ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN VIỄN THÔNG

---------------o0o---------------



**LUẬN VĂN TỐT NGHIỆP**

Chuyên ngành: ĐIỆN TỬ - VIỄN THÔNG

**NGHIÊN CỨU XÂY DỰNG CHƯƠNG TRÌNH**

**ĐIỀU KHIỂN CAMERA TỪ XA QUA INTERNET**

|  |  |  |
| --- | --- | --- |
| CBHD: | TS Nguyễn Chí Ngọc |  |
| SVTH: | Nguyễn Đình Duy | 1510468 |
|  | Nguyễn Huỳnh Nhật Duy | 1510471 |

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2018

**DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN**

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số……………

ngày…………………………… của Hiệu trưởng Trường Đại học Bách Khoa TP.HCM.

1. …………………………………………………….. – Chủ tịch

1. .……………………………………………………. – Thư ký

3.…………………………………………………….. – Ủy viên

4. …………………………………………………….. – Ủy viên

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập – Tự Do – Hạnh Phúc

-----✩-----

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐH BÁCH KHOA

Thành phố Hồ Chí Minh

-----✩----- -----✩-----

Số: /BKĐT

Khoa: Điện – Điện tử

Bộ Môn: Viễn Thông

**NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

Họ và tên: **Nguyễn Huỳnh Nhật Duy**

**Nguyễn Đình Duy**

Ngành: **ĐIỆN TỬ VIỄN THÔNG**

1. **Đầu đề luận văn:** “**Nghiên cứu xây dựng chương trình điều khiển từ xa qua Internet**”.

2. **Nhiệm vụ (** Yêu cầu về nội dung và số liệu ban đầu):

..................................................................................................................................

..................................................................................................................................

3. **Ngày giao nhiệm vụ luận văn:**

4. **Ngày hoàn thành nhiệm vụ:**

5. **Họ và tên người hướng dẫn :**

**Phần hướng dẫn**

................................................................................................................................

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

Ngày ....... tháng .......... năm 2019

CHỦ NHIỆM BỘ MÔN NGƯỜI HƯỚNG DẪN CHÍNH

(*Ký và ghi rõ họ tên*) (*Ký và ghi rõ họ tên*)

PHẦN DÀNH CHO KHOA, BỘ MÔN:

Người duyệt (chấm sơ bộ):

Đơn vị:

Ngày bảo vệ:

Điểm tổng kết:

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐH BÁCH KHOA

Thành phố Hồ Chí Minh

-----✩----- -----✩-----

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập – Tự Do – Hạnh Phúc

-----✩-----

Khoa: **Điện – Điện tử** Ngày……. tháng….. năm 2019

Bộ Môn: **Viễn thông**

**PHIẾU CHẤM BẢO VỆ LVTN**

**(*Dành cho người hướng dẫn*)**

1. Họ và tên: **Nguyễn Huỳnh Nhật Duy -1510471**

**Nguyễn Đình Duy -1510467**

3. Ngành: **ĐIỆN TỬ VIỄN THÔNG**

4. Lớp: **DD15DV3**

5. Đề tài: “**Nghiên cứu xây dựng chương trình điều khiển từ xa qua Internet**”

6. Họ tên người hướng dẫn: **TS. NGUYỄN CHÍ NGỌC**

7. Tổng quát về bản thuyết minh:

Số trang ........ Số chương ........

Số bảng số liệu ........ Số hình vẽ ........

Số tài liệu tham khảo ........ Phần mềm tính toán ........

1. Tổng quát về các bản vẽ:

Số bản vẽ: bản A1 bản A2 khổ khác

Số bản vẽ tay : Số bản vẽ trên máy tính :

1. Những ưu điểm chính của LVTN:

...............................................................................................................................…

1. Những thiếu sót chính của LVTN:

..................................................................................................................................

1. Đề nghị:

Được bảo vệ ¨ Bổ sung thêm để bảo vệ ¨,

Không được bảo vệ ¨.

12.3 câu hỏi sinh viên trả lời trước Hội Đồng:

a)

b)

c)

1. Đánh giá chung (*bằng chữ: giỏi, khá, TB*): Điểm ………/10

Ký tên (*ghi rõ họ tên)*

***LỜI CẢM ƠN***

*Lời đầu tiên, em xin chân thành cảm ơn tiến sĩ Nguyễn Chí Ngọc, người thầy đã tận tình hướng dẫn, chỉ dạy cho em trong suốt thời gian qua để em có thể hoàn thành đề tài này.*

*Bên cạnh đó, em xin cảm ơn các thầy cô ở Bộ môn Viễn thông nói riêng và Khoa Điện – Điện tử, trường Đại học Bách Khoa Thành phố Hồ Chí Minh nói chung. Các thầy cô đã truyền đạt cho em nhiều kiến thức bổ ích, góp phần giúp em hoàn thiện đề tài.*

*Ban Giám đốc công ty Công ty cổ phần CNTM Ưu Việt t đã tạo mọi điều kiện thuận lợi cho em trong thời gian thực tập*

*Cuối cùng, em xin gửi lời cảm ơn đến gia đình và bạn bè, những người đã luôn bên cạnh động viên, giúp đỡ em trong công việc và cuộc sống.*

*Tp. Hồ Chí Minh, ngày 6 tháng 11 năm 2019*

**Sinh Viên Thực Hiện**

|  |  |
| --- | --- |
|  |  |

**TÓM TẮT ĐỀ TÀI**

Ngày nay, lĩnh vực công nghệ thông tin – truyền thông ngày càng phát triển, nhu cầu áp dụng y tế từ xa cũng ngày càng cao. Bên cạnh đó, việc giám sát và trao đổi hình ảnh từ xa thông qua Camera đã được ứng dụng tại nhiều quốc gia phát triển và đem lại những lợi ích to lớn trong việc chuẩn đoán cũng như tiết kiệm thời gian cho cả y bác sĩ lẫn bệnh nhân.

Nhận thấy những lợi ích thiết thực cũng như khả năng phát triển của Camera trong lĩnh vực y tế, tôi đã thực hiện nghiên cứu về khả năng cũng như giải pháp của ứng dụng việc quan sát từ xa Camera thông qua Internet. Từ đó, tôi đã ghi nhận, nghiên cứu và đưa ra định hướng phát triển ở mức độ đề cương cũng như định hướng phát triển cho luận văn, để có thể sử dụng như một sự lựa chọn nhẹ nhàng và hiệu quả trong các hệ thống y tế từ xa.

Chương trình mô phỏng được xây dựng với các tính năng hiển thị trình điều khiển đơn giản, thân thiện và có khả năng phát triển, với giao diện tiếng Việt, tích hợp tính năng giao tiếp với Camera, phục vụ việc trao đổi hình ảnh tại các bệnh viện.

Đề tài bao gồm có 7 chương:

Chương 1: Giới thiệu.

Chương 2: Giới thiệu về Camera Sony EVI D80.

Chương 3: Tổng quan kiến thức

Chương 4: Các giải pháp điều khiển Camera từ xa

Chương 5: Quá trình thực hiện

Chương 6: Kết quả thực hiện.

Chương 7: Kết luận và hướng phát triển.

Contents

1. CHƯƠNG 1: GIỚI THIỆU 15

1.1. Tầm quan trọng trong việc điều khiển Camera từ xa: 15

1.2 Tình hình nghiên cứu trong và ngoài nước: 16

1.3. Nhiệm vụ đề tài 16

1. CHƯƠNG 2 : GIỚI THIỆU VỀ RASPBERRY 17

2.1. Giới thiệu: 17

1. CHƯƠNG 3 : TỔNG QUAN KIẾN THỨC 20

3.1. Tìm hiểu về chuẩn giao tiếp RS422: 20

3.1.1. Tìm hiểu về chuẩn Visca: 20

3.1.1.1. Giới thiệu: 20

3.1.1.2. Truyền thông VISCA: 20

3.1.1.3. Tìm hiểu chuẩn RS422: 21

3.2. Giới thiệu về HTML và CSS: 23

3.2.1. Tổng quan về HTML: 23

3.2.1.1. Giới thiệu: 23

3.2.1.2. Giới thiệu về HTML5: 25

3.2.1.3. Những thành phần của HTML: 26

3.2.2. Tổng quan về CSS: 27

3.2.2.1. Giới thiệu: 27

3.2.2.2. Lịch sử hình thành: 28

3.2.2.3. Ưu nhược điểm của CSS: 28

3.3. Web conference: 32

3.3.1. Ưu nhược điểm của CSS: 32

3.3.2. Lịch sử: 32

3.3.3. Các đặc điểm cơ bản của web conference: 34

3.3.4. Ưu điểm và nhược điểm: 35

3.4. Websocket: 36

3.4.1. Giới thiệu: 36

3.4.2. Lịch sử: 36

3.4.3. Cách hoạt động: 37

3.4.4. Ưu điểm: 41

3.4.5. Nhược điểm: 42

CHƯƠNG 4: CÁC GIẢI PHÁP ĐIỀU KHIỂN CAMERA TỪ XA 47

4.1.Các giải pháp điều khiển camera từ xa: 47

* 1. Giới thiệu phác thảo phương hướng: 48
  2. Giải pháp phần cứng: 57
  3. Giải pháp phần mềm: 59
     1. Truyền lệnh: 60
  4. So sánh các giải pháp: 64

1. CHƯƠNG 5: QUÁ TRÌNH THỰC HIỆN 69
   1. Giải pháp và lựa chọn giải pháp phù hợp: 69
   2. Các chức năng cần có: 71
   3. Xây dựng server: 71
   4. Xây dựng client: 77
   5. Sơ đồ giải thuật: 78
      1. .Truyền lệnh: 78
      2. Truyền hình ảnh: 79
2. CHƯƠNG 6 :KẾT QUẢ THỰC HIỆN 80
   1. Truyền lệnh: 80
   2. Truyền hình ảnh: 81
3. CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 82
   1. Kết Luận 82
   2. Hướng phát triển: 83

**MỤC LỤC**

1. **DANH SÁCH HÌNH MINH HỌA**

[Hình 11. Bảng điều khiển Camera và giá tham khảo](#bookmark) 3

[Hình 21. Camera Sony EVI D80](#bookmark1) 6

Hình 3-1. Cấu trúc truyền thông VISCA…………………………………………………8

[Hình 32. Cấu trúc gói dữ liệu](#bookmark2) 9

[Hình 33.](#bookmark3) Cho thấy luồng dữ liệu được truyền……………………………………..……9

[Hình 34 Các chân trong kết nối](#bookmark4) 10

[Hình 35. Sơ đồ chân giữa hai thiết bị dùng RS422](#bookmark5) 11

[Hình 36. Giới thiệu về HTML5](#bookmark6) 12

[Hình 37. Các thẻ trong HTML5](#bookmark7) 15

[Hình 38.](#bookmark8) Giới thiệu CSS………………………………………………………………..15

[Hình 39. Giao diện trang Web nếu không có CSS](#bookmark9) 18

[Hình 3-10.](#bookmark10)  Ví dụ phân bố các thành phần bằng CSS trên trang web …………………..19

[Hình 311. Sự sai lệch trên các trình duyệt khác nhau của CSS](#bookmark11) 21

[Hình 3-12.](#bookmark12)  Giao thức truyền tin của websocket ……………………………….............27

[Hình 313. Logo của BigBlueButton](#bookmark13) 33

[Hình 314. Kiến trúc BigBlueButton](#bookmark14) 28

[Hình 315. Kiến trúc BigBlueButton Apps 3](#bookmark15)8

Hình 3-16. Tham dự hội nghị video……………………………………………………..39

Hình 3-17. Tải lên bài thuyết trình trong BigBlueButton………………………………40

Hình 3-18. Bigbluebutton Client………………………………………….......……41

Hình 3-19. Kiến trúc Web RTC…………………………………………..………..43

Hình 4-1. Mô hình phác thảo hệ thống thực hiện ……………………….…………48

Hình 4-2. Cáp chuyển đổi USB-RS422…………………………………………….49

Hình 4-3. Bảng điều khiển thiết kế tham khảo…………………...………………...50

Hình 4-4. Bảng tập lệnh của Camera tham khảo……………………………....…...51

Hình 4-5. Dây nối truyền lệnh……………………………………………………...52

Hình 4-6. Dây nối truyền hình ảnh…………………………………………….…...52

Hình 4-7. Joystick điều khiển camera………………………………...………….…53

Hình 4-8. Bảng giá bộ điều khiển camera từ xa………………………….…….…...53

Hình 4-9. Sơ đồ điều khiển camera từ xa………………………………….…….….54

Hình 4-10. Bảng lệnh điều khiển camera Evi-D80N……………………….………55

Hình 4-11.Giao thức truyền tin của websocket……...………………………..…….56

Hình 4-12.Giao thức truyền tin của HTTP………………………………….………57

Hình 4-13. Giao thức truyền tin Long-pollin………………………………….…….58

Hình 4-14. Màn hình OBS…………………………………………………….…….59

Hình 4-15. Chọn nguồn trong giao diện OBS………………………………….……60

Hình 4-16. Giao diện OBS sau khi chọn nguồn là desktop…………………….……61

Hình 4-17.Bảng tùy chỉnh âm thanh của OBS……………………………………….62

Hình 4-18. Giao diện setiing của OBS……………………………………………….62

Hình 5-1 Giao diện điều khiển camera của một phần mềm được tự xây dựng……….72

Hình 5-2.Truyền tin phía server………………………………………………………76

Hình 5-3.Giao diện trang web được xây…………………………...…………………78

Hình 7-1. Dịch vụ Google Cloud Platform………………...………………………….85

Hình 7-2. Các dịch vụ và chức năng của Google Cloud Platform……………………..86

Hình 7-3.Mô hình Static Nat………………………………………….………………87

**DANH SÁCH THUẬT NGỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Thuật ngữ tiếng Anh** | |
| 3G | Third Generation (network) |
| 4G | Fourth generation (network) |
| ACR | American College of Radiology |
| ADSL | Asymmetric Digital Subscriber Line |
| PTZ | Pan Tilt Zoom |
| DSR | Data Set Ready |
| DTR | Data Terminal Ready |
| DTE | Data Terminal Equipment |
| DCE | Data Circuit-Terminating Equipment |
| TTL | Transistor-transistor logic |
| TDE | Data Terminal Equipment |
| XML | eXtensible Markup Language |
| XSLT | [XSL](https://vi.wikipedia.org/w/index.php?title=XSL&action=edit&redlink=1) Transformations |
| HTML | Hyper Text Markup Language |
| XHTML | EXtensible Hyper Text Markup Language |
| UAC | User Account Control |
| OBS | Open Broadcaster Software |
| UI | User Interface |
| W3C | World Wide Web Consortium |
| IE | Internet Exploler |
| VoIP | Voice Over Internet Protocol |
| PC | Personal Computer |
| WHATWG | Web Hypertext Application Technology Working Group |
| CSS | Cascading Style Sheet |
| LMS | Learning Management System |
| PDF | Portable Document Format |
| AEC | Acoustic Echo Canceler |
| NR | Noise Reduction |
| API | Application Programming Interface |
| HTTP | Hypertext Transfer Protocol |
| IETF | Internet Engineering Task Force |
| TCP | Transmission Control Protocol |
| URL | Uniform Resource Locator |

|  |  |
| --- | --- |
| **Thuật ngữ tiếng Việt** | |
| BHYT | Bảo Hiểm Y Tế |
| CNTT | Công Nghệ Thông Tin |
| CSDL | Cơ Sở Dữ Liệu |

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. Tầm quan trọng trong việc điều khiển Camera từ xa:

Loài người đang trải qua một trong những thời kì cực thịnh nhất của chính mình. Theo thống kê của Liên hợp quốc, dân số thế giới đến ngày 22/3/2017 là 7,49 tỷ người. Ước tính dân số thế giới dự kiến đạt 8 tỷ người vào năm 2023 và đạt 10 tỷ người năm 2056. Kèm theo đó là những vấn đề do chính con người tạo ra: ô nhiễm môi trường, thực phẩm bẩn, rác thải v.v, khiến cho sức khỏe con người ngày càng bị đe dọa nghiêm trọng. Song hành với vấn đề này là sự già hóa dân số thế giới ở đa số các quốc gia phát triển cũng như đang phát triển, trong đó có Việt Nam. Từ đó lĩnh vực y tế ngày càng khẳng định địa vị quan trọng của nó trong xã hội loài người hiện đại, chăm sóc sức khỏe trở thành một nhu cầu thiết yếu cho mỗi con người. Để đáp ứng cho việc chăm sóc sức khỏe cho một lượng lớn dân số như vậy, việc ứng dụng các công nghệ hiện đại vào y tế là rất cần thiết, đặc biệt là công nghệ thông tin và truyền thông trong y tế.

Từ những năm 70 của thế kỷ trước, thuật ngữ “*telemedicine*” – “*y tế từ xa*” được ra đời để chỉ các ứng dụng của công nghệ thông tin trong việc khám chữa bệnh cũng như chăm sóc sức khỏe. Hiện nay đa số quốc gia tiên tiến cũng như đang phát triển đã và đang áp dụng các hệ thống y tế từ xa trong việc xử lý hình ảnh, hội chẩn, chẩn đoán, đơn thuốc, …

Y tế từ xa có nhiều lợi ích thiết thực mà ta có thể thấy được như:

* Kết nối các bệnh viện ở thành thị và nông thôn với nhau, góp phần trong việc khám chữa bệnh kịp thời.
* Tốn ít chi phí hơn cho việc đầu tư vào cơ sở vật chất, hạ tầng tại các bệnh viện.
* Hỗ trợ đắc lực trong việc đào tạo từ xa cho đội ngữ y bác sĩ tại các bệnh viện các tuyến dưới.

Tuy nhiên, y tế từ xa cũng có một số bất cập như: phụ thuộc vào hạ tầng viễn thông – thông tin, bất đồng ngôn ngữ khi liên thông bệnh viện giữa các quốc gia, …

Dữ liệu y tế là một phần không thể thiếu trong y tế. Và để phục vụ cho y tế từ xa, việc giám sát y tế từ xa thông qua Camera đã trở thành một xu hướng mới để phát triển.

