồi chuyển: ± 250 500 1000 2000 dps

+ gia tốc: ± 2 ± 4 ± 8 ± 16g

Hơn nữa, MPU-6050 có sẵn bộ đệm dữ liệu 1024 byte cho phép vi điều khiển phát lệnh cho cảm biến, và nhận về dữ liệu sau khi MPU-6050 tính toán xong.

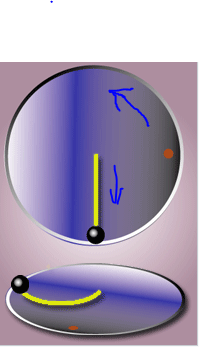
* MPU-6050 còn hỗ trợ ngắt để việc lấy dữ liệu từ vi điều khiển hiệu quả hơn

**Ứng dụng phổ biến của MPU-6050**

* Robot 2 bánh tự cân bằng
* Máy bay điều khiển từ xa
* Chuột máy tính trên không
* Tay cầm chơi game
* Hệ thống giữ thăng bằng cho camera/máy ảnh.
* Các ứng dụng: đếm số bước chân, cảm biến rơi, cảm biến rung/lắc.

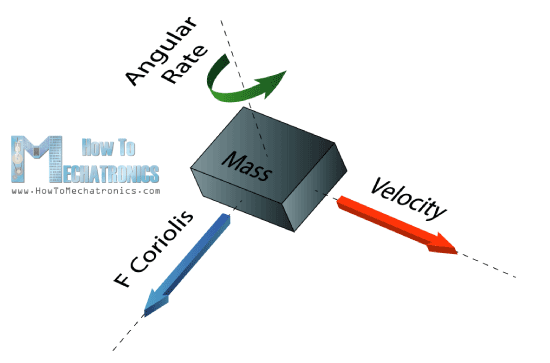
**Cảm biến đo tốc độ góc của cảm biến Gyroscope**

Việc đo của cảm biến đo góc này dựa theo nguyên lý “hiệu ứng Coriolis” . Hiệu ứng này được phát biểu :” hiệu ứng xảy ra trong các hệ qui chiếu quay so với các hệ quy chiếu quán tính. ”. Ví dụ nếu một vật chuyển động dọc theo đường bán kính theo chiều rời xa trục quay của hệ qui chiếu thì sẽ chịu tác động của một lực theo phương vuông góc với bán kính và theo chiều ngược với chiều quay của hệ. Còn nếu vật chuyển động về phía trục quay thì lực sẽ tác động vào vật theo chiều quay của hệ qui chiếu



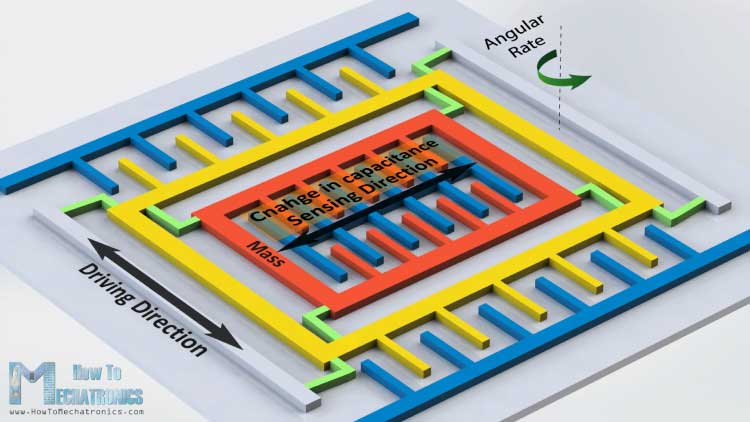
Hình 2. Hiệu ứng Coriolis

Trong hình ảnh bên trên khi vật nặng di chuyển từ tâm ra biên , vật nặng sẽ phải chịu thêm lực Coriolis theo phương vuông góc với quỹ đạo ban đầu



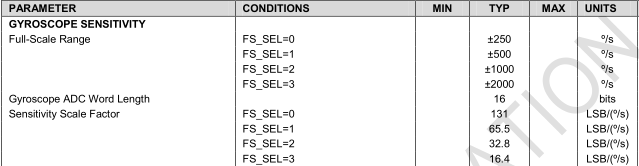
Hình 2. Hiệu ứng Coriolis lên vật Mass

Áp dụng hiệu ứng Coriolis trong việc đo góc : khi vật nặng “Mass” đang di chuyển theo hướng Velocity và trong lúc đó cả hệ chứa mass bị chịu tác động xoay của từ bên ngoài theo chiều “Angular Rate” thì vât “Mass” sẽ chịu lực “F Coriolis”. Dựa trên sự lệch khỏi quỹ đạo của “Mass” người ta sẽ tính toán được tương ứng góc quay của cả hệ chứa Mass .Đó là tính chất vật lý , trong thiết kế cảm biến người ta sẽ áp dụng và dựa trên hiện tượng vật lý đó . Ở đây hệ thống đó được gọi là Micro-Electro-Mechanical-System (MEMS) – là một hệ vi cơ điện tử bao gồm các cảm biến và các bộ chấp hành có kích thước rất nhỏ cỡ micro và milimet.



Hình 2. Cấu trúc của một cảm biến gyroscope

Bên trên là hình ảnh một vi cấu trúc bên trong cảm biến đo góc hiện hay. Cách hoạt động của nó sẽ là phần “Mass” sẽ di chuyển đều đặn theo 2 hướng “Driving Direction” , một khi có tác động lực từ bên ngoài vào cả cảm biến này thì như đã giải thích ở trên quỹ đạo của “Mass” sẽ bị thay đổi và dựa trên sự thay đổi đó các bộ xử lý bên trong sẽ được thiết kế để tính toán và đưa ra giá trị tương ứng với góc nghiêng gây ra. Giá trị đo được của cảm biến này sẽ là độ/giây (dps – degree per second)

Cảm biến gyroscope trong MPU6050 sẽ có một số chế độ đo như sau : 

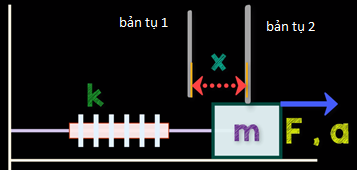
Hình 2. Thông số các chế độ của MPU6050

Theo như bảng thông số em thấy được từ datasheet thì MPU sẽ có 3 chế độ đo góc là ±250 , ±500 ,±1000 và ±2000 đơn vị là độ /giây . Tức là tùy vào ứng dụng sử dụng có sự biến thiên nhanh về gia tốc mà ta có thể chọn tương ứng sao cho phù hợp .

Ngoài ra tương ứng với việc chọn chế độ đo góc thì ta cần phải chú ý đến đơn vị của dữ liệu mà khối MPU thu được qua ADC sẽ là LSB/Unit . Ví dụ như trong thiết kế của mình em đã chọn FS\_SEL = 0 tương ứng với chế độ ±250 độ/giây thì khi ta thu được giá trị từ cảm biến sẽ phải chia cho 131 ( LSB) .

**Cảm biến đo gia tốc góc của cảm biến Accelerometer**

Việc đo được giá trị gia tốc của vật sẽ được quan sát từ hiện tượng vật lý và được thiết kế lại trong hệ thống MEMS .



Hình 2. Nguyên lý vật lý được sử dụng trong đo gia tốc.

Như hình bên trên các thành phần sẽ gồm lò xo có độ cứng là k được nối với 1 vật nặng m.

Theo định luật 2 Newton ta có F = m\*a (1), trong đó m là khối lượng và a là gia tốc của vật m lại có khi m được di chuyển một độ dài là x thì lực cần thiết để kéo dãn lò xo sẽ là F = k\*x (2). Từ (1) và (2) ta có thể suy ra được mối quan hệ :

Mặt khác k và m chúng ta có thể biết trước do vậy bản chất việc tìm ra sự thay đổi của x chúng ta đã có thể xác định được sự thay đổi của gia tốc a

Tiếp theo người ta sẽ thiết kế một tụ điện với 2 bản cực , một cực sẽ được đặt cố định và mặt còn lại sẽ được gắn lên vật nặng m theo như hình vẽ bên trên . Khi đó ta sẽ có:

Trong đó :

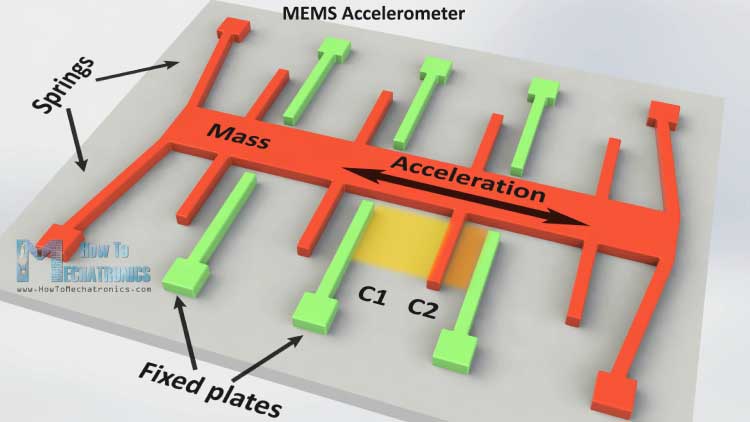
C là điện dung của tụ điện

ℇ là hằng số điện môi

S là phần diện tích bản tụ

x là khoảng cách giữa 2 bản tụ

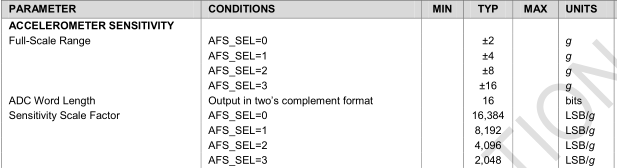
Như vậy cuối cùng ta sẽ có mỗi quan hệ giữa C và x . Mà việc đo giá trị của tụ điện lại được thực hiện khá dễ dàng do đó việc thay đổi giá trị của điện dung C thông qua một số phép tính toán ta sẽ tính ra được sự thay đổi của gia tốc .