## 1.2 Tình hình nghiên cứu trong và ngoài nước:

Trên thế giới đã có nhiều quốc gia áp dụng các hệ thống giám sát từ xa thông qua Camera vào lĩnh vực y tế. Ứng dụng trọng điểm có thể kể đến là quan sát hoặc ghi hình từ phòng phẫu thuật đến phòng họp để trao đổi, chỉ đạo và ứng dụng vào công tác giảng dạy thực tiễn cho sinh viên.

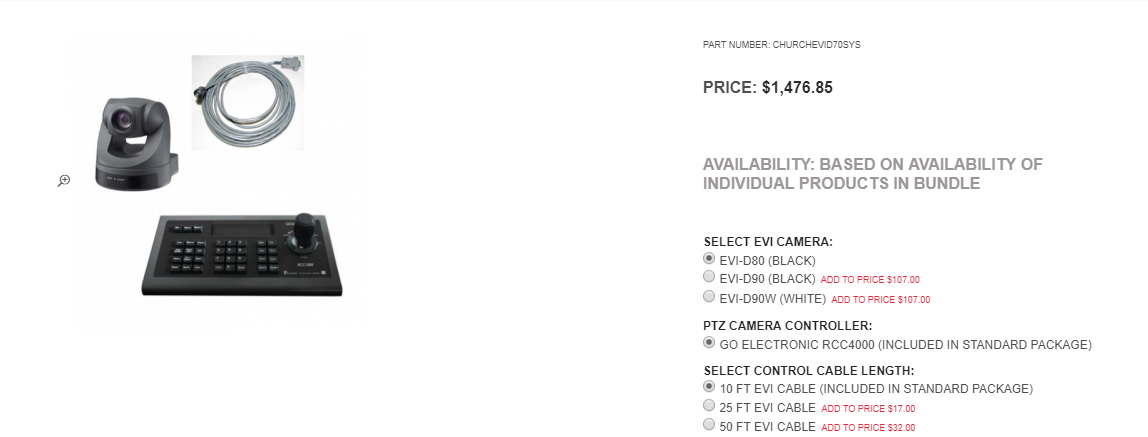
Tại Việt Nam, hạ tầng viễn thông – thông tin đã tương đối hoàn thiện với gần như 100% các cơ sở y tế có máy tính kết nối mạng. Hệ thống mạng trong nước cũng phát triển đa dạng từ ADSL, FTTH, 3G, 4G, là điều kiện rất thuận lợi cho việc khiển khai giám sát từ xa.

Tính đến nay, nước ta đã có những thành tựu đáng kể trong việc áp dụng công nghệ thông tin vào lĩnh vực y tế. Điển hình là trong 6 tháng đầu năm 2017, việc ứng dụng công nghệ thông tin và giám sát trong quản lý khám chữa bệnh và thanh toán Bảo hiểm y tế đã kết nối liên thông hệ thống thông tin y tế giữa 63 Sở Y tế, 63 cơ quan Bảo hiểm xã hội, 1.356 bệnh viện các tuyến, 704 cơ quan Bảo hiểm xã hội quận/huyện, 704 Trung tâm y tế huyện, 710 trung tâm y tế cơ quan xí nghiệp và 11.105 Trạm y tế xã, phường trên cả nước. Các dự án về y tế từ xa cũng đã được triển khai tại nhiều bệnh viện, điển hình là đề án Bệnh viện vệ tinh của Bệnh viện Chợ Rẫy, Bệnh viện Việt – Đức, … (triển khai từ năm 2013), hệ thống Telemedicine tại Bệnh viện Quân y 175, bệnh viện Trung ương quân đội 108.

## 1.3. Nhiệm vụ đề tài

Để phục vụ cho y tế từ xa, hệ thống giám sát bằng Camera là một trong những hệ thống quan trọng, vấn đề chia sẻ ảnh y tế trong ca mổ và giữa các bệnh viện ngày trở nên cấp thiết. Do đó, việc hình ảnh được chia sẻ càng nhanh và trực quan sẽ giúp cho ca mổ thành công hơn về cả kết quả cho bệnh nhân lẫn kinh nghiệm cho y bác sỹ. Và để có thể truyền dẫn hình ảnh, cần phải có một hệ thống chuyên biệt mà quan trọng nhất trong đó là thiết bị lấy hình ảnh mà ở đây là Camera.

Hiện nay do nhu cầu sử dụng trải rộng mà thị trường Camera cũng phát triển rất mạnh. Nhưng do đặc thù về lĩnh vực y tế yêu cầu sự rõ ràng và chính xác cao mà Camera được sử dụng cũng phải đảm bảo những yêu cầu này. Các Camera phân khúc tầm cao cho chất lượng hiện thị cũng như khả năng sử dụng ổn định rất thích hợp dùng vào y tế. Nhưng một vấn đề bức thiết đặt ra là các dụng cụ đi kèm để điều khiển từ xa cũng có chi phí không hề dễ chịu. Điển hình như Camera sẽ được sử dụng trong luận văn sắp tới là Sony EVI D80.



Hình 1. 1.Bảng điều khiển Camera và giá tham khảo

Chi phí cao , lợi ích không quá lớn nhưng khuyết điểm rất nghiêm trọng là bảng điều khiển không thể hỗ trợ cho khoảng cách quá xa. Ví dụ từ bệnh viện đến bệnh viện khác hoặc nhiều nơi cùng muốn theo dõi một ca phẫu thuật.

Thay vào đó, tận dụng các chuẩn kết nối dữ liệu hỗ trợ sẵn trên Camera là RS 232 và RS422, ta có thể xây dựng các chương trình điểu khiển từ xa thông qua chuẩn kết nối này và phát triển tối ưu hơn trong tương lai như điều khiển thông qua Web, chia sẻ dữ liệu đồng thời cho nhiều bên…

Vì lí do này, tôi tập trung phát triển để tài mục đích xây dựng nên một phần mềm điều khiển từ xa nhằm tối ưu hóa khả năng linh hoạt cũng như chủ động trong việc điều khiển Camera. Đồng thời tối thiểu hóa chi phí từ đó có thể mở rộng khả năng phát triển trong ngành y tế nước nhà.

# CHƯƠNG 2 : GIỚI THIỆU VỀ RASPBERRY



## 2.1. Giới thiệu:

Raspberry Pi là một máy tính bảng đơn nhỏ. Bằng cách kết nối các thiết bị ngoại vi như bàn phím, chuột, màn hình với Raspberry Pi, nó sẽ hoạt động như một máy tính cá nhân mini.

Raspberry Pi được sử dụng phổ biến cho xử lý hình ảnh / video thời gian thực, các ứng dụng dựa trên IoT và ứng dụng Robotics.

Raspberry Pi chậm hơn máy tính xách tay hoặc máy tính để bàn nhưng vẫn là một máy tính có thể cung cấp tất cả các tính năng hoặc khả năng mong đợi, với mức tiêu thụ điện năng thấp.

Raspberry Pi Foundation chính thức cung cấp hệ điều hành Raspbian dựa trên Debian. Ngoài ra, họ cung cấp hệ điều hành NOOBS cho Raspberry Pi. Nguời dùng có thể cài đặt một số phiên bản hệ điều hành của bên thứ ba như Ubuntu, Archlinux, RISC OS, Windows 10 IOT Core, v.v.

Hệ điều hành Raspbian là Hệ điều hành chính thức được cung cấp miễn phí. Hệ điều hành này được tối ưu hóa hiệu quả để sử dụng với Raspberry Pi. Raspbian có GUI bao gồm các công cụ để duyệt, lập trình Python, văn phòng, trò chơi, v.v.

Chúng ta nên sử dụng thẻ SD (khuyến nghị tối thiểu 8 GB) để lưu OS (Hệ điều hành).

Raspberry Pi không chỉ là máy tính vì nó cung cấp quyền truy cập vào phần cứng trên chip, tức là các GPIO để phát triển ứng dụng. Bằng cách truy cập GPIO, chúng tôi có thể kết nối các thiết bị như đèn LED, động cơ, cảm biến, v.v. và cũng có thể điều khiển chúng.

Nó có SoC xử lý Broadcom dựa trên ARM cùng với GPU trên chip (Bộ xử lý đồ họa).

Tốc độ CPU của Raspberry Pi thay đổi từ 700 MHz đến 1,2 GHz. Ngoài ra, nó có SDRAM trên bo mạch từ 256 MB đến 1 GB.

Raspberry Pi cũng cung cấp các mô-đun SPI, I2C, I2S và UART trên chip.

Có nhiều phiên bản raspberry pi trên thị trường hiện nay:

1.Raspberry Pi 1 Model A

2.Raspberry Pi 1 Model A+

3.Raspberry Pi 1 Model B

4.Raspberry Pi 1 Model B+

5.Raspberry Pi 2 Model B

6.Raspberry Pi 3 Model B

7.Raspberry Pi Zero

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Raspberry Pi 2**  **Model B** | **Raspberry Pi 3**  **Model B** | **Raspberry Pi zero** |  |  |
| **SoC** | BCM2835 | BCM2836 | BCM2837 | BCM2835 |
| **CPU** | ARM 11 | Quad Cortex A7 | Quad Cortex A53 | ARM 11 |
| **Operating Freq.** | 700 MHz | 900 MHz | 1.2 GHz | 1 GHz |
| **RAM** | 512 MB SDRAM | 1 GB SDRAM | 1 GB SDRAM | 512 MB SDRAM |
| **GPU** | 250 MHz Videocore IV | 250MHz Videocore IV | 400 MHz Videocore IV | 250MHz Videocore IV |
| **Storage** | micro-SD | Micro-SD | micro-SD | micro-SD |
| **Ethernet** | Yes | Yes | Yes | No |
| **Wireless** | WiFi and Bluetooth | No | No | No |

# CHƯƠNG 3 : TỔNG QUAN KIẾN THỨC

## 3.1. Tìm hiểu về chuẩn giao tiếp RS422:

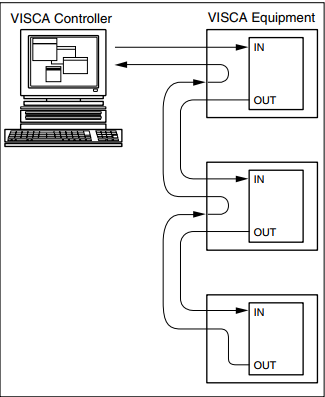
### 3.1.1. Tìm hiểu về chuẩn Visca:

3.1.1.1. Giới thiệu:

Cho ví dụ: một máy tính được gọi là bộ điều khiển khi có thiết bị nhận lệnh, chẳng hạn như: EVI-D80, được gọi là thiết bị ngoại vi thiết bị. EVI-D80 hoạt động như một thiết bị ngoại vi trong VISCA.

Trong VISCA, tối đa bảy thiết bị ngoại vi như EVI-D80N / D80P / D90N / D90P có thể được kết nối với một bộ điều khiển bằng cách sử dụng giao tiếp phù hợp với tiêu chuẩn RS-422.

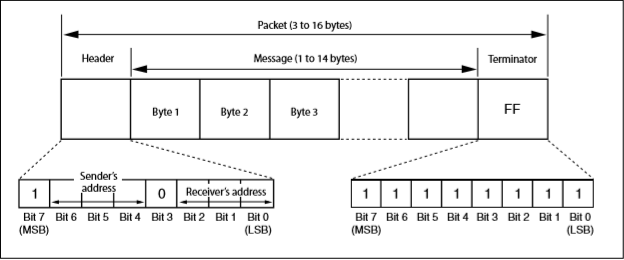
Mỗi thiết bị VISCA đều có VISCA IN và VISCA OUT. Đầu vào DSR (đầu ra DTR của bộ điều khiển) của VISCA IN khi điều khiển thiết bị VISCA từ bộ điều khiển

.

Hình 3.1 Sơ đồ truyền thông VISCA

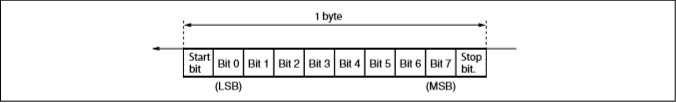
3.1.1.2. Truyền thông VISCA:

Đơn vị cơ bản của truyền thông VISCA được gọi là gói :



Hình 3. 2. Cấu trúc gói dữ liệu

Byte đầu tiên của gói được gọi là tiêu đề và bao gồm địa chỉ người gửi và người nhận, dấu kết thúc là FFH, nó biểu thị sự kết thúc của gói.



Hình 3.3 Cho thấy luồng dữ liệu được truyền

3.1.1.3. Tìm hiểu chuẩn RS422:

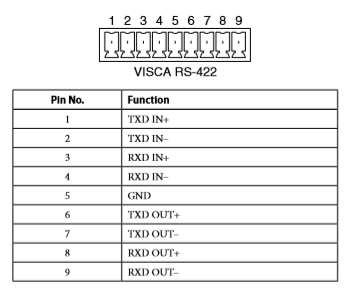
Khi cần truyền thông ở tốc độ cao và khoảng cách lớn như trong điều kiện nhà xưởng thực tiễn, phương pháp truyền thông SE với đại diện là chuẩn **RS 232** không đáp ứng nỗi những yêu cầu Kỹ Thuật đặt ra như độ tin cậy và độ an toàn. Phương pháp truyền thông theo kiểu sai biệt (tín hiệu sai biệt đường truyền cân bằng – Balanced Lines ) tỏ ra hiệu quả hơn trong hầu hết các ứng dựng. Tín hiệu sai lệch có thể loại trừ những ảnh hưởng bởi nhiễu xuất hiện trên mạng. Các tín hiệu nhiễu này gây ra bởi đóng ngắt có sự chêch áp, v.v.....

RS422 là chuẩn được thiết kế với tốc độ cao và khoảng cách truyền dẫn lớn hơn rất nhiều so với chuẩn RS232. Đơn giản nhất, ta thường gặp các thiết bị chuyển đổi ( Converter ) từ chuẩn RS232 sang chuẩn RS422 và ngược lại. Tốc độ truyền dữ liệu rất cao, lên đến 100kb/s và khoảng cách truyền đáng tin cậy lên đến 4000 Ft ( chừng 1,2 km). Với chuẩn RS422, đặc biệt có thể sử dụng ở dạng Multi-Drop (Party- Line); nghĩa là chỉ cần một thiết bị truyền dữ liệu đến nhiều thiết bị nhận khác trên một Bus. Một “Bus” như thế có thể có tối đa 10 thiết bị nhận. Nhược điểm của cách truyền Multi-Drop là không thể dùng trong mạng khi có nhiều thiết bị cần truyền và nhiều thiết bị nhận tín hiệu trên cùng một đường truyền (Single Bus). Người ta gọi loại mạng này là mạng đa điểm (Multi-point Network) .

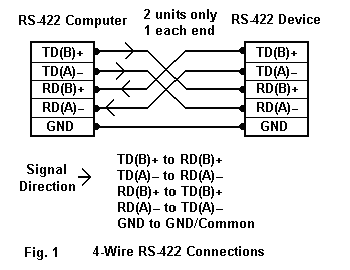
Chiều dài đường truyền và tốc độ tối đa:

* 40 Feet = 12m 10 Mbits/sec
* 400 Feet = 122m 1 Mbits/sec
* 4000 Feet = 1219m 100 kbits/sec

Với yêu cầu thực tế để phát triển cho Camera EVI D80, chuẩn kết nối RS422 được nhà sản xuất khuyến khích dùng hơn RS232C và lưu ý không kết nối đồng thời cả hai cổng cùng lúc.



Hình 3.4 các chân trong đầu kêt nối



Hình 3.5. Sơ đồ đấu chân giữa hai thiết bị dùng RS422

## 3.2. Giới thiệu về HTML và CSS:

### 3.2.1. Tổng quan về HTML:

3.2.1.1. Giới thiệu:

HTML là viết tắt của Hyper Text Markup Language, là ngôn ngữ được sử dụng rộng rãi nhất trên Web để phát triển các trang web. HTML được tạo ra bởi Berners-Lee vào cuối năm 1991 nhưng đến năm 1995 với sự ra đời của phiên bản "HTML 2.0" mới được công nhận là bản tiêu chuẩn đầu tiên. Hiện nay phiên bản HTML 4.01 là phiên bản chính của HTML và được xuất bản vào cuối năm 1999. Bên cạnh đó, phiên bản HTML 5 hiện nay là phần mở rộng của bản HTML 4.01 được tung ra thị trường năm 2012 và vẫn đang tiếp tục được phát triển.

Ban đầu, **HTML** được phát triển với mục đích xác định cấu trúc của các tài liệu như tiêu đề, đoạn văn, danh sách, v.v để chia sẻ thông tin giữa những người sử dụng. Theo thời gian, bằng những ưu điểm của mình, HTML được sử dụng rộng rãi để định dạng các trang web với sự trợ giúp của các thẻ khác nhau có sẵn bằng ngôn ngữ HTML.

Các ứng dụng của HTML có thể kể đến như :

* **Phát triển trang web** - HTML được sử dụng để tạo các trang được hiển thị trên web. Hầu như mọi trang web đều có các thẻ html trong đó để hiển thị chi tiết trong trình duyệt.  Các thẻ HTML được sử dụng để tạo các tài liệu HTML và hiển thị thuộc tính của chúng, các thẻ khác nhau sẽ có các thuộc tính khác nhau.
* **Điều hướng Internet** - HTML cung cấp các thẻ được sử dụng để điều hướng từ trang này sang trang khác và được sử dụng nhiều trong điều hướng internet.
* **UI Responsive** - Các trang HTML hiện nay hoạt động tốt trên tất cả các nền tảng, thiết bị di động, tab, máy tính để bàn hoặc máy tính xách tay nhờ tối ưu đáp ứng cho từng nền tảng .
* **Các** **trang HTML** **hỗ trợ ngoại tuyến** sau khi tải có thể được cung trên máy mà không cần internet.