Hình 2. Cấu trúc của một cảm biến đo góc nghiêng

Như hình bên trên chúng ta có thể thấy việc tái tạo lại hiện tượng vật lý thông qua các thiết kế micro của MEMS. Springs là những lò xo , Mass là vật nặng , các fixed plates là các bản tụ đã được đặt cố định . Nguyên lý hoạt động của chúng sẽ là các vật Mass khi chịu tác động từ lực bên ngoài thì việc Mass thay đổi giá trị điện dung C1 -> C2 sẽ được ghi lại và được tính toán chuyển đổi ra giá trị gia tốc tương ứng như em đã trình bày ở phần lý thuyết bên trên .

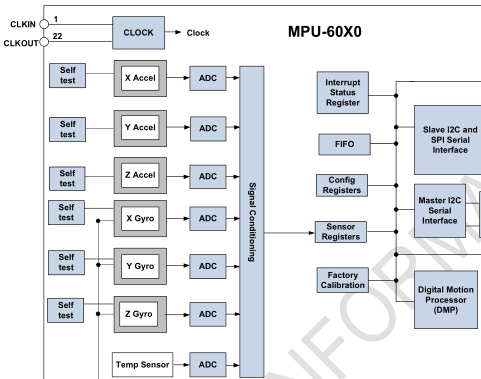
Cũng tương tự như cảm biến đo gia tốc thì cảm biến đo gia tốc ( gyroscope) thì cảm biến đo góc nghiêng ( accelometer) cũng có một bảng thể hiện các chế độ đo và tỉ lệ chia tương ứng khi có được dữ liệu từ cảm biến



Hình 2. Thông số các chế độ của cảm biến đo góc nghiêng trên MPU6050

Trong thiết kế sản phẩm của mình em đã chọn ±2g và tỉ lệ chia tương ứng là 16384 (LSB/g) g là đơn vị đo tốc độ di chuyển của vật ví so với gia tốc trọng trường g = 9.8 Ví dụ một chiếc xe tang tốc độ từ 0->100km/h trong 2.3s tương đương 28m/s trong 2.3s suy ra ta có gia tốc của vật sẽ là 12 sau đó ta lấy 12/9.8 = 1.2g

Dựa trên thông số đo được từ 2 cảm biến trên MPU6050 còn có một khối xử lý phần cứng tổng hợp thông tin từ cảm biến gia tốc và cảm biến góc đưa ra được vector Quaternion như em đã đề cập ở phần trước . khối đó chính là DMP ( Digital Motion Processor – Khối xử lý chuyển động ). Ngoài việc hỗ trợ tính toán trên DMP MPU6050 còn có một bộ đệm FIFO mục đích sau khi tính toán xong DMP sẽ đẩy ra bộ đêm này ( bộ đệm lên tới 1024 bytes) . Việc hỗ trợ tín hiệu sẵn sàng dữ liệu trong bộ đệm thông qua chân tín hiệu ngắt cũng hỗ trợ đáng kể việc tiếp nhận xử lý của bộ xử lý trung tâm của vi điều khiển



Hình 2. Tổng quan về các khối trong MPU6050

Theo như sơ đồ khối bên trên ta có thể thấy các khối quan trọng là :

Gyro X, Y , Z dung để đo thông tin về gia tốc , góc của MPU và biểu diễn dưới dạng tín hiệu số thông qua bộ ADC .

Khối self-test có nhiệm vụ so sánh những cấu hình của ta ở các thanh ghi self-test cho ra được các thông số về góc và gia tốc với kết quả cảm biến đo thực tế . Nếu đầu ra của tín hiệu lệch nhau nằm trong khoảng cho phép ( được nhà sản xuất cung cấp ) thì coi như MPU hoạt động đúng

Khối DMP sẽ làm nhiệm vụ thu nhận dữ liệu từ các thanh ghi accel và gyro sau đó tính toán và xử lý theo như các thanh ghi (Config register) đã được lập trình và đầu ra của khối này sẽ là dữ liệu đẩy vào bộ đệm ( FIFO buffer) và sẽ phát ra tín hiệu ngắt nếu ta cấu hình bật ngắt

Việc truy cập hay điều khiển khối này em sẽ thông qua giao tiếp I2C hoặc SPI . Trong thiết kế của mình em đã sử dụng giao tiếp I2C

Việc DMP tính toán ra được vector Quaternion hoàn toàn phải dựa trên 2 cảm biến đo gia tốc và góc nghiêng do vậy khi khởi động khối DMP thì ta cũng phải cấu hình theo đúng chế độ mong muốn đối với cả 2 loại cảm biến trên .

### 2.2.2 Giới thiệu về Arduino, ESP32 và joystick ( nút nhấn đa hướng)

Trong thiết kế sản phẩm của mình em sẽ chọn Arduino là bộ xử lý trung tâm nhiệm vụ là thu nhận dữ liệu từ MPU6050 qua quá trình xử lý sẽ truyền dữ liệu cho ESP32 thông qua UART để truyền không giây đến máy tính 

Hình 2. Hình ảnh kit Arduino

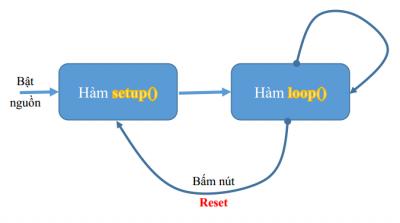
Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.

Vi điều khiển AVR là bộ xử lý trung tâm của toàn bo mạch. Ở con Arduino Uno này thì sử dụng ATMega328

Phần lớn là Arduino cơ bản với việc bổ sung các trình điều khiển đầu ra phổ biến, thường sử dụng trong giáo dục cấp trường để đơn giản hóa việclắp ráp các xe đẩy và robot nhỏ. Những biến thể khác là tương đương về điện nhưng thay đổi tham số dạng (form-factor), đôi khi cho phép tiếp tục sử dụng các Shield, đôi khi không. Một số biến thể sử dụng bộ vi xử lý hoàn toàn khác, với mức độ khác nhau về tính tương thích

Chương trình Arduino được viết bằng C hoặc C++. Arduino IDE đi kèm với một thư viện phần mềm được gọi là "Wiring" từ dự án lắp ráp ban đầu, cho hoạt động đầu vào/đầu ra phổ biến trở nên dễ dàng hơn nhiều. Người sử dụng chỉ cần định nghĩa hai hàm để thực hiện một chương trình điều hành theo chu kỳ :

* setup() : hàm chạy một lần duy nhất vào lúc bắt đầu của một chương trình dùng để khởi tạo các thiết lập.

* loop() : hàm được gọi lặp lại liên tục cho đến khi bo mạch được tắt đi

Hình 2. Cơ chế hoạt động của Arduino

Arduino Uno R3 được lựa bởi vì :

+ Giá thành thấp, dễ sử dụng

+ Mã nguồn phong phú hỗ trợ người lập trình tạo ra sản phẩm thử nghiệm nhanh chóng trước khi hướng tới một sản phẩm hoàn thiện

**ESP32**



Hình 2. Kit ESP32

ESP32-WROOM-32 là mô đun MCU đa dụng, mạnh mẽ và được sử dụng rộng rãi trong thiết kế mạch PCB Wifi- Bluetooth, BLE được ứng dụng rất phổ biến cho nhiều ứng dụng về IoT hiện nay. Phạm vi ứng dụng từ mạng sensor tiết kiệm năng lượng đến những ứng dụng với tác vụ phức tạp nhất, như mã hóa âm thanh, âm nhạc trực tuyến đến giải mã MP3.