Hình 3.6. Giới thiệu về HTML

* Người phát triển HTML là **Tim Berners-Lee** đồng thời cũng là cha đẻ của World Wide Web và chủ tịch của **World Wide Web Consortium** (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet). Năm 1989, ông nghiên cứu ra  ngôn ngữ HTML như là một giao thức truyền đạt thông tin giữa giới khoa học với nhau. Về phương thức HTML kết hợp các đoạn văn bản (text) với cấu trúc code để từ đó quy định cách các text đó thể hiện.Ở phiên bản đầu tiên HTML không cho phép truyền đạt cấu trúc trang phức tạp, nhưng vừa đủ để cho phép tạo ra một trang web đơn giản
* Năm 1993 các chuẩn HTML trở thành nền tảng của Mosaic- Trình duyệt đầu tiên của Internet.
* Dần dần sức mạnh của HTML được công nhận,kéo theo đó là sự phát triển tràn lan của ngôn ngữ HTML và những rắc rối có thể xảy ra trong tương lai, tổ chức World Wide Web Consortium (W3C) đã đi đến thống nhất và phê duyệt bộ ngôn ngữ HTML chuẩn.  Tháng 9 năm 1995, bộ HTML 2.0 được ra đời
* Năm 1997, bộ HTML 3.2 được tung ra. Đến mùa xuân năm 1998 chuẩn HTML lại tiếp tục được nâng cấp lên 4.0

Cùng với sự phát triển của Internet, HTML 4.0 trở nên lỗi thòi, W3C tiếp tục đưa ra phiên bản 4.1 và 4.2 được gọi là XHTML (HTML + XML). Tuy nhiên việc này chỉ khiến HTML trở nên khó hiểu hơn so với ban đầu. Năm 2012, WHATWG ( Web Hypertext Application Technology Working Group), một nhóm nghiên cứu khác, đã tạo ra HTML5, đơn giản và mạnh mẽ hơn. Sau này W3C đã chọn HTML5 làm tiêu chuẩn cho Web.

3.2.1.2. Giới thiệu về HTML5:

HTML5 là sự phát triển mạnh mẽ nhất của HTML. Phiên bản đầu tiên được hình thành trong năm 2008 nhưng chỉ đến năm 2012 mới chính thức được ra mắt công chúng. Tuy nhiên, tại lúc này rất nhiều trình duyệt Web chưa kịp thay đổi để hỗ trợ nên trên thực tế vào thời điểm đó HTML5 chưa đi vào thực tế người sử dụng .Theo đà phát triển mạnh mẽ, ngày nay hầu hết các trình duyệt phổ biến như Chrome, Firefox, Opera, IE … đều đã hỗ trợ HTML5.

Cùng với CSS3, HTML5 vẫn tiếp tục được phát triển và đạt được nhiều thành công hơn.Theo đà phát triển thì HTML5 kết hợp với CSS3 ngày càng đi đến gần với một ngôn ngữ lập trình mạnh.

HTML5 là sự kế thừa của HTML 4.01 được sáng tạo dựa theo các tiêu chuẩn như:

* Các tính năng mới phải dựa trên HTML, CSS, DOM, và JavaScript
* Làm giảm nhu cầu cho các plugins bên ngoài (như Flash)
* Xử lý lỗi tốt hơn
* Đánh dấu để thay thế kịch bản
* HTML5 nên phát triển cho các thiết bị độc lập
* Quá trình phát triển nên được công khai nhằm cho các tổ chức khác có thể lấy làm cơ sở phát triển.

Tính năng mới:

* Mã hóa dễ dàng và hợp lý hơn.
* Có các tính năng mới trong tiến trình xử lý đa phương tiện như < video >, < audio >,< canvas > ; tích hợp các nội dung đồ họa vector. Điều này có nghĩa là media và vector trên website được xử lý và thực hiện dễ dàng hơn, nhanh hơn mà ko cần phải bổ sung them các phần mềm liên quan hoặc thư viện API khác.

3.2.1.3. Những thành phần của HTML:

Các dạng thẻ HTML :

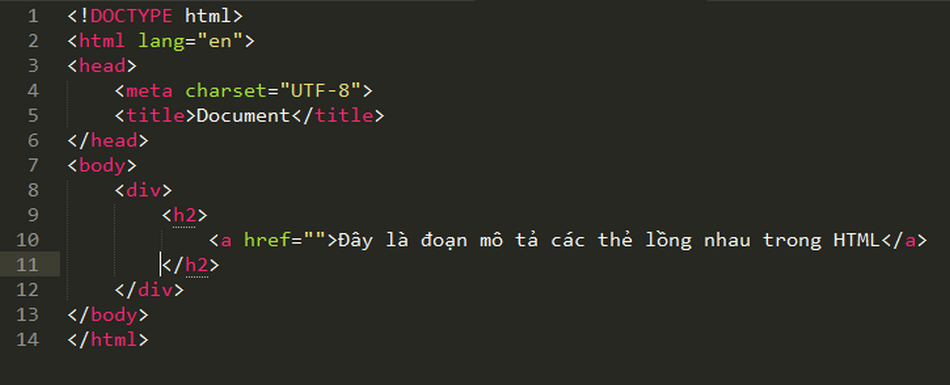
* Thẻ HTML dùng để viết lên những thành tố HTML
* Thẻ HTML được bao quanh bởi hai dấu lớn hơn < và > nhỏ hơn và thường có một cặp giống như <b> và </b>
* Thẻ thứ nhất là thẻ mở đầu và thẻ thứ hai là thẻ kết thúc.
* Dòng chữ ở giữa hai thẻ bắt đầu và kết thúc là nội dung.
* Những thẻ HTML không phân biệt in hoa và viết thường.

Ví dụ dạng <b> và <B> đều như nhau

* Dòng chữ ở giữa hai thẻ bắt đầu và kết thúc là nội dung.

Thành phần HTML

* Thành phần của HTML bắt đầu với thẻ: <b>
* Nội dung của nó là: web design resources
* Thành phần của HTML kết thúc với thẻ: </b>
* Mục đích của thẻ <b> là để xác định một thành phần của HTML phải được thể hiện dưới dạng in đậm
* Phần thân bắt đầu bằng thẻ bắt đầu <body> và kết thúc bằng thẻ kết thúc </body>. Mục đích của thẻ <body> là xác định thành phần của HTML bao gồm nội dung của tài liệu.



Hình 3.7. Các thẻ trong HTML

### 3.2.2. Tổng quan về CSS:

3.2.2.1. Giới thiệu:



Hình 3.8.giới thiệu về CSS

**CSS** là viết tắt củaCascading Style Sheet là ngôn ngữ tạo phong cách cho trang web được sử dụng để tạo phong cách và định kiểu cho những thành phần được viết dưới dạng ngôn ngữ đánh dấu, như là [HTML](https://www.hostinger.vn/huong-dan/html-la-gi/). CSS có thể điều khiển định dạng của nhiều trang web cùng lúc để tối ưu hoạt động cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.  
 **CSS** được phát triển bởi **W3C** năm 1996 để khắc phục nhược điểm lúc bấy giờ của HTML là không được thiết kế để gắn tag để giúp định dạng trang web. Người dùng chỉ có thể dùng nó để “đánh dấu” lên site. Những tag như **<font>**được ra mắt trong HTML phiên bản 3.2 gây nên rất nhiều rắc rối vì website có thể có nhiều font, màu nền và phong cách khác nhau. Vì vậy thay vì phải viết code lại cho một trang Web – vốn là hành động rất tốn thời gian thì CSS được tạo ra để giải quyết vấn đề này.

CSS về lý thuyết có thể không cần thiết, nhưng khi đó website sẽ chỉ đơn thuần là một trang chứa văn bản rất khô khan và không có gì khác.

3.2.2.2. Lịch sử hình thành:

CSS lần đầu tiên được đề xuất bởi Håkon Wium Lie vào ngày 10 tháng 10 năm 1994 khi đang làm việc với Tim Berners-Lee tại CERN

CSS đầu tiên để trở thành Khuyến nghị chính thức của W3C là CSS cấp 1, được xuất bản vào ngày 17 tháng 12 năm 1996. Håkon Wium Lie và Bert Bos được ghi nhận là nhà phát triển ban đầu. Nó có những đặc điểm như:

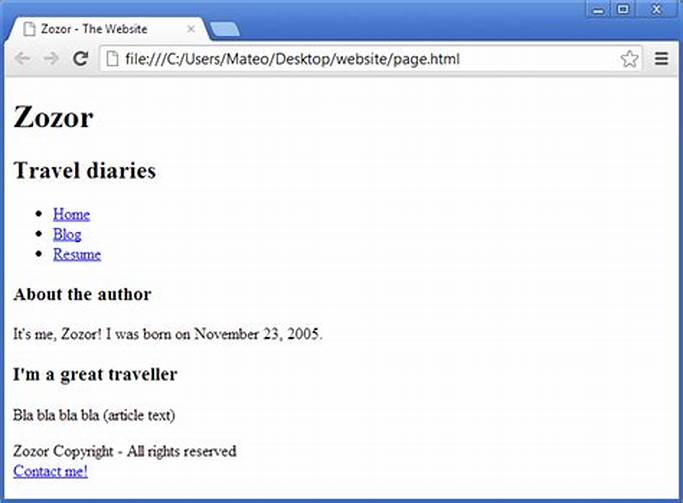
* Có thuộc tính phông chữ như: kiểu chữ và nhấn mạnh chữ
* Tùy chỉnh được màu của văn bản, hình nền và các yếu tố khác
* Các thuộc tính văn bản như khoảng cách giữa các từ, chữ cái và dòng văn bản
* Có thể linh hoạt sắp xếp văn bản, hình ảnh, bảng
* Có thể căn lề, viền, đệm và định vị cho hầu hết các yếu tố
* Nhận dạng duy nhất và phân loại chung các nhóm thuộc tính

CSS cấp 2 được [W3C](https://www.w3.org/) phát triển vào tháng 5 năm 1998. Nó là phiên bản nâng cấp từ CSS1, ngoài ra CSS 2 còn bao gồm một số khả năng mới  Sau đó CSS 2.1 được nâng cấp từ bản 2.0, sửa lỗi loại bỏ những tính năng kém hoặc không tương thích đầy đủ cho người dùng. Trong khoảng thời gian từ năm 2004 đến 2010 chúng được sửa đổi liên tục. Cuối cùng, CSS 2.1 được đề xuất vào ngày 12 tháng 4 năm 2011.

Bản nháp CSS 3 sớm nhất đã được xuất bản vào tháng 6 năm 1999. CSS3 là phiên bản thay thế cho CSS2 . Nó giới thiệu các bộ chọn và thuộc tính mới cho phép linh hoạt hơn với bố cục và trình bày trang. Một số cập nhật, chẳng hạn như thuộc tính bóng hộp (cho phép thêm bóng đổ vào một phần tử), cho phép áp dụng hiệu ứng hình ảnh mà không cần tạo hình ảnh đặc biệt.

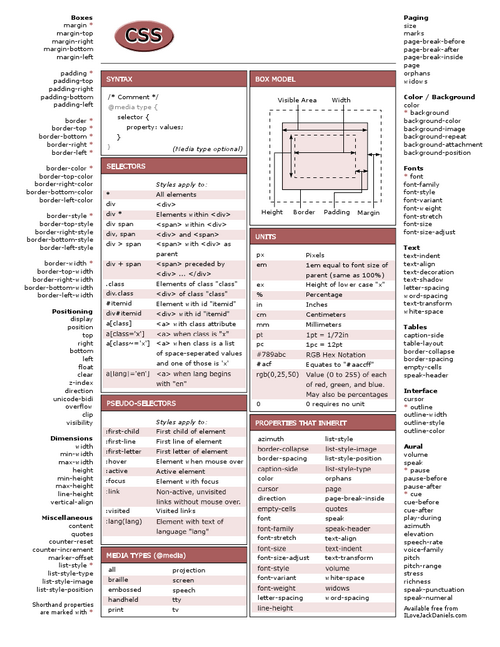
3.2.2.3. Ưu nhược điểm của CSS:

* Ưu điểm:
  1. Tiết kiệm băng thông (bandwith) : CSS được dùng nhiều bởi các lập trình viên để định dạng cấu trúc của webpage hơn dạng table layout vì ưu điểm giảm dung lượng webpage, qua đó tiết kiệm băng thông đáng kể
  2. Bổ trợ cho HTML :  CSS kết hợp với HTML giúp webpage có tính kỹ thuật hơn và giao diện đẹp hơn. Ngoài ra, trong một trang HTML sẽ có rất nhiều các thành phần khác nhau như BODY, HEADER, DIV, HEADLINE… Khi sử dụng, CSS sẽ giúp sắp xếp các thành phần này một cách khoa học, logic và dễ theo dõi hơn.



Hình 3.9. Giao diện trang Web nếu không có CSS

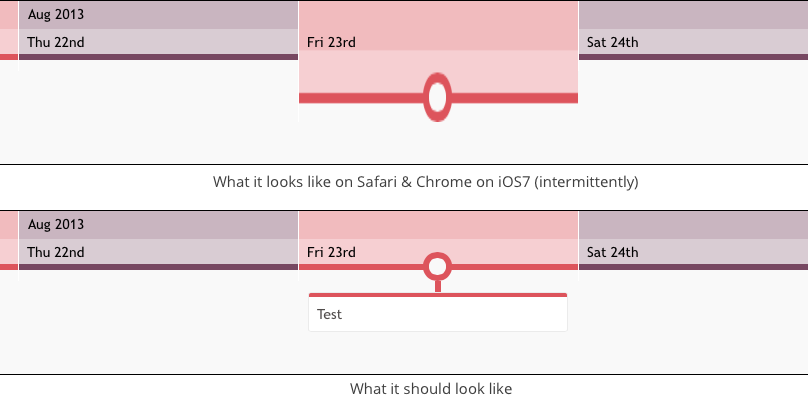
* 1. Có thể “đặt” các đối tượng ở bất cứ vị trí nào trên webpage : CSS giúp lập trình viên có thể “đặt” các đối tượng text, ảnh, form, bảng, flash… ở bất cứ vị trí nào, giúp dễ dàng trong việc phân bố các mảng và giảm rủi ro trong việc bảo trì trang web.f



Hình 3.10. Ví dụ phân bố các thành phần bằng CSS trên trang web

* 1. CSS tương thích với hầu hết các trình duyệt : CSS được hỗ trợ bởi hầu hết các trình duyệt. Khi xem webpage trên các trình duyệt khác nhau, nội dung hiển thị và giao diện webpage có thể được thể hiện nhất quán.
  2. Hỗ trợ cho việc in ấn webpage : CSS giúp tạo ra một giao diện thân thiện và dễ dàng in ấn. Các thuộc tính màu sắc trong CSS hỗ trợ tên màu hoặc mã màu dưới dạng mã hệ thập lục phân (hexadecimal code), giúp thể hiện màu sắc chính xác và an toàn.
  3. Hỗ trợ tối đa việc tùy biến webpage : ho phép người dùng thay đổi layout hoặc giao diện của webpage màkhông làm ảnh hưởng tới nội dung.
  4. Hỗ trợ các công cụ tìm kiếm : CSS sẽ giúp loại bỏ code thừa, những đoạn code được lặp lại nhiều lần hay các thuộc tính không quan trọng được tự động sinh ra bởi công cụ lập trình.
  5. Giúp các webpage có sự đồng bộ tuyệt đối, dễ dàng nâng cấp giao diện : Cho dù website đucợ thiết kế có 100 trang hay 1000 trang thì lập trình viên chỉ phải định dạng trong một file CSS duy nhất. Khi muốn nâng cấo giao diện website, lập trình viên chỉ phải sửa một lần duy nhất file CSS, toàn bộ giao diện các webpage sẽ tự động thay đổi.
* Nhược điểm :

1. CSS không chứa biến : Điều này làm cho CSS khó khăn khi phải thay đổi vì đặc tính làm một "thay thế tất cả" khi mong muốn thay đổi một hằng số cơ bản, chẳng hạn như màu sắc hoặc độ cao khác nhau và độ rộng một đối tượng cụ thể.
2. Hỗ trợ trình duyệt không nhất quán :  Các trình duyệt khác nhau sẽ làm cho CSS bố trí khác nhau như là một kết quả của lỗi trình duyệt hoặc thiếu sự hỗ trợ cho các tính năng CSS.



Hình 3.10. Sự sai lệch trên các trình duyệt khác nhau của CSS

1. Thiếu mô tả Cột : Trong khi có thể trong CSS hiện hành, bố trí với nhiều cột có thể phức tạp để thực hiện.

## 3.3. Web conference:

### 3.3.1. Ưu nhược điểm của CSS:

Web conference là một dịch vụ trực tuyến mà người dùng có thể tổ chức các cuộc họp, hội nghị, thuyết trình và đào tạo trực tiếp qua internet, đặc biệt là trên các kết nối TCP / IP. Người dùng có thể kết nối với hội nghị bằng điện thoại hoặc sử dụng loa và micrô máy tính thông qua kết nối VoIP.

Điểm gia nhập luôn luôn là một trình duyệt web và một liên kết đến một sự kiện. Thiết bị đầu cuối thường được đại diện bằng một trình duyệt web riêng ([sử dụng công nghệ WebRTC](http://trueconf.com.vn/tinh-nang/tinh-nang-he-thong/webrtc) được) hoặc bên thứ ba plug-in và ứng dụng sẽ tự động kết nối với một sự kiện sau khi cài đặt.

### 3.3.2. Lịch sử:

Các phương tiện trò chuyện bằng văn bản thời gian thực như IRC xuất hiện vào cuối những năm 1980. Phần mềm trò chuyện và nhắn tin tức thì dựa trên web xuất hiện vào giữa những năm 1990. Hệ thống học máy tính PLATO cho phép sinh viên cộng tác trên các máy tính nối mạng để hoàn thành các nhiệm vụ học tập ngay từ những năm 1960, nhưng việc kết nối mạng ban đầu không được thực hiện thông qua World Wide Web và các mục tiêu hợp tác của PLATO không phù hợp với điển hình của người thuyết trình hệ thống hội nghị. PLATO II, vào năm 1961, đã giới thiệu hai người dùng cùng một lúc.

Năm 1992, InSoft Inc. đã ra mắt Communique, một sản phẩm hội nghị truyền hình Unix dựa trên phần mềm dành cho các máy trạm cho phép hội thảo video / âm thanh / dữ liệu. Communique hỗ trợ tới 10 người dùng và bao gồm các tính năng mang tính cách mạng như chia sẻ ứng dụng, điều khiển âm thanh, văn bản, đồ họa và bảng trắng cho phép người dùng nối mạng chia sẻ và thao tác với các đối tượng đồ họa và tệp bằng các công cụ vẽ đơn giản.

Một số sản phẩm hội nghị truyền hình điểm-điểm và mạng riêng được giới thiệu vào những năm 1990, chẳng hạn như CU-SeeMe, được sử dụng để liên kết các trường được chọn trên khắp Hoa Kỳ trong các liên lạc hợp tác thời gian thực như một phần của dự án Global Schoolhouse từ Global SchoolNet.

Vào tháng 5 năm 1995, PictureTel đã công bố LiveShare Plus là một sản phẩm cộng tác dữ liệu sử dụng chung cho các máy tính cá nhân dựa trên Windows. Phần mềm cho phép chia sẻ ứng dụng, kiểm soát do người dùng cấp cho PC từ xa, đánh dấu bảng trắng được chia sẻ, truyền tệp và nhắn tin văn bản. Giá niêm yết được đưa ra là $ 249 mỗi máy tính. PictureTel đã tham chiếu một thỏa thuận với Microsoft trong thông cáo báo chí của mình và một bản ghi nhớ ngày 26 tháng 5 năm 1995 từ Bill Gates cho nhân viên điều hành của Microsoft và báo cáo trực tiếp cho biết "Ứng dụng chia sẻ màn hình PictureTel của chúng tôi cho phép chia sẻ Window sẽ hoạt động dễ dàng trên Internet."

Vào tháng 5 năm 1996, Microsoft đã công bố NetMeeting là một thành phần được bao gồm trong Internet Explorer 3.0. Vào thời điểm đó, Microsoft đã gọi NetMeeting là "ứng dụng khách truyền thông thời gian thực đầu tiên của Internet bao gồm hỗ trợ cho các tiêu chuẩn hội nghị quốc tế và cung cấp khả năng chia sẻ ứng dụng và hội thảo dữ liệu nhiều người dùng thực sự."

Năm 1996, PlaceWare được thành lập như một spinoff từ Xerox PARC. Vào tháng 11 năm đó, PlaceWare Auditorium đã được mô tả trong một cuộc nói chuyện công khai tại Đại học Stanford khi cho phép "một hoặc nhiều người trình bày tương tác, trực tuyến, đa phương tiện qua Web cho hàng trăm hoặc hàng ngàn người tham dự đồng thời; bài thuyết trình có thể bao gồm các slide ( được tạo trong PowerPoint hoặc bất kỳ trình chỉnh sửa hình ảnh GIF nào), chú thích trực tiếp trên các hình ảnh slide, các cuộc thăm dò ý kiến thời gian thực của khán giả, âm thanh trực tiếp từ người thuyết trình và những người đặt câu hỏi, văn bản và cuộc hội thoại âm thanh trong "hàng" của khán phòng và khác tính năng. " Thính phòng PlaceWare được chính thức công bố vào tháng 3 năm 1997 với mức giá $ 150 mỗi người dùng đồng thời.

Được công bố vào năm 1996 bởi InSoft Inc., CoolTalk là một công cụ phần mềm đa phương tiện cho phép người dùng PC xem dữ liệu được hiển thị trên bảng trắng được chia sẻ, trao đổi tin nhắn thời gian thực qua công cụ trò chuyện hoặc nói chuyện với nhau thông qua kết nối thoại TCP / IP. Sản phẩm hoạt động với bảng âm thanh tương thích với Microsoft Sound System và có sẵn ở phiên bản 14,4 kbit / s hoặc 28,8 kbit / s. CoolTalk sau đó đã được đóng gói với các trình duyệt Web phổ biến thời bấy giờ. CoolTalk 14.4 và 28.8 được bán với giá lần lượt là 49,95 và 69,95 USD vào năm 1996.

Vào tháng 2 năm 1998, Starlight Networks đã phát hành StarLive! (dấu chấm than là một phần của tên sản phẩm). Thông cáo báo chí cho biết "khách hàng có thể truy cập các giao diện trình duyệt Web quen thuộc để xem các bài thuyết trình trực tiếp và được ghi lại trước công ty, cùng với các slide được đồng bộ hóa. Người dùng cuối có thể giao tiếp trực tiếp với người trình bày bằng công nghệ trò chuyện thời gian thực và các công cụ cộng tác dựa trên Web khác."

Vào tháng 6 năm 1998, Trung tâm Hội nghị PlaceWare 2.0 đã được phát hành, cho phép tới 1000 người tham dự trực tiếp trong một phiên họp.

Vào tháng 2 năm 1999, ActiveTouch đã công bố Trung tâm Hội nghị WebEx và trang web webex.com. Vào tháng 7 năm 1999, Trung tâm Hội nghị WebEx đã chính thức được phát hành với khả năng họp 1000 người được thể hiện. Vào tháng 9 cùng năm, ActiveTouch đã đổi tên công ty thành WebEx.

Vào tháng 4 năm 1999, Vstream đã giới thiệu sản phẩm Netcall cho hội nghị web là "tiện ích phần mềm Internet dựa trên phí cho phép bạn gửi các bài thuyết trình kinh doanh và thông tin đồ họa khác qua e-mail đến máy chủ Vstream. Vux chuyển đổi nội dung, một lần nữa sử dụng công nghệ phát trực tuyến, và làm cho bài thuyết trình có sẵn để xem tới 1.200 người cùng một lúc. " Vstream đã đổi tên công ty thành Evoke Communications vào năm 2000, với một sự thay đổi tiếp theo thành Raindance Communications vào năm 2002. Vào tháng 2 năm 2006, Raindance đã được mua lại bởi Bộ phận InterCall của West Corporation.

### 3.3.3. Các đặc điểm cơ bản của web conference:

* VoIP: giao tiếp âm thanh thời gian thực, thông qua
* Ghi âm cuộc họp: ghi lại nội dug của cuộc họp và lưu lại trong database
* Khung chat: giúp mọi người có thể đóng góp ý kiến, trò chuyện với nhau
* Chia sẻ màn hình: giúp cho việc trình bày được dễ dàng, tăng tính tương tác giữa người tham gia
* Khu vực video:nơi hiển thị hình ảnh từ webcam lên trang web theo thời gian thực
* Bảng trắng:giúp người dùng trình bày một cách đơn giản ý tưởng của mình trong buổi họp, với những công cụ như trong Paint.

### 3.3.4. Ưu điểm và nhược điểm:

**Ưu điểm:**

* + Giảm chi phí đi lại, nâng cao hiệu quả hoạt động
  + Nâng cao tính cơ động cho các cuộc họp. Người muốn tham gia các cuộc họp  có thể ngồi ở bất cứ đâu
  + Thêm một điểm họp mới đơn giản thuận tiện
  + Cho phép nhiều cuộc họp diễn ra cùng một thời điểm
  + Cho phép phối hợp nhanh và hiệu quả đối với các văn phòng được phân bổ theo vùng địa lý
  + Tăng cường hiệu quả đối với dự án
  + Chi phí thấp nhưng vẫn đạt hiệu quả cao
* **Nhược điểm:**
  + Chi phí đầu tư trang thiết bị hội nghị trực tuyến khá cao.
  + Các cuộc họp bị giới hạn số lượng người tham gia.
  + Cuộc họp bị phụ thuộc chính vào đường truyền, hình ảnh, âm thanh đôi khi dễ bị gián đoạn trong các cuộc họp
  + Có thể bị hack.

## 3.4. Websocket:

### 3.4.1. Giới thiệu:

Ban đầu , internet không được xây dựng để dành cho những app động và phức tạp. Nó được hình thành như là một tập hợp của các trang HTML, kết nối với nhau để cấu thành nên khái niệm về "Web" chứa thông tin. Phần lớn mọi thứ được xây dựng xung quanh mô hình request/response nổi tiếng của HTTP. Một client tải trang và không có gì xảy ra cho đến khi user bắt đầu click và di chuyển đến trang tiếp theo.

Khoảng năm 2005, AJAX được giới thiệu và rất nhiều người đã khám phá khả năng tạo kết nối giữa client và server theo 2 chiều (bidirectional). Và vẫn như thế, tất cả giao tiếp HTTP được chỉ đạo bởi client yêu cầu user tương tác hoặc thực hiện theo chu kỳ để lấy dữ liệu mới từ server.

Một Websocket là một giao thức truyền thông được sử dụng bởi các máy tính. Nó cung cấp các kênh truyền thông song công đầy đủ qua giao thức TCP IP.Nó được coi là một kênh hai chiều giữa trình duyệt và server. WebSocket rất ổn định à, vì vậy client có thể mở một kết nối và duy trì kết nối đó với server trong suốt phiên làm việc, không giống như HTTP chỉ là một yêu cầu duy nhất,đáp ứng, sau đó đóng kết nối. Các thuộc tính này của WebSocket làm cho nó lý tưởng cho giao tiếp thời gian thực giữa máy khách và máy chủ.

### 3.4.2. Lịch sử:

WebSocket lần đầu tiên được tham chiếu là TCPConnection trong đặc tả HTML5, với tư cách là trình giữ chỗ cho API ổ cắm dựa trên TCP. Vào tháng 6 năm 2008, một loạt các cuộc thảo luận đã được dẫn dắt bởi Michael Carter dẫn đến phiên bản đầu tiên của giao thức được gọi là WebSocket.

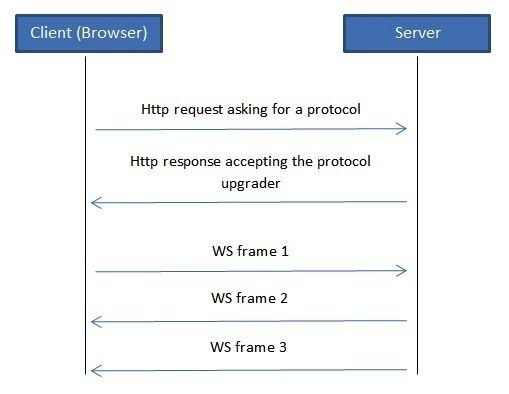
Cái tên "WebSocket" được đặt ra bởi Ian Hickson và Michael Carter ngay sau đó thông qua sự hợp tác trên phòng chat #whatwg IRC, và sau đó được tác giả đưa vào đặc tả HTML5 của Ian Hickson và được công bố trên blog cometd Daily của Michael Carter . Vào tháng 12 năm 2009, Google Chrome 4 là trình duyệt đầu tiên cung cấp hỗ trợ đầy đủ cho tiêu chuẩn, với WebSocket được bật theo mặc định.

Việc phát triển giao thức WebSocket sau đó đã được chuyển từ nhóm W3C và WHATWG sang IETF vào tháng 2 năm 2010 và là tác giả cho hai phiên bản dưới thời Ian Hickson.

Sau khi giao thức được vận chuyển và bật theo mặc định trong nhiều trình duyệt, RFC đã được hoàn thành theo Ian Fette vào tháng 12 năm 2011.

### 3.4.3. Cách hoạt động:

Giao thức có hai phần: Bắt tay và truyền dữ liệu.Ban đầu client sẽ gửi yêu cầu khởi tạo kết nối websocket đến server, server kiểm tra và gửi trả kết quả chấp nhận kết nối, sau đó kết nối được tạo và quá trình gửi dữ liệu có thể được thực hiện, dữ liệu chính là các Ws frame, xếp lớp trên TCP



Hình 3.12. Giao thức truyền tin của websocket

Client thiết lập một kết nối WebSocket thông qua một tiến trình được gọi là WebSocket handshake (bắt tay WebSocket). Tiến trình này bắt đầu với client gửi một request HTTP thông thường đến server. Nó kèm theo header Upgrade để thông báo cho server rằng client muốn tạo một kết nối WebSocket.

Quá trình mở kết nối WebSocket ở phía client:

var socket = new WebSocket('ws://websocket.example.com');

WebSocket URL sử dụng ws scheme. Chúng ta còn có cả wss cho những kết nối WebSocket bảo mật hơn, tương tự như HTTPS.Scheme này bắt đầu một tiến trình mở kết nối WebSocket đến websocket.example.com.

Ví dụ Header của request khởi tạo:

GET ws://websocket.example.com/ HTTP/1.1Origin: http://example.comConnection: UpgradeHost: websocket.example.comUpgrade: websocket

Nếu server hỗ trợ giao thức WebSocket, nó sẽ đồng ý để nâng cấp và giao tiếp thông qua header Upgrade trong response.

Sau khi thành lập kết nối, server trả về:

HTTP/1.1 101 Switching ProtocolsDate: Wed, 25 Oct 2017 10:07:34 GMTConnection: UpgradeUpgrade: WebSocket

Khi kết nối đã được thiết lập, sự kiện open sẽ được bắn ra cho instance WebSocket ở phía client:

var socket = new WebSocket('ws://websocket.example.com');

socket.onopen = function(event) {

console.log('WebSocket is connected.');

};

Giờ thì quá trình "bắt tay" đã hoàn tất, kết nối khởi tạo HTTP được thay thế bằng WebSocket và sử dụng cùng loại nền tảng kết nối TCP/IP. Tại thời điểm này, cả 2 bên đều có thể gửi dữ liệu.

Với WebSocket, bạn có thể truyền bao nhiêu thông tin tùy thích mà không cần phải gánh chịu những chi phí không đáng có liên quan đến request HTTP truyền thống. Dữ liệu được truyền đi thông qua WebSocket dưới dạng tin nhắn (message), mỗi tin nhắn bao gồm một hoặc nhiều frame chứa dữ liệu người dùng gửi đi (gọi là kiện hàng - payload). Để đảm bảo message có thể tái cấu trúc một cách chính xác khi nó đến với client, mỗi frame được gán cứng từ 4-12 byte thông tin về payload. Sử dụng hệ thống thông tin dựa trên frame như thế này giúp giảm tải khối lượng dữ liệu dư thừa (non-payload data) phải truyền đi, có thể làm cho độ trễ giảm đi đáng kể.

**Lưu ý**: Đặc biệt chú ý là client chỉ được thông báo về message mới một khi tất cả frame đều được nhận và payload message gốc được tái cấu trúc đầy đủ.URL có cấu trúc ngữ pháp cụ thể về scheme. WebSocket URL đặc biệt vì nó không hỗ trợ nhóm ký tự anchor.

* **WebSocket URLs**

Có những luật chung được áp dụng cho cả style của WebSocket URL và HTTP URL. ws không được mã hóa, nó có cổng mặc định là 80 trong khi đó wss yêu cầu mã hóa TLS và dùng cổng 443 mặc định.

* **Framing protocol (Giao thức framing):**

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-------+-+-------------+-------------------------------+

|F|R|R|R| opcode|M| Payload len | Extended payload length |

|I|S|S|S| (4) |A| (7) | (16/64) |

|N|V|V|V| |S| | (if payload len==126/127) |

| |1|2|3| |K| | |

+-+-+-+-+-------+-+-------------+ - - - - - - - - - - - - - - - +

| Extended payload length continued, if payload len == 127 |

+ - - - - - - - - - - - - - - - +-------------------------------+

| |Masking-key, if MASK set to 1 |

+-------------------------------+-------------------------------+

| Masking-key (continued) | Payload Data |

+-------------------------------- - - - - - - - - - - - - - - - +

: Payload Data continued ... :

+ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - +

| Payload Data continued ... |

+---------------------------------------------------------------+

Theo như RFC định nghĩ về WebSocket thì nó chỉ có duy nhất một header trước mỗi gói tin:

* + fin (1 bit): biểu thị nếu frame đó là frame cuối cùng tạo nên message. Hầu hết message đều có thể chứa trong một frame duy nhất và bit này luôn có giá trị. Thực nghiệm cho thấy Firefox tạo 1 frame thứ 2 sau mỗi 32K.
  + rsv1, rsv2, rsv3 (Mỗi thứ 1 bit): phải bằng 0 trừ khi có một [extension](https://tools.ietf.org/html/rfc6066) (đại khái như tiện ích mở rộng) được trao đổi và nó định nghĩa ý nghĩa cho giá trị khác 0. Nếu nhận về một giá trị khác 0 và không có extension nào xác định ý nghĩa của giá trị khác 0 đó được trao đổi thì người nhận cuối sẽ bị mất kết nối.
    - opcode (4 bit): nó cho biết những gì frame đại diện. Có những giá trị sau đây được dùng:.
    - 0x00: frame này tiếp tục payload từ frame trước đó.
    - 0x01: frame này chứa dữ liệu chữ (text).
    - 0x02: frame này chứa dữ liệu nhị phân (binary).
    - 0x08: frame này hủy kết nối.
    - 0x09: frame này là ping.
    - 0x0a: frame này là pong.

( Có nhiều giá trị không sử dụng, chúng được dự trữ cho tương lai khi cần)

* mask (1 bit): biểu thị nếu kết nối đã được che đậy (mask). Rõ ràng là mỗi message từ client đến server phải được che đậy và về mặt kỹ thuật thì nó sẽ hủy kết nối nếu không được che đậy.
* payload\_len (7 bit): độ dài của payload. WebSocket frame có những độ dài theo các khoảng tính theo byte như sau:
  + 0-125 biểu thị độ dài của payload
  + Nếu là 126 byte thì 2 byte tiếp theo biểu thị độ dài payload dưới dạng số nguyên 16bit không dấu
  + Nếu là 127 byte thì 8 byte tiếp theo biểu thị độ dài payload dưới dạng số nguyên 64bit không dấu
  + Vậy là độ dài payload sẽ vào khoảng ~7bit, 16bit hoặc 64bit.
  + masking-key (32 bit): tất cả frame được gửi từ client đến server sẽ được che đậy (mask) bằng một giá trị 32bit đính kèm trong frame.
  + payload: Dữ liệu thật sự cần được che đậy. Độ dài của nó chính là payload\_length
* **Dữ liệu trên Frame:**

Dữ liệu có thể phân mảnh thành nhiều frame. Frame đầu tiên chuyển giao dữ liệu có một opcode biểu thị kiểu dữ liệu đang được truyền. Điều này cần thiết bởi vì Javascript hầu như không có hỗ trợ cho kiểu dữ liệu nhị phân (binary) khi nó được xây dựng. 0x01 biểu thị kiểu encode văn bản UTF-8, 0x02 là dữ liệu nhị phân. Đa số mọi người sẽ chuyển giao JSON trong trường hợp người dùng muốn chọn opcode văn bản. Khi phát tín hiệu (emit) dữ liệu nhị phân nó sẽ được thể hiện trên trình duyệt dưới dạng cụ thể là [Blob](https://developer.mozilla.org/en-US/docs/Web/API/Blob).