Lõi của module là họ chip ESP32-D0WDQ6, chip nhúng được thiết kế cho khả năng mở rộng và tùy biến cao. Có đến 2 lõi CPU độc lập có thể điều khiển, tần số clock của CPU có thể được điều chỉnh tử 80MHZ đến 240 Mhz. Người lập trình có thể tắt CPU để sử dụng bộ đồng xử lý công suất thấp để theo dõi sự thay đổi hoặc vượt ngưỡng của các ngoại vi . ESP32 tích hợp bộ ngoại vi khá phong phú từ cảm biến điện dung, cảm biến Hall, SD card, Ethernet, SPI tốc độ cao, UART, I2S hay I2C.

Việc tích hợp cả Bluetooth, BLE và Wifi đảm bảo cho khả năng ứng đáp ứng nhiều loại ứng dụng khác nhau và module đó sử dụng với ngoại vi, thiết bị nào: wifi cho phép kết nối rộng rãi về mặt vật lý ra Internet qua Wi-fi router, trong khi sử dụng Bluetooth cho phép người dùng thuận tiện khi kết nối với smartphone, hay thiết bị beacon tiết kiệm điện. Ở chế độ ngủ, chíp ESP32 tiêu thụ dòng dưới 5 µA, phù hợp với những thiết kế mạch dùng pin hay thiết bị đeo được. Tốc độ truyền thông cho phép lên đến 150 Mbps, và công suất tín hiệu khoảng 20 dBm trên anten cho phép phạm vi tín hiệu xa. Như vậy module này có thông số kỹ thuật thuộc dạng đầu bảng trên thị trường cũng như hiệu suất, độ tin cậy tốt nhất cho tích hợp, thiết kế ứng dụng điện tử, tự động hóa, đòi hỏi phạm vi hoạt động rộng, tiết kiệm năng lượng cũng như khả năng kết nối đa dạng.

Hệ điều hành chạy được trên ESP32 là FreeRTOS vơi LwIP, TLS 1.2. Hỗ trợ update firmware qua OTA mã hóa, điều này cho phép nhà phát triển sản phẩm có thể nâng cấp phần mềm sản phẩm ngay cả khi thiết bị đang được sử dụng một cách tiết kiệm tiền bạc và nhân lực.

Esp32 được chọn bởi vì :

Thích hợp với ứng dụng của em là truyền không dây qua WIFI – môi trường mạng phổ biến cho các ứng dụng IoT

CPU của ESP32 có cấu hình khá mạnh phù hợp cho các ứng dụng cần truyền dữ liệu liên tục ( trong ứng dụng của mình em sẽ truyền sau mỗi 50ms)

**MQTT – Message Queue Telemetry Transport**

Giới thiệu chung

Tiếp theo sau với việc lựa chọn ESP em sẽ lựa chọn MQTT là giao thức truyền tin qua môi trường WIFI

Đây là một giao thức truyền thông điệp ( message ) theo mô hình xuất bản/ theo dõi ( publish/subcribe) sử dụng bang thông thấp , độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định .

Kiến trúc mức cao ( high-level) của MQTT gồm 2 phần chính là Máy chủ MQTT( Broker) và thiết bị khách ( Client) .

Trong đó máy chủ MQTT coi như là trung tâm , nó là giao của tất cả các kết nối đến từ máy khách . Nhiệm vụ chính của broker là nhận thông điệp ( message) từ các máy xuất bản ( publisher) , xếp các thông điệp vào một hàng đợi và chuyển đến một địa chỉ cụ thể . Nhiệm vụ phụ của một broker là nó có thể đảm nhiệm thêm một vài tính năng liên quan đến truyền thông như : bảo mật , lưu trữ thông điệp , …

Client thì được chia thành 2 nhóm là publisher ( máy xuất tin) và subscriber( máy đăng ký nhận tin) . Client sẽ làm ít nhất một trong 2 việc là xuất tin với một mã kênh truyền( topic) hoặc nhận tin từ một mã kệnh truyền ( topic )

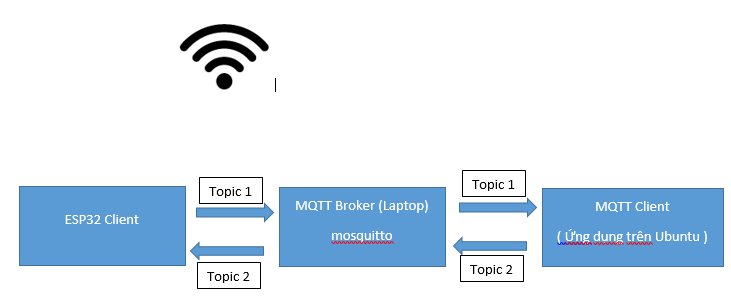
Bảo mật đối với MQTT

Theo mô hình mạng MQTT có 2 lớp bảo mật ở các tầng : ứng dụng và giao vận

Tại lớp ứng dụng : Ta có thể xác thực với tài khoản và mật khẩu trước khi có thể kết nối vào mạng của MQTT . Ngoài ra sGiá trị của trường có thể chứa tối đa 65535 ký tự, trường hợp thường được sử dụng trong thực tế để thể hiện giá trị của trường clientid là sử dụng chuẩn UUID gồm 36 ký tự để tạo ra các giá trị random không trùng nhau hoặc sử dụng địa chỉ MAC của thiết bị. Và trong gói tin phản hồi kết nối CONNACK, cũng sẽ bao gồm mã trạng thái, tất cả các trạng thái trả về được cho trong bảng sauau khi xác nhận xong mỗi Client sẽ nhận được một số định danh duy nhất .

Bảo mật tại lớp giao vận : Giao thức MQTT sử dụng TLS . TLS/SSL là các giao thức mã hóa sử dụng cơ chế handshake (bắt tay) để tạo kết nối an toàn giữa máy khách và máy chủ. Sau khi hoàn thành các bước “handshake”, một kênh giao tiếp mã hóa giữa client và server được thiết lập và đảm bảo không kẻ tấn công nào có khả năng nghe trộm thông tin trong suốt quá trình giao tiếp

Trong thiết kế sản phẩm của mình để đảm bảo tốc độ và thiết bị không cần phải quá đảm bảo dữ liệu khỏi tấn công thì sản phẩm của em chỉ sử dụng định danh để bảo mật



Hình 2. Mô hình hoạt động của MQTT trong sản phẩm.

Với mô hình trên em đã kết nối board mạch có ESP32 Client với Laptop của mình ( MQTT Broker)

Topic 1 sẽ là kênh truyền thông tin về dữ liệu cảm biến. Laptop sẽ có nhiệm vụ là đăng ký( subcribe) vào kênh này để lấy thông tin và xử lý

Topic 2 sẽ kênh truyền thông báo yêu cầu lấy dữ liệu . Khi ở trạng thái mới mở nguồn ESP32 Client sẽ chưa gửi thông tin lên kênh “Topic 1” ngay mà đợi có thông báo từ kênh này trước

Ở đây MQTT Broker của em sẽ là ứng dụng mosquito có thể dễ dàng cài đặt trên HĐH Ubuntu . Ứng dụng này sẽ mở cổng dịch vụ 1883 để tiếp nhận thông tin từ môi trường mạng sau đó gửi về cho ứng dụng của em đã viết , tiếp đó sẽ là nhiệm vụ lấy thông tin , xử lý điều khiển trên ứng dụng của em

Đối với ESP32 thì em đã sử dụng thư viện được cộng đồng phát triển với MQTT trên nền tảng phần cứng này

Tóm lại ESP32 sẽ làm nhiệm vụ : đăng ký tin nhắn trên kênh Topic 2 để nhận tín hiệu gửi dữ liệu sau đó gửi dữ liệu qua Topic 1

Về phía Laptop sẽ làm nhiệm vu : chạy Mosquitto , Mosquito đóng vai trò trung gian truyền thông tin giữa Ứng dụng của em đã viết và ESP32

Em lựa chọn ESP32 bởi vì :

* Tăng khả năng mở rộng . MQTT có thể dễ dàng áp dụng thêm một hoặc nhiều thiết bị kết nôi vào mạng này thông qua việc thêm một topic
* Chi phí ít , An toàn và bảo mật
* Tiết kiệm thời gian phát triển

**Joystick**



Hình 2. Hình ảnh Joystick

Joystick hoạt động như nút di chuyển trên tay game PS. Bên trong Joystick là 2 biến trở (10K) có thể di chuyển tự do tới/lùi/trái/phải . Hiện nay, Joystick được áp dụng vào các module như Joystick Shield,...đặc biệt đặc biệt là Module Joystick. Nó khá nhỏ gọn và dễ sử dụng. Module Joystick hoạt động ở mức điện áp 5V. Để sử dụng được, chúng ta cần đọc giá trị ở các biến trở và nút nhấn trên các chân của module

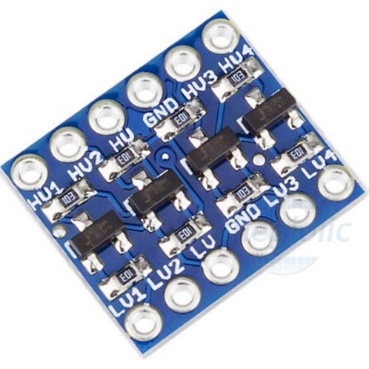
VRx hay X tương ứng giá trị biến trở trục X

VRy hay Y tương ứng giá trị biến trở trục Y

Thông qua bộ ADC của Arduino với độ phân giải 10 bit chúng ta có thể lấy thông tin từ biến trở dễ dàng để xử lý . Trong dự án của mình em có sứ dụng thiết bị này để xác định hướng đi của nhân vật trong ứng dụng trò chơi.