API để gửi dữ liệu thông qua WebSocket :

var socket = new WebSocket('ws://websocket.example.com');

socket.onopen = function(event) {

socket.send('Some message'); *// Gửi dữ liệu đến server.*

};

Khi WebSocket nhận dữ liệu (ở phía client), một sự kiện message được bắn ra. Sự kiện này bao gồm một thuộc tính gọi là data có thể dùng để truy cập nội dung của message.

*// Xử lý message gửi đi từ server.*

socket.onmessage = function(event) {

var message = event.data;

console.log(message);

};

* **Sự phân mảnh:**

Dữ liệu payload có thể được chia thành nhiều frame riêng. Nơi nhận có nhiệm vụ lưu đệm chúng cho đến khi bit fin được set. Vì vậy ,có thể chuyển 1 chuỗi "Hello World" trong 11 gói tin của 6 (độ dài header) + 1 byte cho mỗi gói. Sự phân mảnh không được chấp nhận cho gói tin điều khiển (control packages). Tuy nhiên, đặc điểm kỹ thuật yêu cầu có thể xử lý những frame điều khiển xen kẽ nhau. Đó là trường hợp gói tin TCP nhận được có thứ tự lộn xộn.

Logic để kết nối frame được mô tả sơ lược như sau:

* + Nhận frame đầu tiên
  + Ghi nhớ opcode
  + Nối các frame payload lại với nhau cho đến khi bit fin được set.
  + Giả định rằng opcode cho mỗi gói tin là 0

Mục đích chính của sự phân mảnh là cho phép gửi message khi không biết rõ kích thước ban đầu của message. Với sự phân mảnh, server có thể chọn một kích thước buffer (bộ đệm) phù hợp và khi buffer đầy thì ghi mảnh (fragment) đó vào network. Trường hợp sử dụng phụ của sự phân mảnh là truyền tin đa luồng (multiplexing), vốn dĩ không cần một message lớn trên một kênh logic để tiếp nhận toàn bộ kênh đầu ra, vì thế multiplexing cần phải giải phóng để cắt message ra thành nhiều mảnh để có thể chia sẻ đến kênh đầu ra tốt hơn.

* **Heartbeating (nhịp tim) là gì ?**

Tại một thời điểm sau khi "bắt tay" (handshake), cả client và server có thể lựa chọn để gửi đi một ping đến phía kia. Khi ping được nhận, người nhận phải gửi ngược lại một pong ngay khi có thể. Đó gọi là heartbeat (nhịp tim đập). người dùng có thể dùng nó để đảm bảo client vẫn đang được kết nối.

Một ping hay pong chỉ là frame bình thường, không phải frame điều khiển. Ping có opcode là 0x9 và opcode của pong là 0xA. Khi nhận được ping, gửi ngược lại pong với chính xác cùng dữ liệu Payload như ping (với ping và pong thì độ dài payload tối đa là 125). Người dùng cũng có thể nhận được pong mà chưa từng gửi ping.

Heartbeat có thể rất có ích. Có nhiều dịch vụ (chẳng hạn như bộ cân bằng tải - load balancer) sẽ hủy những kết nối đứng yên (idle). Thêm nữa, bên nhận không thể biết nếu bên kia (bên gửi) đã bị kết thúc hay chưa. Chỉ có đến lần gửi thông tin tiếp theo ta mới nhận ra có gì đó không ổn.

* **Xử lý lỗi**

Người dùng có thể xử lý bao nhiêu lỗi xảy ra cũng được bằng cách listen đến sự kiện error.  
 Ví dụ:

var socket = new WebSocket('ws://websocket.example.com');

*// Xử lý lỗi xảy ra.*

socket.onerror = function(error) {

console.log('WebSocket Error: ' + error);

};

### 3.4.4. Ưu điểm:

* WebSockets giữ một kết nối duy nhất mở trong khi loại bỏ các vấn đề về độ trễ phát sinh với Long Polling.
* WebSockets thường không sử dụng XMLHTTPRequest, và do đó, các tiêu đề không được gửi mỗi lần người dùng cần để có thêm thông tin từ máy chủ. Điều này đã làm giảm tải dữ liệu đắt tiền được gửi đến máy chủ.

### 3.4.5. Nhược điểm:

* Các trình duyệt cũ hơn 2011 không thể hỗ trợ các kết nối WebSocket - nhưng điều này ngày càng trở nên không còn quan trọng.
* WebSockets không tự động khôi phục khi kết nối bị chấm dứt - đây là điều người dùng cần tự thực hiện và là một phần lý do tại sao có nhiều thư viện phía máy khách tồn tại

**3.5. iframe:**

3.5.1. Giới thiệu:

iFrame là viết tắt của Inline Frame. Nó là một yếu tố mạnh mẽ trong thiết kế web. Bạn có thể đã thấy vô số video YouTube được nhúng trên các trang web khác ngoài YouTube. IFrame có thể chèn tất cả các loại phương tiện từ video, hình ảnh, âm thanh cho đến cả chèn 1 trang web khác.

IFrame là một khung trong một khung. Nó là một thành phần của phần tử HTML cho phép bạn nhúng tài liệu, video và phương tiện tương tác vào trong một trang. Bằng cách này, bạn có thể hiển thị một trang web phụ trên trang chính của mình.

Phần tử iFrame cho phép bạn hiển thị một phần nội dung từ các nguồn khác. Nó có thể tích hợp nội dung ở bất kỳ đâu trong trang của bạn mà không cần phải đưa chúng vào cấu trúc bố cục web của bạn, giống như một phần tử truyền thống.

3.5.2. Cách sử dụng:

* Cú pháp:

<iframe src="url" title="description">

* Thuộc tính:

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Giá trị** | **Mô tả** |
| allow |  | Chỉ định tính năng cho iframe |
| allowfullscreen | true false | Đặt thành true nếu iframe có thể kích hoạt chế độ toàn màn hình bằng cách gọi phương thức requestFullscreen () |
| allowpaymentrequest | true false | Đặt thành true nếu một iframe có nhiều nguồn được phép gọi API Payment Req |
| [height](https://www.w3schools.com/tags/att_iframe_height.asp) | *pixels* | Điều chỉnh chiều cao cho khung iframe. Chiều cao mặc định là 150 pixel |
| [name](https://www.w3schools.com/tags/att_iframe_name.asp) | *text* | Đặt tên cho khung iframe |
| [referrerpolicy](https://www.w3schools.com/tags/att_iframe_referrerpolicy.asp) | no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin strict-origin-when-cross-origin unsafe-url | Chỉ định số lượng / thông tin liên kết giới thiệu nào sẽ được gửi khi xử lý các thuộc tính iframe |
| [sandbox](https://www.w3schools.com/tags/att_iframe_sandbox.asp) | allow-forms allow-pointer-lock allow-popups allow-same-origin allow-scripts allow-top-navigation | Bật thêm một nhóm hạn chế cho nội dung trong iframe |
| [src](https://www.w3schools.com/tags/att_iframe_src.asp) | *URL* | Chỉ định địa chỉ của tài liệu để nhúng vào iframe |
| [srcdoc](https://www.w3schools.com/tags/att_iframe_srcdoc.asp) | *HTML\_code* | Chỉ định nội dung HTML của trang để hiển thị trong iframe |
| [width](https://www.w3schools.com/tags/att_iframe_width.asp) | *pixels* | Chỉ định chiều rộng của <iframe>. Chiều rộng mặc định là 300 pixel |

3.5.3. Ưu điểm và nhược điểm:

* Ưu điểm:Việc bạn nhúng nhiều thành phần của các trang web khác nhau vào website của mình để tạo ra các **multiple view** (các cửa sổ độc lập) được **iFrame** đảm nhiệm rất tốt. Và nếu website bạn vận hành trơn tru với phương pháp này, nó sẽ đem lại lợi ích cực kì lớn cho người đọc trong quá trình truy cập. Bởi họ chỉ cần tìm đến một địa chỉ website duy nhất lại có thể tổng hợp [kiến thức](https://wiki.matbao.net/kien-thuc/) trực quan sinh động từ nhiều nguồn khác nhau.
* Nhược điểm:Bảo mật thấp, dễ dính mã độc nếu site nguồn không an toàn.Việc tấn công thông qua chèn mã độc trong iFrame đã tăng lên, đặc biệt là trên những website danh tiếng. Cuộc tấn công hướng người dùng tới một site độc hại và khiến họ cài đặt virus vào máy tính để đánh cắp thông tin nhạy cảm. Đó cũng chính là lý do vì sao iFrame không được trở thành một phần chính trong bố cục của website.

3.6. Apache:

3.6.1.Giới thiệu:

Apache gọi tắt của Apache HTTP Server là một chương trình máy chủ giao tiếp bằng giao thức HTTP và hoạt động trên hầu hết các hệ điều hành**.** Apachelàphần mềm web servermiễn phímã nguồn mở.Mục đích của Apachelà giúp chủ trang website đưa nội dung lên web. Do đó Apache được xếp vào dạng phần mềm “web server”. Đây là một trong số những webserver lâu đời và đáng tin cậy nhất với phiên bản đầu tiên đã được ra mắt từ hơn 20 năm trước.

3.6.2. Cách hoạt động:

Apache là một phần mềm chạy trên server,thiết lập kết nối giữa server và các trình duyệt của người dùng, chẳng hạn như Firefox, Google Chrome, Safari,… rồi truyền, trao đổi file cấu trúc 2 chiều dạng client**-**server. Như vậy xét về cách thức hoạt động, Apache là một phần mềm đa nền tảng, hoạt động tốt với cả server Unix và Windows.

Khi khách truy cập một trang web, trình duyệt sẽ gửi yêu cầu tải trang web đó lên server. Sau đó, Apache sẽ trả kết quả với tất cả đầy đủ các file cấu thành nên trang mà khách hàng muốn truy cập bao gồm cả nội dung, hình ảnh, video,… Server và client sẽ giao tiếp với nhau qua giao thức HTTP, khi đó Apache đảm nhận trách nhiệm đảm bảo tiến trình này diễn ra mượt mà và bảo mật một cách tối ưu nhất.

Với cách hoạt động như vậy, Apache là một nền tảng module có độ tùy biến rất cao. Nó cho phép quản trị server có cả quyền tắt hoặc thêm chức năng. Đặc biệt, Apache có module cho bảo mật caching, URL rewriting, chứng thực mật khẩu,… Giúp bạn bảo vệ trang web của mình một cách an toàn nhất.

3.6.3. Tomcat:

Tomcat là một web server cũng được phát triển bởi Apache Software Foundation, vì vậy tên chính thức của nó là [Apache Tomcat](https://tomcat.apache.org/). Nó cũng là một server HTTP, tuy nhiên, nó hỗ trợ mạnh cho ứng dụng Java thay vì website tĩnh. Tomcat có thể chạy nhiều bản Java chuyên biệt như Java Servlet, JavaServer Pages (JSP), Java EL, và WebSocket.

Tomcat được tạo đặc biệt riêng cho Java apps, mặc dù Apache là vẫn là một server HTTP. Có thể sử dụng Apache với nhiều ngôn ngữ lập trình khác (PHP, Python, Perl, vâng vâng.) với sự giúp đỡ của module Apache phù hợp (mod\_php, mod\_python, mod\_perl, etc.).

* Tomcat được tạo đặc biệt riêng cho Java apps, mặc dù Apache là vẫn là một server HTTP. Bạn có thể sử dụng Apache với nhiều ngôn ngữ lập trình khác (PHP, Python, Perl, vâng vâng.) với sự giúp đỡ của module Apache phù hợp (mod\_php, mod\_python, mod\_perl, etc.).
* Mặc dù bạn có thể sử dụng Tomcat server để phục vụ trang web tĩnh, nhưng nó không hiệu quả như là khi sử dụng Apache. Ví dụ, Tomcat sẽ tải máy ảo Java lên trước và những thư viện Java liên quan khác, mà website thông thường thì không cần thiết.
* Tomcat cũng khó cấu hình hơn các web server khác. Ví dụ, để chạy [WordPress](https://www.hostinger.vn/huong-dan/cach-tao-blog), hãy dùng các server dành cho HTTP như là Apache hoặc NGINX.

3.6.4.Ưu điểm và nhược điểm:

* Ưu điểm:
  + Là phần mềm mã nguồn mở cho phép sử dụng miễn phí cho mọi mục đich
  + Có độ ổn định cao,được cập nhật thường xuyên,nâng cấp và vá lỗi bảo mật lien tục
  + Đa nền tảng, hoạt động được trên nhiều hệ điều hành: Windows,Unix,Linux,Mac Os,….
* Nhược điểm:
  + Không đảm bảo ổn định nếu website có lượng truy cập lớn
  + Có các điểm yếu bảo mật

3.8. Spring Boot:

3.8.1.Giới thiệu:

Spring Boot là một dự án phát triển bởi JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

Để phát triển một ứng dụng web cơ bản sử dụng Spring framework bạn sẽ cần ít nhất 5 công đoạn sau;

* Tạo một project sử dụng Maven với các dependency cần thiết của Spring MVC và Servlet API.
* Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC.
* Một tập tin cấu hình của Spring MVC.
* Một class Controller trả về một trang web khi có request đến.
* Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy.

Trong các công đoạn này, chỉ có công đoạn tạo một class Controller thì có thể khác cho các ứng dụng khác nhau vì mỗi ứng dụng có một yêu cầu khác nhau. Còn các công đoạn khác thì như nhau.

Giờ đây với Spring Boot, chúng ta có thể tạo dự án Spring một cách nhanh chóng và cấu hình cũng đơn giản dùng Sublime Text để phát triển luôn khỏi cần cài đặt eclipse hay netbean nặng bỏ bà.  
Dưới đây là một số tính năng nổi bật của Spring Boot:

* Tạo các ứng dụng Spring độc lập
* Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần phải deploy ra file WAR)
* Các starter dependency giúp việc cấu hình Maven đơn giản hơn
* Tự động cấu hình Spring khi cần thiết
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …

# CHƯƠNG 4: CÁC GIẢI PHÁP ĐIỀU KHIỂN CAMERA TỪ XA

## 4.1.Các giải pháp điều khiển camera từ xa:

Ngày này, với sự phát triển của công nghệ điều khiển đã mang lại sự thuận tiện cho con người trong nhiều lĩnh vực:

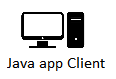
* Giáo dục – đào tạo từ xa: dạy và học trực tuyến từ xa
* Y tế - chăm sóc từ xa: hội chẩn bệnh, tham gia phẩu thuật gián tiếp giữa các bệnh viện và chăm sóc bệnh nhân từ xa.
* Hội họp, hội nghị, giao ban, trao đổi công việc giữa các đơn vị có vị trí địa lý cách xa nhau.
* An ninh - giám sát hoạt động của nhân viên, bảo vệ tài sản doanh nghiệp

Có thể nói điều khiển camera từ xa là lựa chọn tối ưu với bất kỳ doanh nghiệp hay tổ chức nào khi khoảng cách giữa các điểm với nhau là khá xa, không thuận lợi cho việc đi lại để trực tiếp quan sát, trao đổi công việc. Hiện tại có 2 giải pháp chính để điều khiển từ xa:

* Giải pháp dùng phần cứng: sử dụng dây nối
* Giải pháp phần mềm: tạo một chương trình truyền và nhận tín hiệu từ camera qua một server trên Internet

## Giới thiệu phác thảo phương hướng:

****

****

HTTP

HTTP

RS422

****

****

Hình 4.1.Mô hình phác thảo hệ thống thực hiện

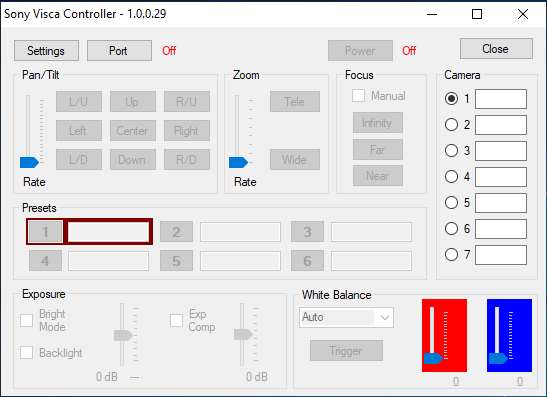
* + 1. Điều khiển camera:

Sử dụng các sợi cáp có sẵn chuyển đổi từ USB sang chuẩn RS422 hoặc các bộ chuyển đổi trên thị trường, ta có thể kết nối Camera EVI D80 với máy tính một cách trực tiếp.



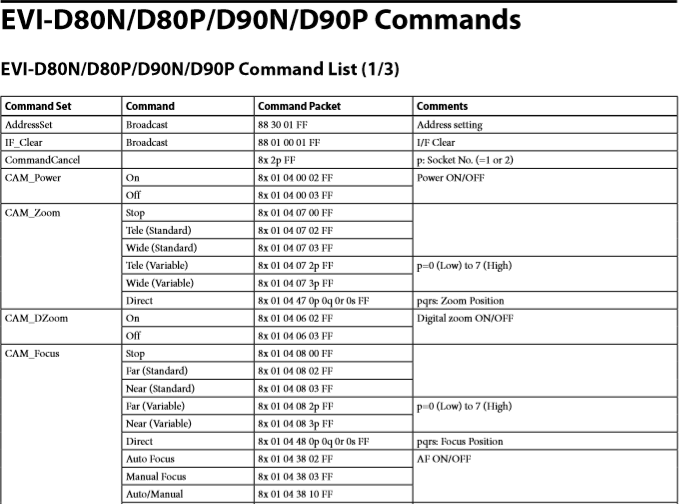
Hình 4.2. Cáp chuyển đổi USB-RS422

Thông qua C#, sẽ thiết kế Layout mô phỏng lại Remote Control của bộ điều khiển từ xa. Bao gồm các nút bấm điều hướng và các pím chức năng của Camera EVI D80.



Hình 4.3 Bảng điều khiển thiết kế tham khảo

Chèn các lệnh và tập lệnh của Camera vào các phím chức năng trong Layout Control. Bảng lệnh lấy từ hướng dẫn sử dụng có sẵn của Camera và được nhà sản xuất cung cấp. Từ đây đã có thể điều khiển Camera thông qua máy tính.



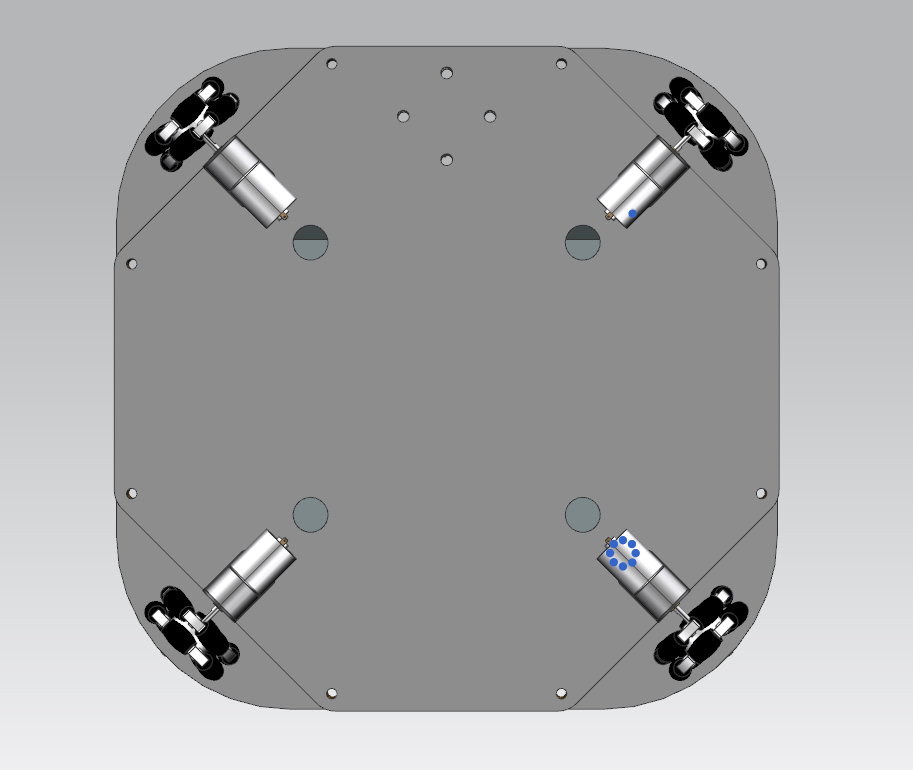
Hình 4.4 Bảng tập lệnh của Camera tham khảo.

* + 1. Điều khiển xe:
       1. Thiết kế:

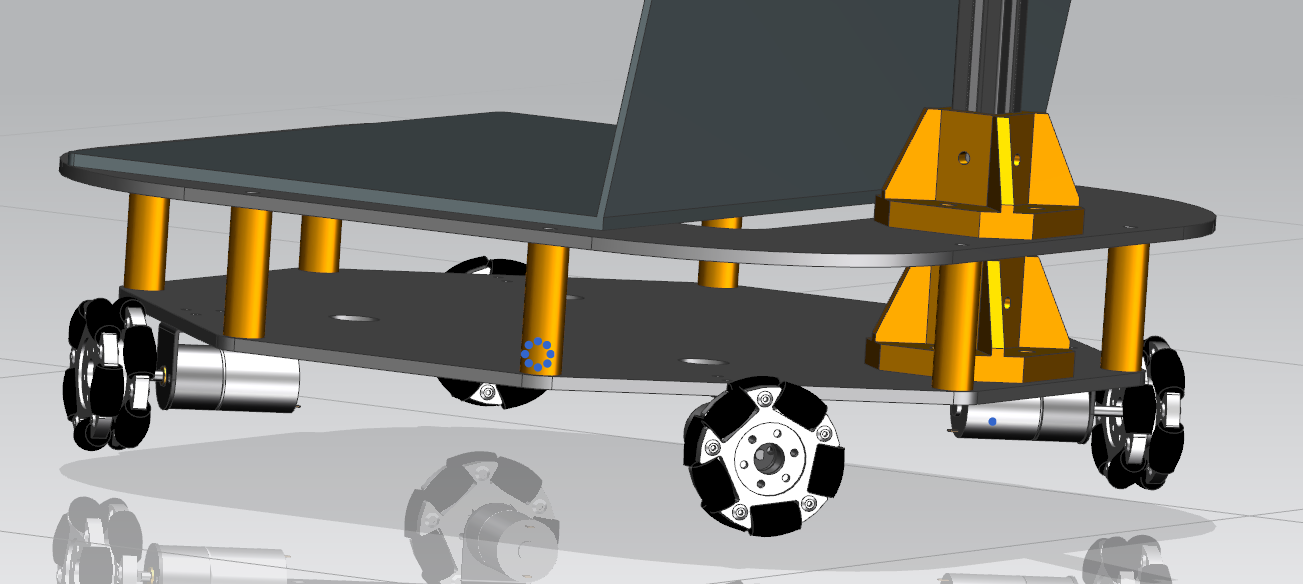
Yêu cầu:

* Xe chở được camera và 1 laptop với tổng trọng lượng hơn 7kg
* Xe có khả năng điều khiển từ xa

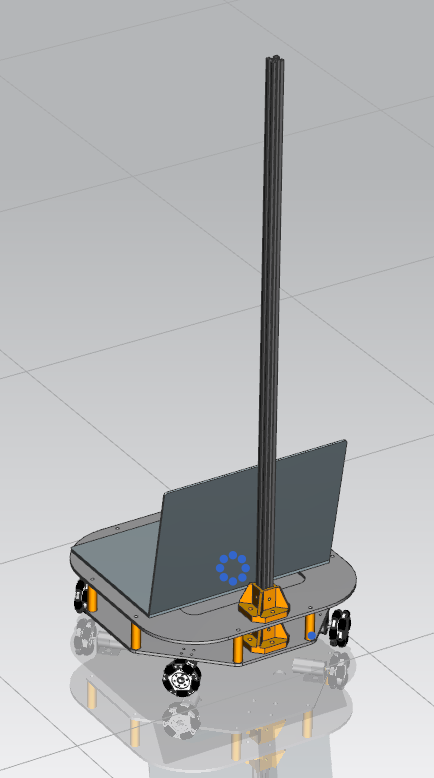
Bản vẽ:



Hình 4.5. Mặt dưới xe



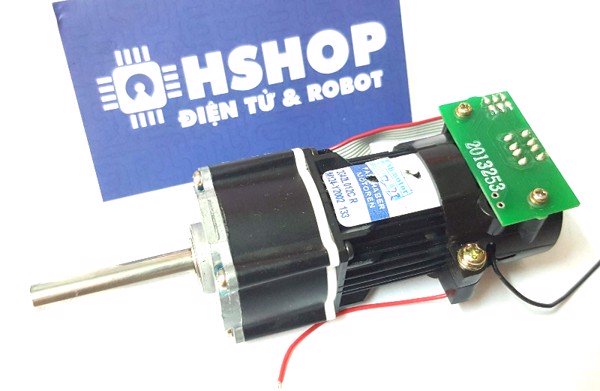
Hình 4.6. Xe điều khiển



Hình 4.7. Xe điều khiển

* + - 1. Động cơ:

Sử dụng động cơ servo giảm tốc falhauber:



Hình 4.5. Động cơ falhauber

Sản phẩm này là loại động cơ mạnh có kèm hộp số với tỉ số cao và encoder. Động cơ có thể được ứng dụng để chế tạo các loại xe và robot cần các cơ cấu có moment quay cao và chính xác.

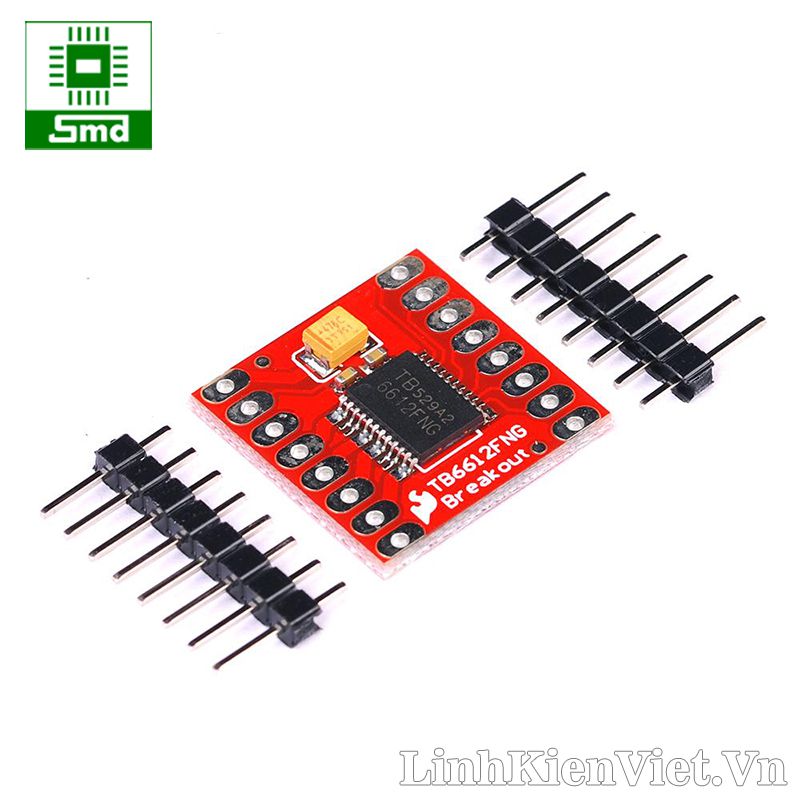
Thông tin cơ bản của động cơ:

|  |  |
| --- | --- |
| Điện áp hoạt động | 12V |
| Tốc độ motor | 8100 RPM |
| Tốc độ sau hộp giảm tốc | 120 RPM |
| Tỉ số giảm tốc | 64:1 |
| Dòng tải tối đa | 1400mA |
| Dòng không tải | 74mA |
| Momen xoắn | 1.72 Nm |
| Encoder | 2 kênh với độ phân giải 12 xung/vòng |
| Đường kính | 30mm |
| Chiều dài động cơ | 85mm |
| Đường kính trục | 6mm |
| Chiều dài trục | 35mm |

Để lấy được tín hiệu kênh 1, kênh 2 của encoder, tại mỗi chân kênh ra cần có thêm trở 10k  kéo lên nguồn 5V.Nguồn 5V cấp cho ENCODER phải mắc thêm điện trở có giá trị 120 ohm để hạn dòng cho 2 led phát hồng ngoại.

* + - 1. Bộ phận điều khiển:

Module điều khiển động cơ bước TB6612



Hình 4.8. Module điều khiẻn TB6612

Module điều khiển động cơ bước TB6612 1A dùng để điều khiển hai động cơ DC hoặc một động cơ bước với dòng điện liên tục ở 1.2A (đạt đỉnh 3.2A). Hai tín hiệu ngõ vào có thể sử dụng để điều khiển motor ở 4 chế độ khác nhau như quay thuận, quay ngược, hãm, stop. Hai motor có thể điều khiển riêng biệt, tốc độ mỗi motor được điều khiển qua tín hiệu PWM với tần số lên đến 100KHz.

Thông số kỹ thuật:

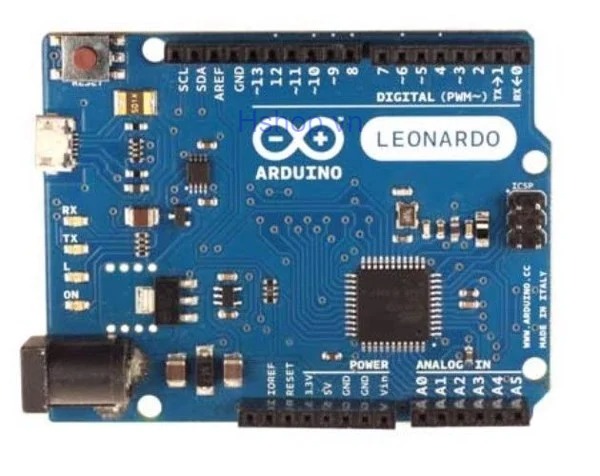
|  |  |
| --- | --- |
| Điện áp | 2.7~5.5 VDC |
| Tần số | 100KHz |
| Imax | 3.2A |
| I | 1.2A |

-Driver hỗ trợ 2 cầu H có thể dùng cho hai động cơ DC hoặc một động cơ bước.

 -4 chế độ điều khiển: quay thuận, quay ngược, hãm, stop.  
- Bảo vệ quá nhiệt và quá áp.  
- Tụ lọc ở cả 2 ngõ cấp nguồn.  
- Bảo vệ chống ngược nguồn cấp cho motor.

* + - 1. Bộ phận xử lý:

Arduino Leonardo



Hình 4.9.Arduino Leonardo

Thông số kỹ thuật

|  |  |
| --- | --- |
| Điện áp | 5VDC |
| Tần số hoạt động | 16MHz |
| Dòng tiêu thụ ở các chân | 40mA |
| Điện áp vào khuyên dung | 7-12VDC |
| Điện áp vào giới hạn | 20VDC |
| Số chân Digital I/O | 14 (7 chân PWM) |
| Số chân Analog | 12 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 40mA |
| Dòng ra tối đa (5V) | 500mA |
| Dòng ra tối đa (3.3V) | 50mA |
| Bộ nhớ flash | 32 KB (ATmega32u4) với 4KB dùng bởi bootloader |
| SRAM | 2.5KB (ATmega32u4) |
| EEPROM | 1KB |
| Kích thước | 68.6mm x 53.3mm |

Hướng thứ hai là dùng một công cụ Plugging ( web app) để xây dựng một Video Conferrence

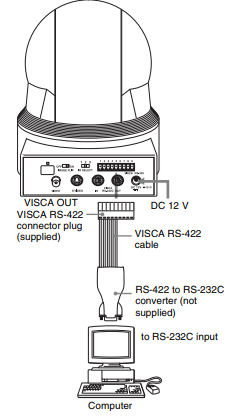
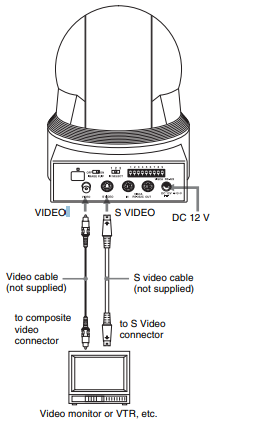
( Hội nghị trực tuyến). Từ đây người dùng có thể trực tiếp điều khiển Camera từ xa thông qua Web

* + - 1. Sơ đồ mạch:

## Giải pháp phần cứng:

Camera Evi-D80N sử dụng 2 đường truyền :

* Đường truyền lệnh: dùng để điều khiển camera xoay, zoom, thay đổi các chế độ làm việc thông qua sử dụng đầu nối RS242,RS422
* Đường truyền hình ảnh: sử dụng dây cáp s-video để truyền hình ảnh tới bộ thu

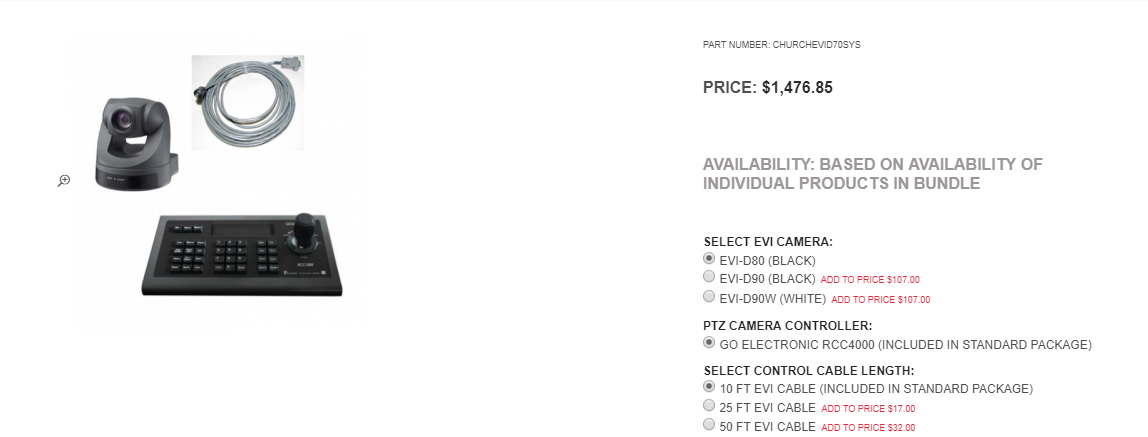
Hình 4.5. Dây nối truyền lệnh Hình 4.6. Dây nối truyền hình ảnh

Để điều khiển camera từ xa chúng ta chỉ cần tăng độ dài dây cáp nối camera đến nơi nhận dữ liệu. Phần lớn camera hiện nay được điều khiển thông qua dây. Để cho công việc dễ dàng, người ta có thể kết nối camera với tay cầm điều khiển.



Hình 4.7. Joystick điều khiển camera

Nhưng một vấn đề bức thiệt đặt ra là các dụng cụ đi kèm để điều khiển từ xa cũng có chi phì không hề dễ chịu, có thể lên tới vài nghìn đô

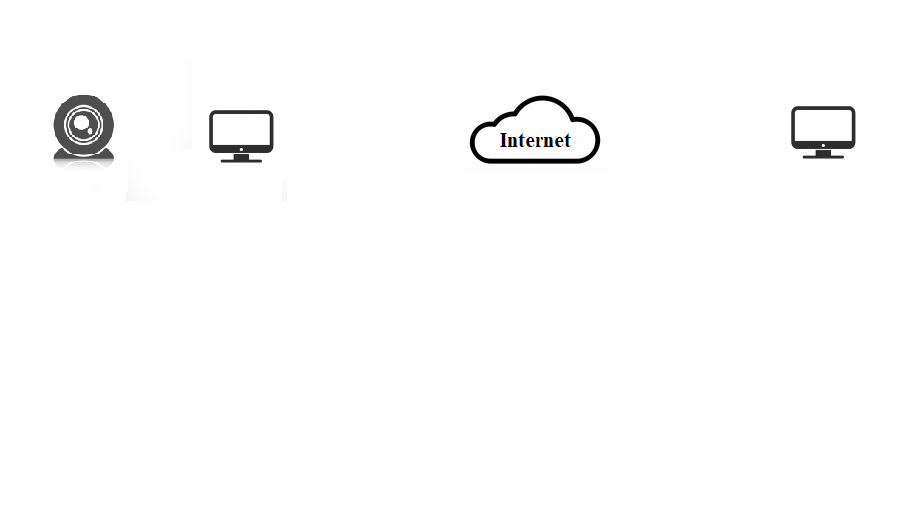


Hình 4.8. Bảng giá bộ điều khiển camera từ xa

## Giải pháp phần mềm:

Do camera đươc thiết kế để điều khiển và nhận lệnh thông qua dây nối. Vì vậy để áp dụng phương pháp này, người dùng cần có một máy tính để giao tiếp trực tiếp với camera.