**Mạch Chuyển Mức Tín Hiệu**

Vì Arduino Uno hoạt động với các chân tín hiệu ở điện áp 5V mà ESP lại hoạt động ở điện áp 3.3V do đó để có thể giao tiếp giữa Arduino và ESP em sẽ cần một bộ chuyển đổi 2 chiều giữa 3.3 và 5V sử dụng trong giao tiếp UART



Hình 2. Mạch chuyển đổi logic 2 chiều 3.3v-5v

## 2.2.3 Các giao tiếp giữa cảm biến với vi điều khiển.

Để các thiết bị giao tiếp được với nhau thì cần có các chuẩn giao tiếp . Một số chuẩn giao tiếp mà em đã sử dụng trong sản phẩm của mình đó là

|  |  |  |
| --- | --- | --- |
| STT | Module | Giao tiếp với VĐK |
| 1 | ESP32 | UART |
| 2 | MPU6050 | I2C |
| 3 | Joystick | ADC |

Bảng 2. Các loại giao tiếp với VĐK

### 2.2.4 Một số giao tiếp sử dụng trong đề tài.

#### **2.2.4.1. ADC**

Các tín hiệu mà bạn thường gặp trong tự nhiên chẳng hạn như điện áp, ánh sáng, âm thanh,… đều tồn tại dưới dạng tương tự có nghĩa là tín hiệu liên tục và mức độ chia nhỏ vô hạn.

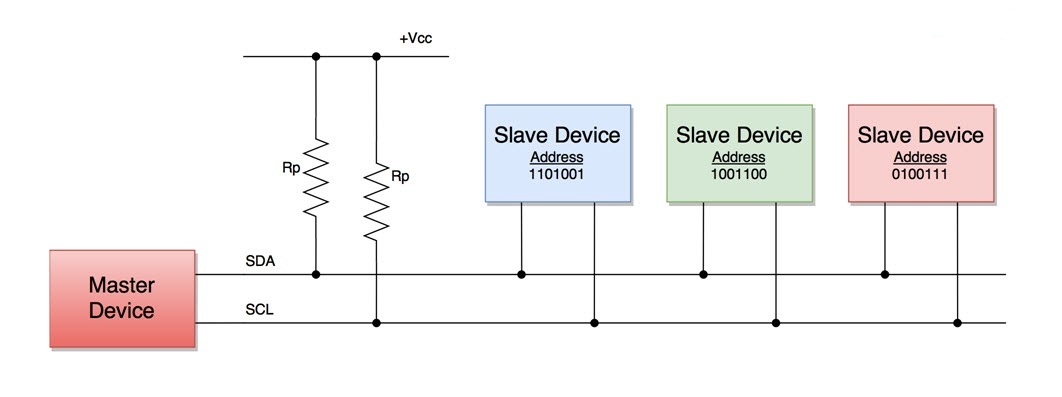
Ví dụ như trong khoảng điện áp từ 0 -> 5V thì sẽ có vô số khoảng giá trị điện áp, ánh sáng sẽ tồn tại từ mờ cho tới sáng tỏ, âm thanh từ nhỏ cho đến lớn. Ngược lại trong vi điều khiển chỉ có khái niệm số (digital), cấu trúc từ nhân cho đến bộ nhớ hoạt động dựa trên các bóng bán dẫn chỉ gồm mức 0-1 nên bạn muốn giao tiếp với chip thì phải số hóa trước khi đưa vào chip. Quá trình số hóa có thể thực hiện bằng nhiều cách và nhiều công đoạn nhưng mục đích cuối cùng là để vi điều khiển hiểu được tín hiệu tương tự đó.

ADC (**analog-to-digital converter**) bộ chuyển đổi tín hiệu tương tự-số là thuật ngữ nói đến sự chuyển đổi một tín hiệu tương tự thành tín hiệu số hóa để dùng trong các hệ số (digital) hay vi điều khiển. Vi điều khiển có hỗ trợ ADC chuyển đổi tín hiệu điện áp thành tín hiệu số với độ phân giải 12bit. VD: Cần đo điện áp tối thiểu là 0V và tối đa là 3.3V, trong vi điều khiển sẽ chia 0->3.3V thành 4096 khoảng giá trị (từ 0 -> 4095), khi que đo từ chân IO đọc được 0 thì tương đương 0V, đọc được 2047 tương đương 1.65V và đọc được 4095 tương đương 3.3V.

#### **2.2.4.2. I2C ( Inter-Integrated Circuit)**

I2C sẽ được sử dụng trong quá trình giao tiếp giữa Arduino để lấy thông tin từ cảm biến

#### I2C là một giao thức giao tiếp được phát triển bởi Philips Semiconductors để truyền dữ liệu giữa một bộ xử lý trung tâm với nhiều IC trên cùng một board mạch chỉ sử dụng hai đường truyền tín hiệu.



Hình 2. Mô hình kết nối I2C

Cả hai đường bus I2C (SDA, SCL) đều hoạt động như các bộ lái cực máng hở (open drain). Nó có nghĩa là bất kỳ thiết bị / IC trên mạng I2C có thể lái SDA và SCL xuống mức thấp, nhưng không thể lái chúng lên mức cao. Vì vậy, một điện trở kéo lên (khoảng 1 kΩ đến 4,7 kΩ) được sử dụng cho mỗi đường bus, để giữ cho chúng ở mức cao (ở điện áp dương) theo mặc định

Tốc độ truyền: 100Kbits/s ở chế độ chuẩn (Standard mode), có thể lên tới 400Kbits/s ở chế độ nhanh(Fast mode) và cao nhất là 3,4Mbits/s – Chế độ cao tốc

(High‐speed mode).Trong sản phẩm của mình em lựa chọn tốc độ 400kHz là tần số xung clock để giao tiếp với MPU6050

I2C bus sử dụng

Chân SCL( Serial Clock Line) cấp truyền tín hiệu xung clock cấp từ master

Chân SDA( Serial Data Line) là đường truyền dữ liệu

Đặc điểm quan trọng của giao tiếp I2C là

Chỉ cần có hai đường bus (dây) chung để điều khiển bất kỳ thiết bị / IC nào trên mạng I2C

Không cần thỏa thuận trước về tốc độ truyền dữ liệu như trong giao tiếp UART. Vì vậy, tốc độ truyền dữ liệu có thể được điều chỉnh bất cứ khi nào cần thiết

Cơ chế đơn giản để xác thực dữ liệu được truyền

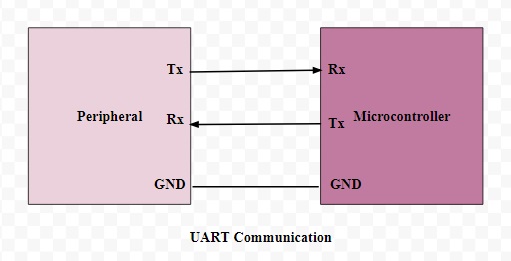
Sử dụng hệ thống địa chỉ 7 bit để xác định một thiết bị / IC cụ thể trên bus I2C

Các mạng I2C dễ dàng mở rộng. Các thiết bị mới có thể được kết nối đơn giản với hai đường bus chung I2C

#### **2.2.4.3. UART (Universal asynchronous receiver transmitter)**

UART là một ngoại vi cơ bản trong board Arduino thường được dùng trong các quá trình giao tiếp với các loại module như: Zigbee, Bluetooth, Wifi…

Chuẩn giao tiếp UART sẽ được em sử dụng trong giao tiếp giữa Arduino và ESP32 trong chế độ truyền không dây hoặc cũng có thể giao tiếp với máy tính qua dây chuyển đổi USB sang UART



Hình 2. Giao diện kết nối UART

Trong kết nối của một giao tiếp UART sẽ có :

Chân TX dùng để truyền tin, Chân RX để nhận tin

Các đặc điểm quan trọng của UART:

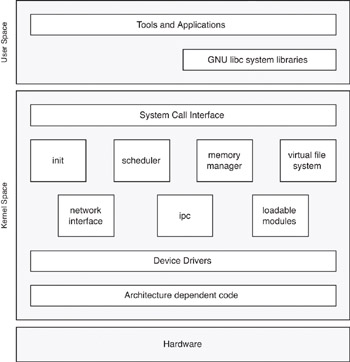
* Nó chỉ cần hai dây để truyền dữ liệu
* Tín hiệu xung clock là không cần thiết.
* Nó bao gồm một bit chẵn lẻ để cho phép kiểm tra lỗi
* Kích thước khung dữ liệu tối đa là 9 bit

## 2.3. Giới thiệu hệ thống và cách tương tác với trình quản lý thiết bị trên HĐH Linux ( driver )

### 2.3.1. Giới thiệu chung

**Linux** là tên gọi của một hệ điều hành máy tính và cũng là tên nhân của hệ điều hành. Nó có lẽ là một ví dụ nổi tiếng nhất của phần mềm tự do và của việc phát triển mã nguồn mở

Phiên bản Linux đầu tiên do Linus Torvalds viết vào năm 1991, lúc ông còn là một sinh viên của Đại học Helsinki tại Phần Lan. Ông làm việc một cách hăng say trong vòng 3 năm liên tục và cho ra đời phiên bản Linux 1.0 vào năm 1994. Bộ phận chủ yếu này được phát triển và tung ra trên thị trường dưới bản quyền GNU General Public License. Do đó mà bất cứ ai cũng có thể tải và xem mã nguồn của Linux.