Lợi dụng đặc tính hỗ trợ cổng kết nối RS232 và RS422, ta có thể xây dựng các chương trình điều khiển trên máy tính kết nối với camera. Từ đó, để điều khiển camera từ xa, ta chỉ cần sử dụng chương trình đã xây dựng.



Điều khiển

Hình ảnh

Hình 4.9.Sơ đồ điều khiển camera từ xa

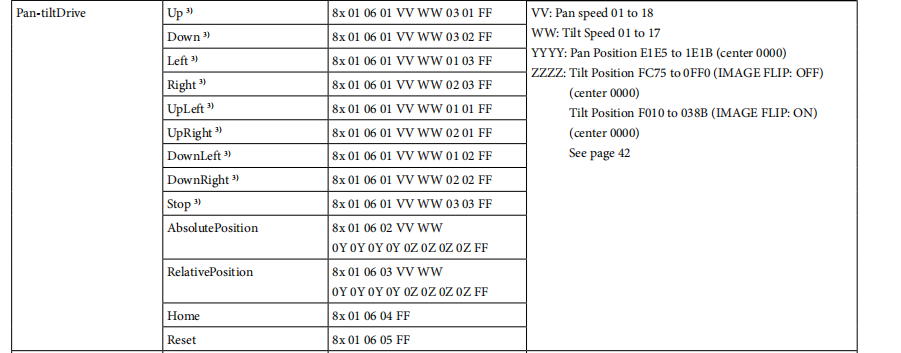
Chúng ta chia việc điều khiển camera thành 2 phần riêng biệt : Truyền lệnh và truyền hình ảnh.

### Truyền lệnh:

Chia phần truyền lệnh từ xa thành 2 bước : điều khiển ở gần bằng máy tính và điều khiển từ xa.

* + - 1. Điều khiển ở gần bằng máy tính:

Camera điều khiển thông qua giao thức truyền tin serial port và sử dụng các mã lệnh hex để hoạt động



Hình 4.10. Bảng lệnh điều khiển camera Evi-D80N

Ví dụ, để xoay trái, ta phải gửi mã lệnh 8x010601VVWW0103FF đến camera

Trong đó: - *x* tương đương với số thứ tự của camera trong hệ thống, được đặt khi khai báo ban đầu

- *VV* là tốc độ xoay trục ngang của camera

- *WW* là tốc độ xoay trục dọc của camera

Để truyền lệnh từ xa tới camera, trước hết chúng ta cần có một máy tính kết nối với camera. Kết hợp với việc sử dụng thư viện SerialPort của java, có thể xây dựng một chương trình giao tiếp với camera.Và từ đó ta có thể điều khiển camera quay trái , quay phải, phóng to, thu nhỏ, thay đổi chế độ làm việc.

* + - 1. Điều khiển từ xa :

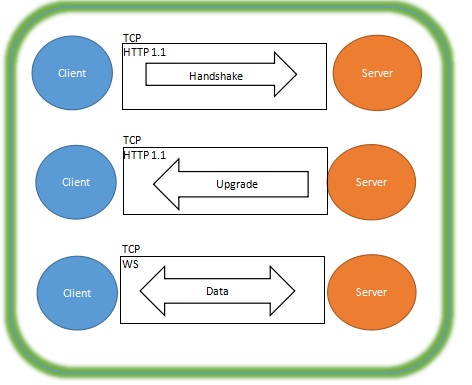
Có 3 phương pháp được đề ra:

* **Websocket:**

Websocket cho phép server và client gửi tin nhắn cho nhau bất kì lúc nào sau khi kết nối được thiết lập mà không cần gửi request từ phía này hay phía kia.Điều này trái ngược với HTTP, được liên kết theo nguyên tắc phản hồi - để có được dữ liệu người ta phải yêu cầu nó một cách rõ ràng.Nghĩa là, WebSockets cho phép kết nối song công hoàn toàn giữa máy khách và máy chủ.

Trong hệ thống sử dụng request-response, không có cách nào để client biết khi nào có dữ liệu mới (ngoại trừ bằng cách hỏi server định kỳ hoặc bỏ long-polling).Với Websockets, máy chủ có thể gửi dữ liệu mới bất cứ lúc nào. Vì vậy, websocket luôn là một trong những lựa chọn hàng đầu khi xây dựng các ứng dụng trực tuyến thời gian thực.

Điều quan trọng cần lưu ý là WebSockets chuyển đổi kết nối HTTP sang kết nối WebSocket. Nói cách khác, kết nối WebSocket chỉ sử dụng HTTP để thực hiện handshake (để ủy quyền và xác thực), sau đó kết nối TCP được sử dụng để gửi dữ liệu qua giao thức riêng (hybd).

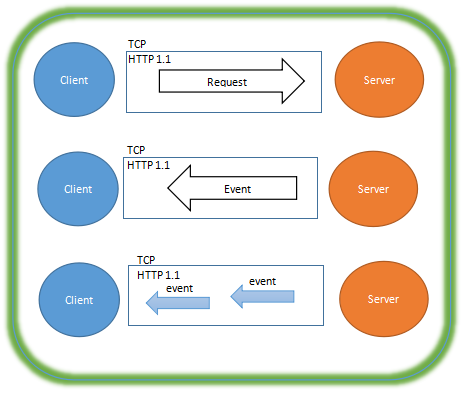


Hình 4.11.Giao thức truyền tin của websocket

* **Server-sent events:**

HTTP cung cấp một cơ chế để truyền liệu từ máy chủ đến máy khách thông qua Server-sent events (hay còn gọi là EventSource).Serve-sent events chỉ cho phép truyền bán công, nghĩa là tại một thời điểm, client hoặc server chỉ có thể gửi hoặc nhận chứ không thể cùng lúc gửi và nhận data

Server-sent events sử dụng luồng octet HTTP thông thường và do đó bị giới hạn ở nhóm kết nối của trình duyệt là 6 kết nối HTTP đồng thời trên mỗi máy chủ. Nhưng nó cung cấp phương pháp chuẩn để đẩy dữ liệu từ máy chủ đến máy khách qua HTTP, giúp các bộ cân bằng và proxy hiểu được sự vượt trội. Ưu điểm lớn nhất giống như của WebSockets là nó chỉ sử dụng một kết nối TCP. Nhược điểm lớn nhất là Serial-sent events gửi máy chủ không cung cấp cơ chế phát hiện các client ngắt kết nối cho đến khi tin nhắn được gửi.



Hình 4.12.Giao thức truyền tin của HTTP

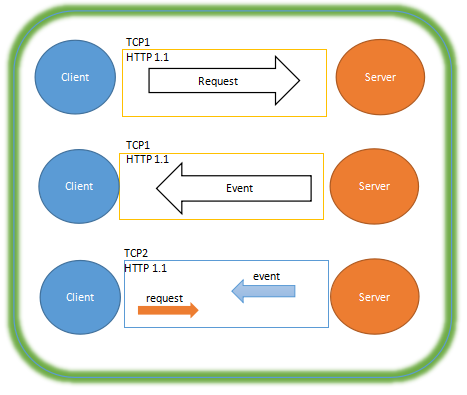
event

event

* **Long-polling:**

Trước khi có Server-sent events, mọi người thường dùng đến các Long-polling để nhận dữ liệu thời gian thực từ máy chủ. Hiện nay,Long-polling vẫn được sử dụng ngày nay trong một số tình huống.

Long-polling hoạt động bằng cách gửi các request HTTP thường xuyên. Khi một request được gửi đến server, nó sẽ phản hồi ngay lập tức nếu có dữ liệu. Nếu không có dữ liệu, server sẽ rút response trong khi client chờ. Nếu trong thời gian đó, dữ liệu mới có sẵn, server sẽ gửi cho client. Khi client nhận được dữ liệu hoặc hết thời gian request, nó sẽ lập tức đưa ra request mới để thiết lập lại kết nối.Long-polling là



Hình 4.13. Giao thức truyền tin Long-polling

Ưu điểm lớn nhất của Long-polling là nó hoạt động trong mọi môi trường và trong mọi trình duyệt. Có thể nói, ứng dụng này tốt hơn cho các ứng dụng có số lượng lớn người dùng đồng thời vì nó không yêu cầu kết nối TCP liên tục đến máy chủ . Mặc dù phải xác thực lại và ủy quyền lại cho client theo từng yêu cầu. Server cần thực hiện một số tổng hợp sự kiện để khắc phục sự cố mất điện khi kết nối. Không có API JS có sẵn cho cơ chế này, không có xử lý kết nối lại, cũng như không phát hiện khi client ngắt kết nối.Nếu dữ liệu cần được gửi đến máy chủ, yêu cầu POST HTTP được thực hiện, giống như với Server-sent events.

### So sánh các giải pháp:

* + 1. So sánh giải pháp phần cứng và phần mềm:

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| Phần cứng | -Đơn giản  -Tốc độ cao, độ trễ thấp, đường truyền ổn định  -Tính bảo mật cao: chỉ những đối tượng được kết nối mới có khả năng truy cập vào hệ thống | -Chi phí cho các thiết bị truyền dẫn rất đắt đỏ  -Lắp đặt khó khăn,dây dẫn chiếm diện tích  -Không thể truyền quá xa do nhiễu  -Bất tiện, chỉ có thể truy cập vào camera ở những máy có kết nối cáp  -Khó khăn trong việc mở rộng, cải tạo hệ thống |
| Phần mềm | -Không tốn nhiều chi phí  -Tiện lợi, có thể truy cập ở bất cử đâu  -Có thể gia tăng số lượng người dùng mà không cần xài bộ chia cáp và thêm dây dẫn.  -Dễ cải tạo, nâng cấp mới hệ thống | -Độ trễ cao, đường truyền phụ thuộc nhiều vào tốc độ Internet  -Tốc độ thấp hơn so với việc sử dụng cáp  -Bảo mật: có khả năng bị xâm nhập bởi các đối tượng bên ngoài |

* Kết luận: Dựa trên tình hình thực tế, chúng em nhận thấy phương pháp sử dụng phần mềm để xây dựng hệ thống phù hợp hơn so với việc sử dụng phần cứng bởi vì:
  + Giá thành rẻ, phù hợp với túi tiền của sinh viên
  + Được áp dụng các kiến thức vào thực tiễn và xây dựng một hệ thống hoàn chỉnh
  + Có thể mở rộng, cải tạo, phát triển để đưa vào ứng dụng thực tế
  + Tuy tốc độ , độ ổn định thấp hơn so với sử dụng phần cứng nhưng vẫn đem lại trải nghiệm tốt cho người dùng.
    1. So sánh các giải pháp truyền lệnh:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Websocket | Server-sent events | Long-polling |
| Số lượng người dùng kết nối từ trình duyệt | 1024 | 6/server | 6/server |
| Phát hiện client ngắt kết nối | Có | Không | Không |
| Kết nối lại khi bị ngắt | Không | Có | Không |
| Phương thức truyền | Song công | Bán công | Bán công |
| Tạo kết nối TCP | 1 lần duy nhất | 1 lần duy nhất | Nhiều lần |
| Ưu điểm | -Độ trễ thấp, thích hợp xây dựng các ứng dụng thời gian thực  -Server và client có thể truyền dữ liệu đồng thời  -WebSockets đi qua hầu hết các tường lửa mà không cần cấu hình lại.  -WebSockets chỉ sử dụngXMLHttpRequest 1 lần, và do đó, các handshakekhông được gửi mỗi lần người dùng cần để có thêm thông tin từ máy chủ. Điều này đã làm giảm tải dữ liệu được gửi đến máy chủ. | -Độ trễ thấp  -Tại một thời điểm, chỉ có 1 bên được gửi dữ liệu | -được các thiết bị hỗ trợ gần như toàn cầu.  -không yêu cầu kết nối TCP liên tục đến máy chủ |
| Nhược điểm | -WebSockets không tự động khôi phục khi kết nối bị chấm dứt | -Tốn dữ liệu để gửi request cho Server mỗi khi Client muốn truyền dữ liệu | -Tốn nhiều tài nguyên nhất để gửi handshake và request mỗi khi Client muốn truyền dử liệu |

* **Nhận xét:**Do hệ thống camera giám sát yêu cầu xử lý thời gian thực, đòi hỏi độ trễ thấp và tối ưu hóa dữ liệu truyền nên chúng em quyết định sử dụng Websocket để xây dựng hệ thống
  + 1. So sánh các giải pháp truyền hình ảnh:

OBS và BigBlueButton cũng gần như tương tự nhau trong khoảng stream video.Tuy nhiên có vài điểm khác biệt như sau:

* + BigBlueButton quản lý người xem tốt hơn.Người tham gia đòi hỏi phải có tài khoản và khi tham gia vào phòng hội nghị để xem camera, thông tin của họ được hiển thị cho người chủ trì.
  + OBS stream trực tiếp hình ảnh camera lên server đã xây dựng. Còn BigBlueButton cần phải sử dụng Iframe để chèn 2 server . Nó có thể làm chậm  trang web và tạo ra lỗ hổng bảo mật

Tuy còn một ít hạn chế, nhưng sử dụng OBS khả quan hơn so với sử dụng BigBlueButton. Theo đề nghị của giáo viên hướng dẫn, chúng em quyết định sử dụng BigBlueButton

# CHƯƠNG 5: QUÁ TRÌNH THỰC HIỆN

## Giải pháp và lựa chọn giải pháp phù hợp:

Giải pháp điều khiển Camera gồm có :

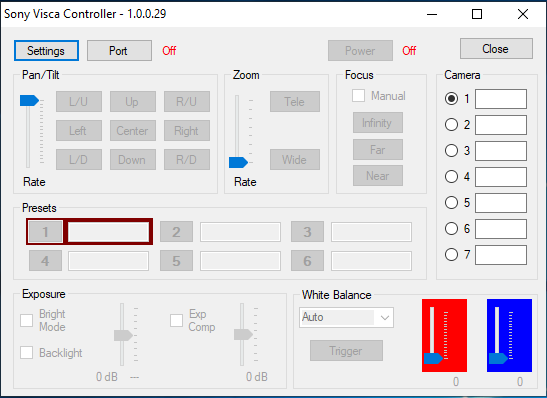
* Giải pháp dựa trên thiết bị phần cứng: hệ thống Joy Sticks điều khiển từ xa của các hãng Sony,.. hay các bên thứ ba có thể cung cấp.

Ưu điểm:

* + Độ trễ thấp hoặc rất thấp, gần như không có nếu khoảng cách kết nối ngắn,
  + Tính bảo mật cao

Nhược điểm:

* Mỗi bộ thiết bị điều khiển camera bao gồm bộ thiết bị chuyên dụng như: hiết bị đầu cuối (MCU, endpoint), dây nối, bảng điều khiển. Nên chí phí cho các bộ thiết bị chuyên dụng này rất cao.
* Cần một không gian sẵn có để lắp đặt thiết bị.
* Các hệ thống phần cứng kém linh động hơn hệ thống phần mềm trong việc tương thích với công nghệ khác.
* Không khả thi khi triển khai từ việc điều khiển từ một khu vực ở xa. Ví dụ : điều khiển và lấy hình ảnh ca phẫu thuật ở TP HCM từ Hà Nội.
* Giải pháp dựa trên phần mềm: hiện nay có nhiều lập trình viên đã xây dựng nên các chương trình mô phỏng lại giao diện điều khiển camera. Các chương trình này có hoàn toàn đầy đủ các chức năng của một chiếc remote điều khiển từ xa. Vì nhà cung cấp có đính kèm các file chứa các mã lệnh Hexa hoặc có sẵn trên website của hãng, dựa vào các lệnh này, việc xây dựng nên một chương trình là hoàn toàn khả thi.



Hình 5.1 Giao diện điều khiển camera của một phần mềm được tự xây dựng.

Ưu điểm:

* + Chi phí thấp hoặc miễn phí.
  + Sử dụng cho hầu hết các thiết bị như laptop, mobile, tablet.
  + Độ di động cao.
  + Tính tương thích với các công nghệ khác cao. Ví dụ : Video Conference

Nhược điểm:

* + Phải tốn công xây dựng, cần nhiều công nghệ.
  + Cần nhiều ngôn ngữ lập trình để xây dựng nên các lỗi xảy ra khi khác biệt môi trường, ngôn ngữ xảy ra thường xuyên. Các bản hướng dẫn đa số đều cũ nên các lỗi mới xuất hiện trong quá trình xây dựng tương đối nhiều.
  + Tính bảo mật không cao

→ Sau khi tìm hiểu các giải pháp cho việc điều khiển camera, giải pháp được lựa chọn hợp lý nhất cho các bệnh viện tỉnh cùng như bệnh viện thành phố là giải pháp xây dựng dựa trên phần mềm. Giải pháp này phù hợp với các yêu cầu cần có và chủ động cho người tham gia từ xa.

## Các chức năng cần có:

Những chức năng cần có cho một hệ thống điều khiển camera từ xa :

- Khu vực điều khiển: nơi người dùng có thể sử dụng để điều khiển camera, có thể mô phỏng lại hình ảnh 2D của remote control để người dùng thuận tiện thao tác trên các nút bấm quen thuộc.

- Khu vực cửa sổ video: hình ảnh được tải từ webcam lên trang web với thời gian thực, đầy đủ mọi chuyển động mà camera thu được, xác minh camera có thể lấy được nội dung và chuyển động có độ trễ thấp.

## Xây dựng server:

Sử dụng IDE để tạo Spring boot để xây dựng server cho Websocket:

* Bước 1: tạo thư mục theo cấu trúc như sau

└── src

└── main

└──java

└──hello

* Bước 2:Xây dựng thư viện

Chúng ta sử dụng thư viện Spring Boot để hỗ trợ xây dựng websocket

Mở file build gradle,**build.gradle** là file để bạn cấu hình các thư viện sẽ được sử dụng trong project

Plugin Spring Boot cung cấp nhiều tính năng tiện lợi:

Nó thu thập tất cả các jar trên đường dẫn và xây dựng một "über-jar" duy nhất, có thể chạy được, giúp thuận tiện hơn khi thực hiện và vận chuyển code của bạn.