Hình 2. Kiến trúc của hệ thống Linux

Theo như kiến trúc hệ thống trên thì một hệ thống Linux sẽ bao gồm 2 vùng không gian là không gian người dung và không gian hệ thống .

ở không gian người dùng là nơi các ứng dụng tương tác với người dùng chạy hoặc các ứng dụng ngầm chạy ở chế độ nền

ở không gian hệ thống ( kernel space) sẽ là nơi nhân Linux chạy với nhiều khối kết hợp nhỏ hơn làm nhiệm vụ đặc thù ( khối lập lịch , khối quản lý bộ nhớ, khối quản lý hệ thống file , khối quản lý thiết bị ,.. )

trong đó phần device driver sẽ là phần liên quan trực tiếp đến việc điều khiển thiết bị phần cứng

**Linux được chọn để sử dụng bởi các ưu điểm như:**

+ Mã nguồn của Linux là mở đối với mọi người do đó em dễ dàng tham khảo và học hỏi được kiến trúc của nó từ đó đưa ra được hướng giải quyết về việc tương tác ngoại vi trên Linux

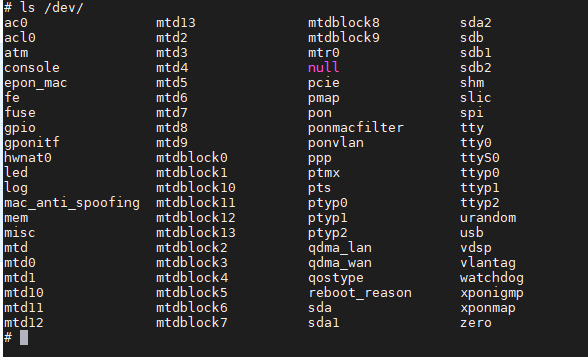
+ Cộng đồng rộng lớn , tài liệu khá là chi tiết cho phép người lập trình dễ tiếp cận

+ Linux là nền tảng phần lớn cho các thiết bị nhúng , việc học hỏi theo kiến trúc này sẽ là phù hợp với xu thế

**2.3.2 Trình quản lý thiết bị ( driver)**

Trong phần đồ án của mình em sẽ tập trung vào khối trình quản lý thiết bị ( device driver) cụ thể là loại character device driver – loại trình quản lý giao tiếp với thiết bị theo từng byte

Trong hệ thống Linux thì các thiết bị được nhận diện sẽ được biểu diễn dưới dạng file và hầu hết chúng sẽ nằm trong thư mục /dev



Hình 2. Các file trong thư mục /dev

Với hình bên trên ta có thể thấy rằng các thiết bị đại diện cho màn hình sẽ là tty, các thiết bị đại diện cho đĩa cứng sẽ là mtd, ….

Sau đó việc các ứng dụng tương tác với phần cứng như việc đọc thông tin , ghi thông tin , ra lệnh ,.. sẽ tương ứng với các thao tác với file

Ví dụ việc kết nối một board Arduino vào một máy tính

+ Arduino board sẽ đại diện trên Linux dưới dạng file và nằm ở /dev/ttyACM0

+ việc cài đặt chế độ baudrate, bít chẵn lẻ sẽ sử dụng thông qua cơ chế ioctl dùng để ra lệnh cho thiết bị . Các thư viện đã được viết sẵn dễ dàng cho người lập trình sẽ nằm ở “linux/serial.h”

+ Việc đọc/ghi dữ liệu từ Arduino board qua cổng UART sẽ giống như việc đọc/ghi vào file /dev/ttyACM0 thông qua các hàm read , write đã được khai báo trong “stdio.h”

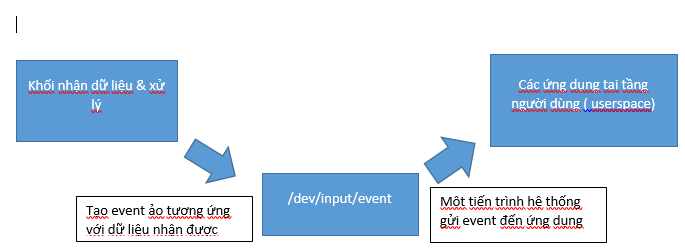
Dựa vào những kiến thức cơ bản về driver em đã thấy vận dụng được vào trong đề tài của mình bằng việc những việc thao tác với thiết bị như em đã trình bày bên trên . Ở chế độ giao tiếp bằng UART để truyền thông tin về ngóc nghiêng pitch , yaw , roll ,.. em đã đọc/ghi trực tiếp vào những file thiết bị đó . Một phần nữa là làm sao để có thể biến thiết bị của mình như một thiết bị của hệ thống thì em cũng thông qua 2 file thiết bị là chuột và bàn phím để có thể giả lập thiết bị của mình chính là chuột và bàn phím để gửi tín hiệu tới các ứng dụng khác . Chi tiết em sẽ trình bày ở phần ngay dưới đây

**2.3.4 hệ thống thông báo tín hiệu cho các ứng dụng từ trình quản lý thiết bị**

Trong HĐH Linux mọi thứ đều biểu diễn dưới dạng file ( thông tin thiết bị , dữ liệu , thiết bị , cấu hình ,…)

Và các ứng dụng ở trên tầng người dùng thì sẽ lấy thông tin từ những file đó để thực hiện cho đúng nhiệm vụ của mình . Ví dụ hầu hết các ứng dụng sẽ có một file cấu hình mặc đinh , khi nó được khởi động lên thì việc đầu tiên sẽ là đọc file cấu hình và áp dụng vào ứng dụng đó như việc truyền các tham số vào cho một chương trình vậy . Sau việc cấu hình đó đa số ứng dụng sẽ cần có sự tương tác của con người khi đó chuột và bàn phím là 2 thiết bị chính để tương tác trực tiếp

Với ứng dụng của em là sử dụng các thiết bị bên ngoài để giả lập là chuột và một số chức năng của bàn phím thì việc quan trọng là cần hiểu được HĐH Linux tương tác với các ứng dụng như thế nào ( ở đây em muốn nói đến là các ứng dụng đang chạy chính – được người dùng trực tiếp điều khiển , nó khác với các ứng dụng ngầm sẽ không chịu sự tác động trực tiếp của con người từ bàn phím hay chuột )

Với mỗi ứng dụng như khi được chọn làm ứng dụng chạy chính ( khi ta click vào nó ) thì mỗi khi phần mã nhân hệ thống điều khiển thiết bị qua khối driver nhận được thông tin phần cững có sự thay đổi ( click , di chuyển ,..) thì nó sẽ phát ra sự kiên để một khối khác trong hệ thống điều khiển ứng dụng theo như sự kiện ( event ) đã nhận được . Cụ thể event đó có thể được đọc được thông qua một file /dev/input/event

Hình 2. Mô hình chi tiết hoạt động của MQTT trong sản phẩm

Với mô hình trên em có thể giải thích như sau : khôi nhận dữ liệu có thể qua MQTT hoặc UART sau đó được tính toán vị trí chuột , phím nhấn thích hợp sẽ tạo event ảo tương ứng với dũ liệu nhận được và ghi sự kiện đó vào /dev/input/event sau đó sẽ có một tiến trình của hệ thống khác sẽ lấy các event được gửi tới và chuyển đến ứng dụng mà người dùng đang trực tiếp điều khiển . Nhờ đó mà em có thể coi như thiết bị của mình chính là bàn phím hay chuột , việc giải thích triển khai mã nguồn đối với phần này em sẽ trình bày ở phần 3.

Tuy nhiên việc thiết kế tạo event ảo này sẽ có nhược điểm là xử lý sẽ chậm vì nếu đúng ra việc viết ứng dụng như thế này sẽ phải coi là phần driver và phải được thực hiện ở miền nhân hệ thống ( kernel space ) . Nhưng việc lựa chọn nền tảng là laptop để dễ tiếp cận các ứng dụng đã có và việc viết một driver hoàn chỉnh mà không biết được đến phần cứng của laptop đồng thời phải tạo được các file event tương thích với hệ thống là điều khá khó khăn với em ở thời điểm hiện tại .Tuy nhiên với nhưng mã nguồn hiện tại việc tìm hiểu sâu thêm để có thể viết một driver hoàn chỉnh sẽ là thực hiện được trên một board nhúng chạy Linux

# 

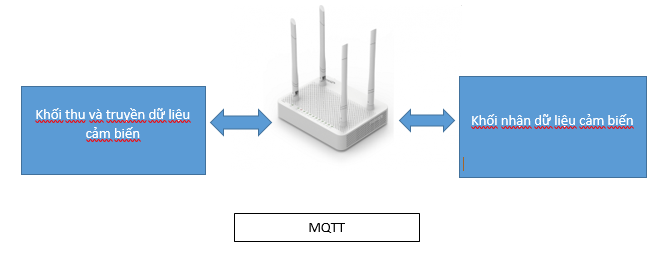
# CHƯƠNG III: THIẾT KẾ SẢN PHẨM

Từ những cơ sở lý thuyết đã được trình bày trong 2 chương trên, trong **CHƯƠNG III** này em sẽ trình bày về cách mà các công nghệ cũng như cơ sở lý thuyết sẽ được áp dụng vào sản phẩm như thế nào. Nội dung chính của chương sẽ gồm các phần sau:

* **Sơ đồ khối của sản phẩm**
* **Thiết kế sản phẩm**
* **Thiết kế phần mềm**

Dưới đây em xin trình bày cụ thể, chi tiết quá trình thiết kế cũng như cách thức hoạt động của sản phẩm.

## 3.1. Sơ đồ khối hệ thống



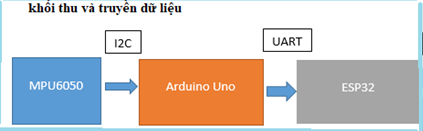
Hình 3. Sơ đồ khối hệ thống

Về cơ bản thiết bị có 2 phần chính là:

- Khối thu và truyền dữ liệu sẽ bao gồm Arduino và ESP32 . Arduino sẽ đọc dữ liệu cảm biến và xử lý tính toán góc , sự kiện bấm nút sau đó gửi dữ liệu cho ESP32 để truyền qua wifi thông qua giao thức MQTT

- Khối nhận dữ liệu cảm biến : Một Laptop có cài đặt MQTT Broker nhiệm vụ là ứng dụng trung gian phục vụ nhiệm vụ chuyển tiếp các bản tin từ 2 Client với nhau ( client thứ nhất là ESP32 , client thứ hai là một ứng dụng trên Linux do em viết )

### 3.1.1. Sơ đồ khối thu và truyền dữ liệu cảm biến



Hình 3. Sơ đồ khối thu và truyền dữ liệu

Giải thích các khối

* MPU6050 sẽ thực hiện nhiệm vụ tính vector quaternion thông qua khối DMP.
* Arduino Uno: thực hiện cấu hình cảm biến , nhận dữ liệu từ cảm biến , nút nhấn rồi truyền cho ESP32 để chuyển tiếp
* ESP32: thực hiện truyền dữ liệu sang bên nhận thông qua sóng WIFI với giao thức MQTT

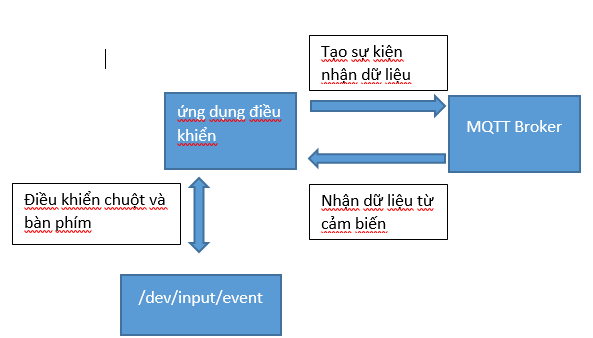
Khối nhận và xử lý dữ liệu hoạt động như sau : Arduino sẽ cấu hình MPU6050 qua I2C để khởi động các khối đo góc , gia tốc là gyroscope , accelometer và cho phép tạo ngắt qua chân INT trên MPU6050 .

Arduino sẽ tiến hành đọc dữ liệu trên bus I2C khi có tín hiệu gây ngắt trên chân INT .

Arduino truyền dữ liệu qua UART với ESP32 , vì Arduino hoạt động ở mức 5V và ESP32 hoạt động ở 3.3V do đó chúng sẽ được kết nối với nhau qua một module chuyển logic 2 chiều .

ESP32 sau khi nhận được dữ liệu từ UART thì nó thực hiện kết nối với MQTT broker và tiến hành truyền dữ liệu theo đúng chủ đề ( topic) mà em đã quy ước .

### 3.1.2. Sơ đồ khối nhận và xử lý dữ liệu cảm biến



Hình 3. Sơ đồ khối nhận và xử lý dữ liệu cảm biến

Giải thích các khối

+ MQTT Broker : Khối trung gian truyền và gửi dữ liệu dựa trên chủ đề ( topic)

+ Ứng dụng điều khiển : lấy dữ liệu từ MQTT Broker từ cảm biến qua Broker . Sau đó điều khiển chuột và bàn phím qua /dev/input/event

+ File sự kiện ( /dev/input/event) : sẽ nhận sự kiện và được hệ thống điều khiển bàn phím và chuột dựa trên event

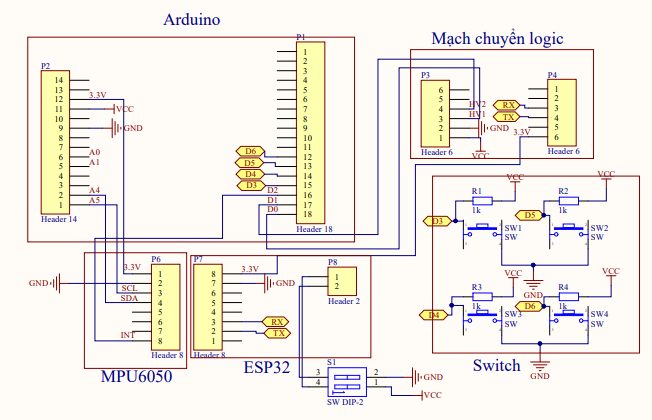
Khối này sẽ hoạt động như sau: Ứng dụng điều khiển sẽ gửi yêu cầu nhận sự kiện đến Broker , Broker sẽ là trung gian gửi và nhận đến khối thu và truyền dữ liệu cảm biến , tiếp đó Ứng dụng của em sẽ nhận được dữ liệu rồi xử lý phù hợp và tạo sự kiện tương ứng đến /dev/input/event để có thể điều khiển chuột và bàn phím

## 3.2. Thiết kế phần cứng

### 3.2.1. Mạch kết hợp Arduino, MPU6050 và ESP32

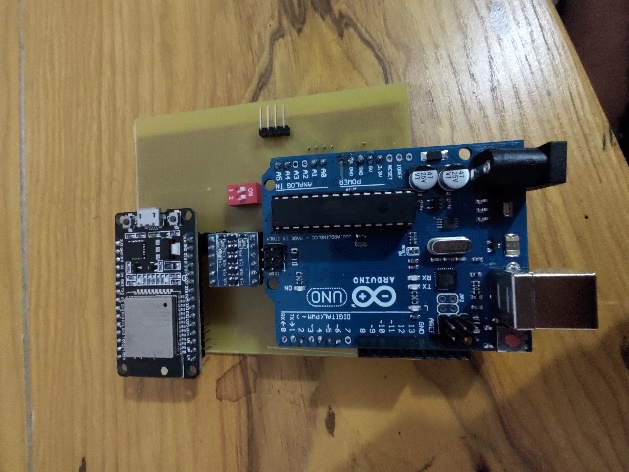
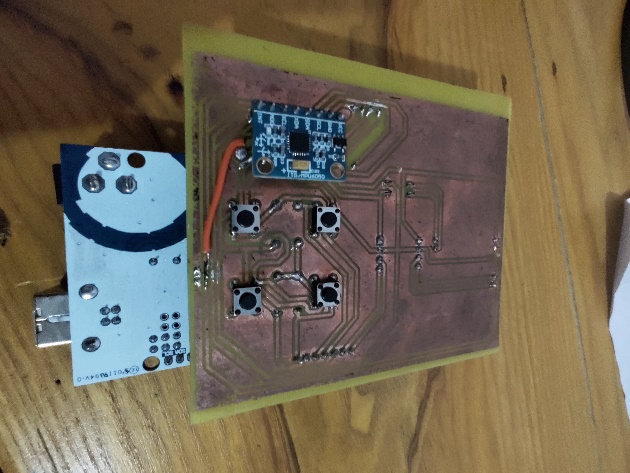
Mạch nguyên lý sẽ bao gồm Arduino giao tiếp với MPU6050 qua I2C và giao tiếp với ESP32. Một số nút nhấn được kết nối để hỗ trợ thêm tính năng cho chuột

Ở đây em sẽ cấp nguồn cho Arduino và từ Arduino sẽ cấp cho toàn bộ mạch thông qua các chân 5v , 3.3v.



Hình 3. Mạch nguyên lý của sản phẩm

Mạch sau khi hoàn thiện :

Sản phẩm thực tế ( mặt sau bên trái , mặt trước bên phải )

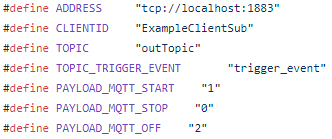
## 3.3. Thiết kế phần mềm

Mô hình thực tế em đang sử dụng sẽ là một thiết bị kết nối với một MQTT Broker . Trong hướng phát triển sẽ là nhiều thiết bị cùng kết nối với một MQTT Broker và chúng sẽ được phân biệt với nhau qua chủ đề ( topic ) gửi đến

### 3.3.1. Các bước thiết lập hệ thống

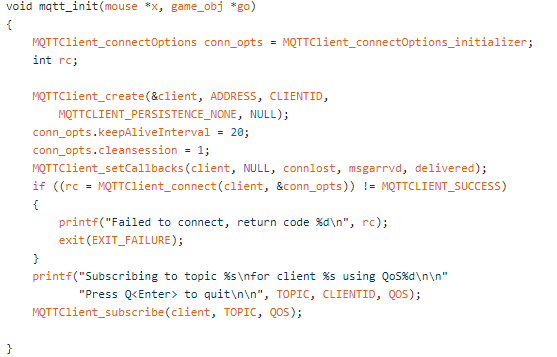
**a) Cấu hình MQTT Broker**

Dựa trên mã nguồn mở của giao thức MQTT với ngôn ngữ C , em sử dụng thư viện đó và áp dụng cho ứng dụng của mình với các API đã được cung cấp .



Hình 3. Cấu hình kết nối MQTT

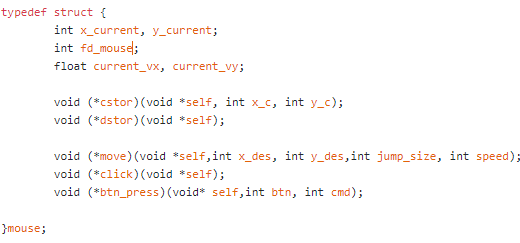
Trên đây là một số cấu hình để phía ứng dụng của em có thể kết nối được với MQTT Broker và một số lệnh , chủ đề ( topic ) truyền xuống ESP32 .



Hình 3. Khời tạo kết nối với MQTT

Hàm mqtt\_init sẽ có nhiệm vụ khởi tạo kết nối tới MQTT Broker thông qua hàm MQTT\_connect , và đăng ký các hàm để khi nhận được thông báo từ Broker là msgarrdv. Cuối cùng là hàm này sẽ đăng ký chủ đề ( topic ) nhận dữ liệu từ cảm biến với Broker .

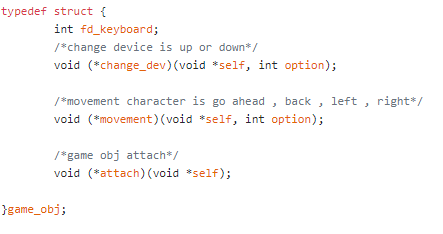
**b) Cấu hình đối tượng chuột , bàn phím , điều khiển trò chơi**



Hình 3. đối tượng “chuột” được biểu diễn trong chương trình

Đối tượng chuột được tạo ra sẽ có một số đặc tính :

* x\_current , y\_current sẽ đại diện cho giá trị vị trí hiện tại
* fd\_mouse sẽ đại diện cho một file event để tác động lên thiết bị chuột
* current\_x, current\_y sẽ là giá trị vận tốc của chuột theo 2 trục x, y
* cstor/dstor sẽ là khởi tạo , giải phóng đối tượng chuột
* move, click , btn\_press sẽ làm các nhiệm vụ liên quan đến thao tác với chuột



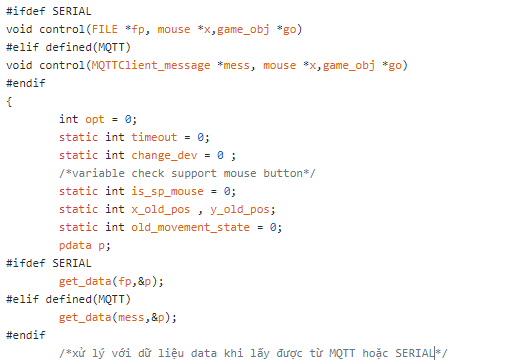
Hình 3. đối tượng “bàn phím” được biểu diễn trong chương trình

Đối tượng game\_obj sẽ có một số thuộc tính liên quan đến việc điều khiển bàn phím:

* fd\_keyboard sẽ là file đại diện cho event tác động lên bàn phím
* change\_dev , movement , attach sẽ làm các nhiệm vụ nhấn nút tương ứng với các dữ liệu nhận được từ MQTT Broker

Việc xây dựng theo dạng đối tượng như bàn phím và chuột như thế này sẽ giúp cho em dễ dàng hơn trong việc kiểm soát , nâng cấp một đối tượng hay phương thức cho nó . Lập trình theo hướng đối tượng này đã được áp dụng rất nhiều trong mã nguồn của nhân HĐH Linux với ngôn ngữ lập trình C.

### 3.3.2. Thiết kế thành phẩn điều khiển trung tâm



Hình 3. Hàm điều khiển trung tâm

Ở phần trên đây sẽ là hàm control hàm điều khiển chính của hệ thống . tùy theo cấu hình mà hàm này có thể đọc trực tiếp từ cổng USB2UART trên laptop nếu mạch kết nối với PC qua đường truyền USB2UART hoặc với cách truyền thông tin không dây bằng giao thức MQTT qua sóng WIFI .

Nhiệm vụ chính của hàm này sẽ là lấy được dữ liệu và tính toán vị trí tương ứng với dữ liệu đã nhận được từ một trong 2 cách bên trên sau đó nó sẽ tác động vào các file event của chuột , bàn phím tương ứng .

### 3.3.3. Thiết kế khung dữ liệu truyền

Dữ liệu của em sẽ được truyển theo khung như sau :

[ ^ yaw pitch roll state\_mouse oriented\_joystick $ ]

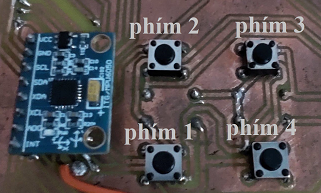
Trong đó ký tự đầu và cuối là ”^” và “$” bắt đầu một khung truyền dữ liệu để ESP32 sau khi nhận biết được khi nào là nhận xong dữ liệu , khi nào là bắt đầu nhận mới

Tiếp theo sau đó sẽ là 3 thông số yaw , pitch , roll đại diện cho góc nghiêng hiện tại của cảm biến

State\_mouse là trạng thái của các button tuân theo quy ước trước của em , hiện tại em có 4 nút nhấn sẽ tối đa là trạng thái. Quy ước của em sẽ là mỗi phím sẽ tương ứng với ảnh hưởng của 4 bit được truyền đi . Như hình bên dưới mã nguồn của em sẽ triển khai như công thức bên dưới ( trong trường hợp bấm trạng thái nút nhấn sẽ bằng 0 ) thì

state\_mouse = (!button1 << 0) | (!button2 << 1) | (!button3 << 2) | (!button4 << 3);

trong đó : “! button thứ i” chính là đảo logic của button thứ i



Hình 3. Phím bấm trên sản phẩm

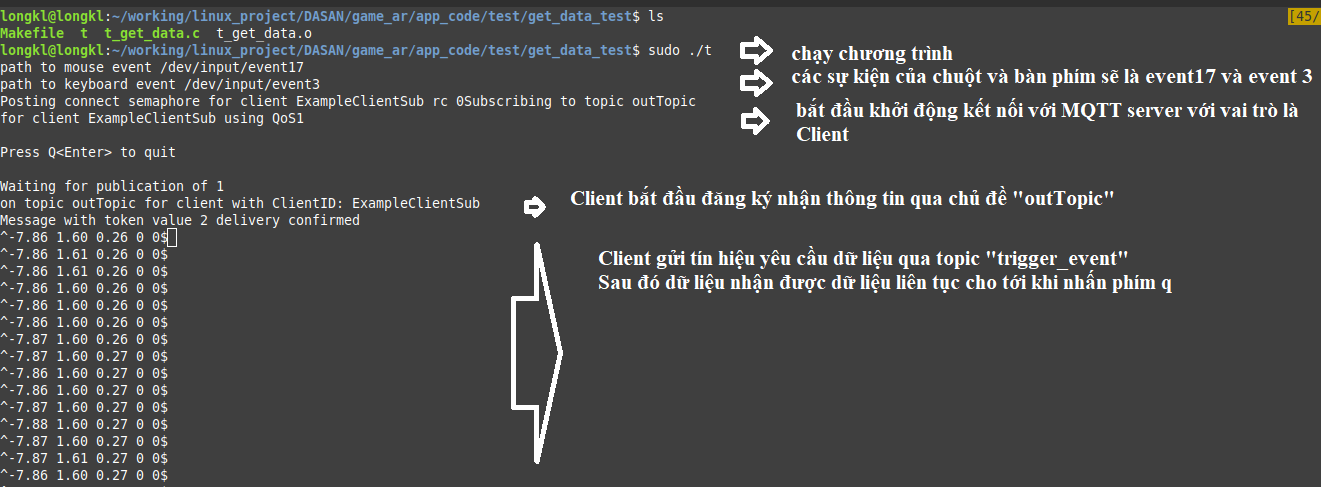
## 3.4 Thử nghiệm và đánh giá

### 3.4.1 Sản phẩm thử nghiệm

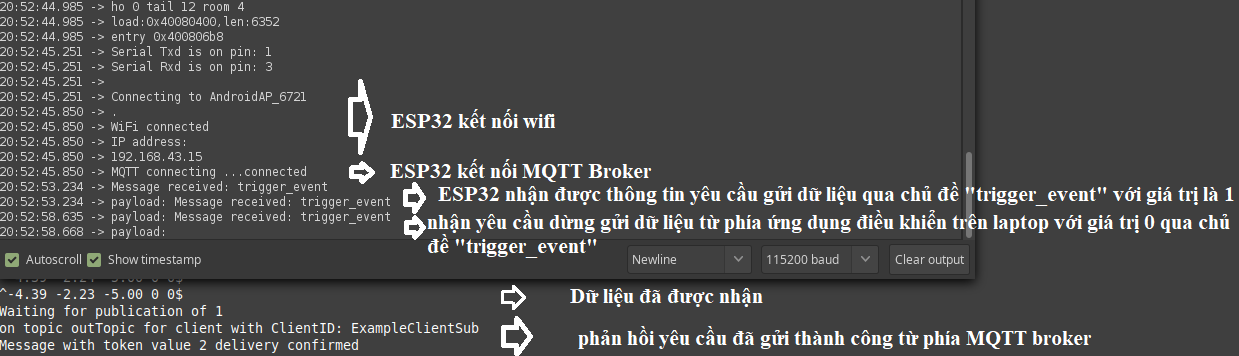
Ghép module cảm biến, Arduino , ESP32 , Joystick . tìm hiểu ở chương II vào với nút cảm biến mẫu ta được sản phẩm có thể điều khiển tương tác ảo có thể thay thế một phần chức năng của chuột và bàn phím nhưng có thể đem lại cảm giác chân thực hơn.

Sản phẩm hoạt động khá ổn định và dễ dàng thêm mới tính năng sao cho phù hợp với ứng dụng hướng tới để điều khiển.

Dữ liệu thu được từ MQTT hay từ UART sẽ không có sự khác biệt nhau dữ liệu thu được sẽ như hình bên dưới :



Hình 3. 11 thông tin khi chạy của ứng dụng điều khiển trung tâm



Hình 3. Thông tin khi ESP32 kết nối với MQTT Broker

Sau việc thu được dữ liệu em sẽ tiến hành tính toán và điều khiển bàn phím và chuột như đã trình bày ở các phần trước.

### 3.4.2 Đánh giá

**a) Tốc độ đáp ứng**

Với mã nguồn hiện tại có thể cho phép 2 loại kết nối tới ứng dụng điều khiển là qua UART hoặc MQTT

Sau quá trình thử nghiệm thì ứng dụng lấy dữ liệu qua UART sẽ hoạt động ổn định hơn và đáp ứng tốt hơn , còn đối với MQTT sẽ phụ thuộc vào môi trường mạng khả năng phát sóng WIFI của thiết bị phát .

Hiện tại cách để vẫn đảm bảo tối thiểu hóa về độ trễ dữ liệu đối với phương thức truyền qua sóng WIFI với giao thức MQTT sẽ là phải đảm bảo một đường truyền đảm bảo tốc độ , ổn định chỉ phục vụ cho việc truyền nhận các bản tin trọng mạng MQTT.

**b) Thiết kế hệ thống**

Đối với phần ứng dụng :

Hệ thống có thiết kế theo hướng tách biệt các thành phần theo từng đối tượng , với việc làm như vậy sẽ dễ dàng trong việc nâng cấp hay quan trọng hơn là việc hướng đến nhiều người cùng thao tác trong một môi trường ảo trên các thiết bị độc lập với nhau thông qua việc thay đổi topic cho từng đối tượng riêng biệt .

Thực chất việc thiết kế ứng dụng này vẫn chưa được tối ưu được tốc độ do các phần thiết kế đọc dữ liệu và điều khiển chuột , bàn phím nên được tích hợp dưới lớp điều khiển thiết bị của nhân Linux ( device driver ) tuy nhiên việc viết mã để tạo được driver trên một Laptop , liên kết driver đó với hệ thống ( thông qua tạo event file ) để đảm bảo sự tích hợp là vấn để em cần tìm hiểu thêm.

Đối với phần VĐK :

Việc thu nhận dữ liệu hiện chạy khá ổn định tuy nhiên việc sử dụng phần cứng của em mới là bước đầu trong việc thử nghiệm nhằm mục đích tập trung vào việc phát triển phần mềm nên sử dụng các phần cứng chủ yếu sẽ là module chưa tích hợp tối ưu được

c) thiết kế

Hiện tại với phiên bản có thể điều khiển các chức năng cơ bản của chuột , bàn phím trong các ứng dụng giải trí nên việc thiết kế về phía tương tác người dụng vẫn chưa được hợp lý

## 3.5 Kết luận chương

Chương III là kết quả thiết kế thành công thiết bị tương tác ảo , việc điều khiển khá là đảm bảo khả năng đáp ứng về tốc độ. Qua việc làm sản phẩm thử nghiệm cho thấy sản phẩm hoàn toàn có khả năng sử dụng lâu dài , tương tác tốt với người dùng .

# KẾT LUẬN

Sau thời gian nghiên cứu với sự nỗ lực của bản thân và sự hướng dẫn tận tình của thầy Phạm Văn Sự, em đã hoàn thành đồ án “Thiết bị giải trí VR”.

Kết quả, đồ án đã đạt được những nội dung cụ thể sau đây:

* Hiểu được nguyên lý cũng như cách thức hoạt động của cảm biến , lớp điều khiển thiết bị trên HĐH Linux ( device driver )
* Hiểu được cách các ứng dụng sử dụng các sự kiện từ phần cứng trên HĐH Linux
* Hiểu và áp dụng được các kiến thức đã được học về giao thức truyền thông , phần cứng , thiết kế phần mềm ,… vào việc tạo ra một ứng dụng thực tế.

Hướng phát triển của đồ án:

* Thiết kế mã nguồn ở tầng điều khiển thiết bị đểm đảm bảo tốc độ.
* Thiết kế tối ưu phần cứng hơn thông qua việc thay đổi các module bằng dùng chíp nhằm giảm giá thành , tăng khả năng linh hoạt với kích thước nhỏ hơn.
* Phát triển việc điều khiển với nhiều người dùng cùng tương tác.

# TÀI LIỆU THAM KHẢO

[1]. TS. Đặng Hoài Bắc, TS. Nguyễn Ngọc Minh, Thiết kế hệ thống nhúng, Nhà xuất bản Thông tin và Truyền thông, Hà Nội, 2013.

[2]. ThS. Trần Thúy Hà, ThS. Đỗ Mạnh Hà, Giáo trình điện tử số, Nhà xuất bản Bưu điện, 2009.

[3]. ThS. Đỗ Mạnh Hà, ThS. Trần Thúy Hà, ThS. Trần Thục Linh, Bài giảng Cấu kiện điện tử, Học viện Công nghệ Bưu chính Viễn thông, Hà Nội, 2013.

[4]. ThS. Trần Thục Linh, ThS. Vũ Anh Đào, Bài giảng Mạng cảm biến, Học viện Công nghệ Bưu chính Viễn thông, Hà Nội, 2013

[5]. John Madieu, Linux Device Drivers Development, 2017

[6]. Mark Pedley, NXP, Tilt Sensing Using a Three-Axis Accelerometer

Danh mục các Website tham khảo:

1. https://www.kernel.org/doc/html/v4.14/input/event-codes.html

2. http://www.alldatasheet.com

3. <http://www.electronics-tutorials.ws>

4. https://www.kernel.org/doc/html/latest/input/input-programming.html

5. https://mosquitto.org/

6. <https://github.com/eclipse/paho.mqtt.c>