Nó tìm kiếm phương thức static void () công khai để gắn cờ là một class để chạy

Nó cung cấp một phương pháp tích hợp sẵn để đặt số phiên bản phù hợp với các nhánh Spring Boot

Bạn có thể ghi đè bất kỳ phiên bản nào bạn muốn, nhưng nó sẽ mặc định là Boot phiên bản đã chọn

* Bước 3:Tạo class hiển thị dữ liệu:

Dịch vụ sẽ chấp nhận các thông báo chứa tên trong thông báo STOMP có phần thân là đối tượng

JSON

Dịch vụ sẽ chấp nhận các thông báo chứa tên trong thông báo STOMP có phần thân là đối tượng

JSON.

Khi nhận được tin nhắn và trích xuất tên, hê thống sẽ xử lý nó bằng cách tạo một lời chào và hiển

thị lời chào đó trên một hàng đợi riêng mà client đăng ký. Lời chào cũng sẽ là một đối tượng JSON.

* Bước 4: Tạo class xử lý tin nhắn

Tạo một bộ điều khiển để nhận tin nhắn Hello và gửi Greeting Mesasage .

Theo cách tiếp cận của Spring, để làm việc với nhắn tin STOMP, các tin nhắn STOMP có thể được định tuyến đến các lớp @Control. GreetingController được ánh xạ để xử lý các thông báo đến đích "/hello".

@MessageMapping đảm bảo rằng nếu một tin nhắn được gửi đến “/hello”, thì phương thức *greeting()* sẽ được gọi.

*Payload* của tin nhắn được liên kết với đối tượng HelloMessage và chuyển vào *greeting()*.

Trong nội bộ, việc thực hiện phương thức mô phỏng độ trễ xử lý bằng cách làm cho luồng ngủ trong 1 giây. Điều này là để chứng minh rằng,sau khi client gửi tin nhắn, server sẽ có đủ thời gian cần để có thể xử lý tin nhắn không đồng bộ . Client có thể tiếp tục với bất kỳ công việc nào cần làm mà không cần chờ phản hồi.

Sau 1 giây, phương thức *greeting()* tạo một đối tượng Greeting và trả về . Giá trị trả về được phát cho tất cả những người đăng ký "/ topic / greetings" như được chỉ định trong chú thích @SendTo. Lưu ý rằng tên từ thông báo đầu vào được lọc vì trong trường hợp này, nó sẽ được lặp lại và được kết xuất lại trong DOM trình duyệt ở phía client.

* Bước 5: cấu hình Spring cho tin nhắn Stomp

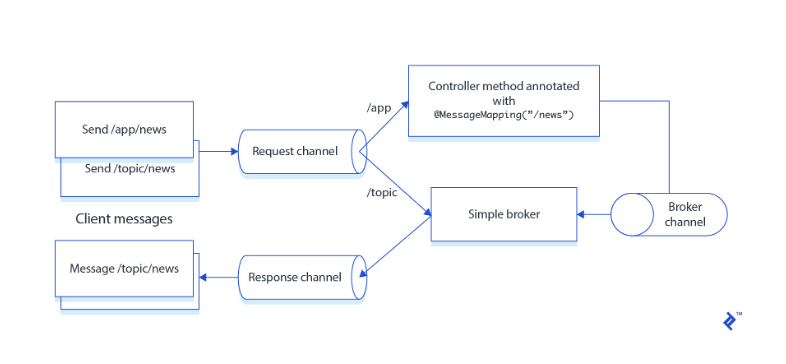
Sau khi các thành phần thiết yếu của dịch vụ được tạo, cấu hình Spring để bật nhắn tin WebSocket và STOMP.

WebSocketConfig được chú thích bằng @Configuration để chỉ ra rằng đó là lớp cấu hình Spring. Nó cũng được chú thích là @EnableWebSocketMessageBroker.

@EnableWebSocketMessageBroker cho phép xử lý tin nhắn WebSocket, được hỗ trợ bởi message broker

Phương thức configureMessageBroker () thực hiện mặc định trong WebSocketMessageBrokerConfigker để định cấu hình message broker. Nó bắt đầu bằng cách gọi enableSimpleBroker () để cho phép một nhà môi giới tin nhắn dựa trên bộ nhớ đơn giản mang greeting() trở lại client trên các đích đến có tiền tố "/ topic". Nó cũng chỉ định tiền tố "/app" cho các tin nhắn được ràng buộc cho các phương thức có chú thích @ MessageMapping. Tiền tố này sẽ được sử dụng để xác định tất cả các ánh xạ tin nhắn; ví dụ: "/ app / hello" là điểm cuối mà phương thức GreetingController.greeting() được ánh xạ để xử lý.

Phương thức registerStompEndpoint () đăng ký endpoint"/ gs-guide-websocket", cho phép các tùy chọn dự phòng SockJS để có thể sử dụng các phương tiện vận chuyển thay thế nếu WebSocket không khả dụng. Ứng dụng khách SockJS sẽ cố gắng kết nối với "/ gs-guide-websocket" và sử dụng phương truyền tải tốt nhất hiện có (websocket, xhr-streaming, xhr-polling, v.v.).



Hình 5.2.Truyền tin phía server

* Bước 6: Tạo trình duyệt :

Sau khi xây dựng server, chúng ta tiến hành tạo ứng dụng khách JavaScript sẽ gửi tin nhắn đến và nhận tin nhắn từ phía máy chủ.

Sau khi hoàn thành, chúng ta sẽ được 1 trang web với phần ngoại hình bao gồm các nút bấm như sau:

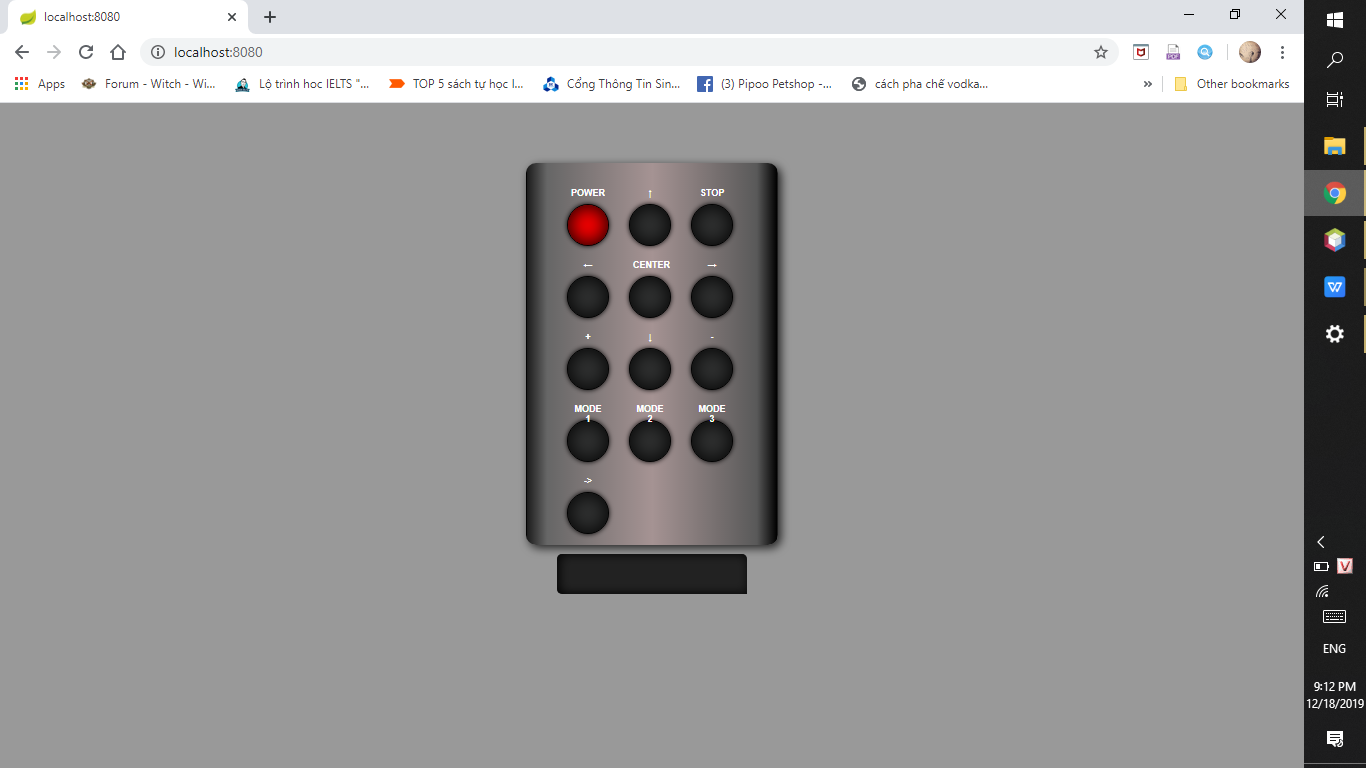
POWER ↑ Stop ← Center → + ↓ - Mode 1 Mode 2 Mode 3 ->

Tệp HTML này nhập các thư viện javascript SockJS và STOMP sẽ được sử dụng để liên lạc với máy chủ của chúng tôi bằng STOMP qua websocket. Chúng ta cần tạo một file app.js chứa logic của ứng dụng client.

Phần chính của tệp JavaScript này cần chú ý là các hàm connect() và sendName ().

Hàm connect () sử dụng SockJS và stomp.js để mở kết nối tới "/ gs-guide-websocket", đây là nơi máy chủ SockJS đang chờ kết nối. Sau khi kết nối thành công, client đăng ký đích "/ topic / greetings", nơi server sẽ gửi bản tin nhắn greeting . Khi một greeting được nhận ở đích đó, nó sẽ nối một phần từ đoạn vào DOM để hiển thị thông báo greeting.

Hàm sendName () lấy tên mà người dùng đã nhập và sử dụng ứng dụng khách STOMP để gửi nó đến đích "/ app / hello"



Hình 5.3.Giao diện trang web được xây

* Bước 7: chạy ứng dụng

Mặc dù có thể đóng gói dịch vụ này dưới dạng tệp WAR truyền thống để triển khai cho máy chủ ứng dụng bên ngoài, ta có thể tạo ra một ứng dụng độc lập. Chỉ cần đóng gói mọi thứ trong một tệp JAR duy nhất, có thể thực hiện được, được điều khiển bởi một phương thức main () cũ của Java. Sử dụng hỗ trợ Spring, để nhúng bộ chứa Tomlet servlet làm thời gian chạy HTTP, thay vì triển khai đến một phiên bản bên ngoài.

@SpringBootApplication là một chú thích bao gồm những thứ dưới đây:

@Configuration: Gắn thẻ lớp là nguồn định nghĩa bean cho bối cảnh ứng dụng.

@EnableAutoConfiguration: Yêu cầu Spring Boot bắt đầu thêm các bean dựa trên cài đặt đường dẫn lớp, các loại bean khác và các cài đặt thuộc tính khác nhau. Ví dụ: nếu spring-webmvc nằm trên đường dẫn lớp, chú thích này gắn cờ ứng dụng dưới dạng một ứng dụng web và kích hoạt các hành vi chính, chẳng hạn như thiết lập một DispatcherServlet.

@ComponentScan: Yêu cầu Spring tìm kiếm các thành phần, cấu hình và dịch vụ khác trong gói hello, để nó tìm bộ điều khiển.

Phương thức main () sử dụng phương thức SpringApplication.run () để khởi chạy một ứng dụng. Ứng dụng web này được xây dựng 100% trên nền tảng java

## Xây dựng client:

* Xây dựng class Serial Port để truyền dữ liệu nối tiếp từ Client sang camera
* Xây dựng Client websocket:

Mở kết nối websocket

Sau khi kết nối được mở, chúng ta có thể gửi tin nhắn đến điểm đến.Tin nhắn sẽ được gửi đến tất cả Client đã đăng ký cùng topic

Spring giúp cho việc gửi tin nhắn trở nên dễ dàng. Chúng ta chỉ cần 1 chú thích @SendToUser. Sau đó điểm đến sẽ được xử lý bởi UserDestinationMessageHandler, chạy trên định danh phiên làm việc. Ở phía Client,khi client đăng ký điểm đến với tiền tố /user,đích đến này được chuyển thành đích duy nhất cho người dùng này. Về phía server, đích người dùng được giải quyết dựa trên Principle của người dùng.

## Sơ đồ giải thuật:

### **.Truyền lệnh:**

Server lắng nghe request

Tạo kết nối websocket

Server chờ nhận lệnh

Không

Có

Server truyền lệnh cho client

Client truyền lệnh camera

### **Truyền hình ảnh:**

Bắt đầu

Nhận hình ảnh từ camera

BigBlueButton server

Nhập tên đăng nhập

Stop

Kết nối các Client

Có

Kếtthúc

phiên họp

Trao đổi media

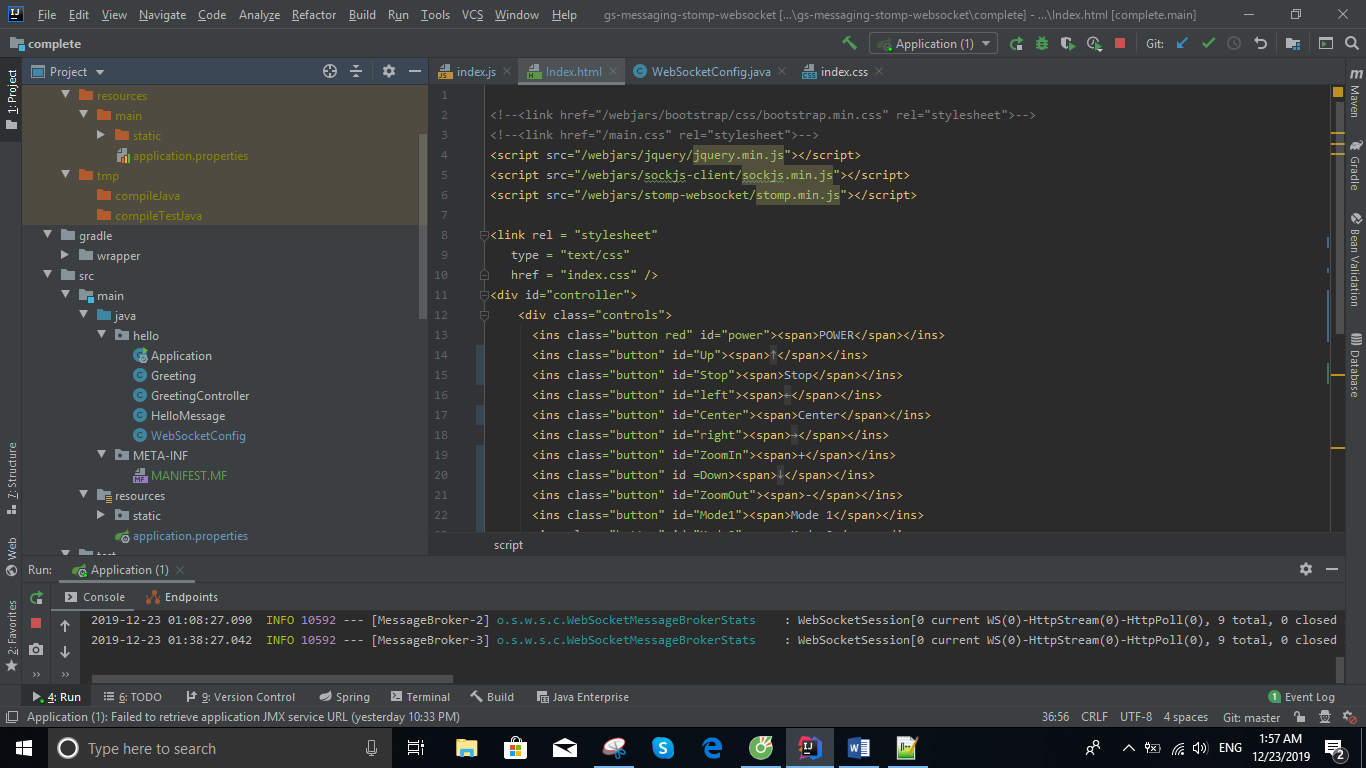
Hình

# CHƯƠNG 6 :KẾT QUẢ THỰC HIỆN



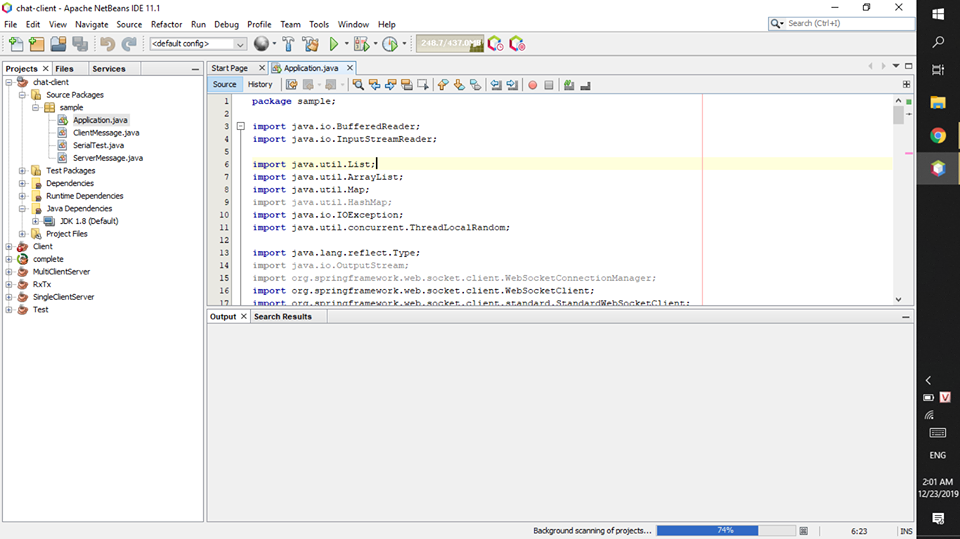
## Truyền lệnh:

Xây dựng websocket server và web page

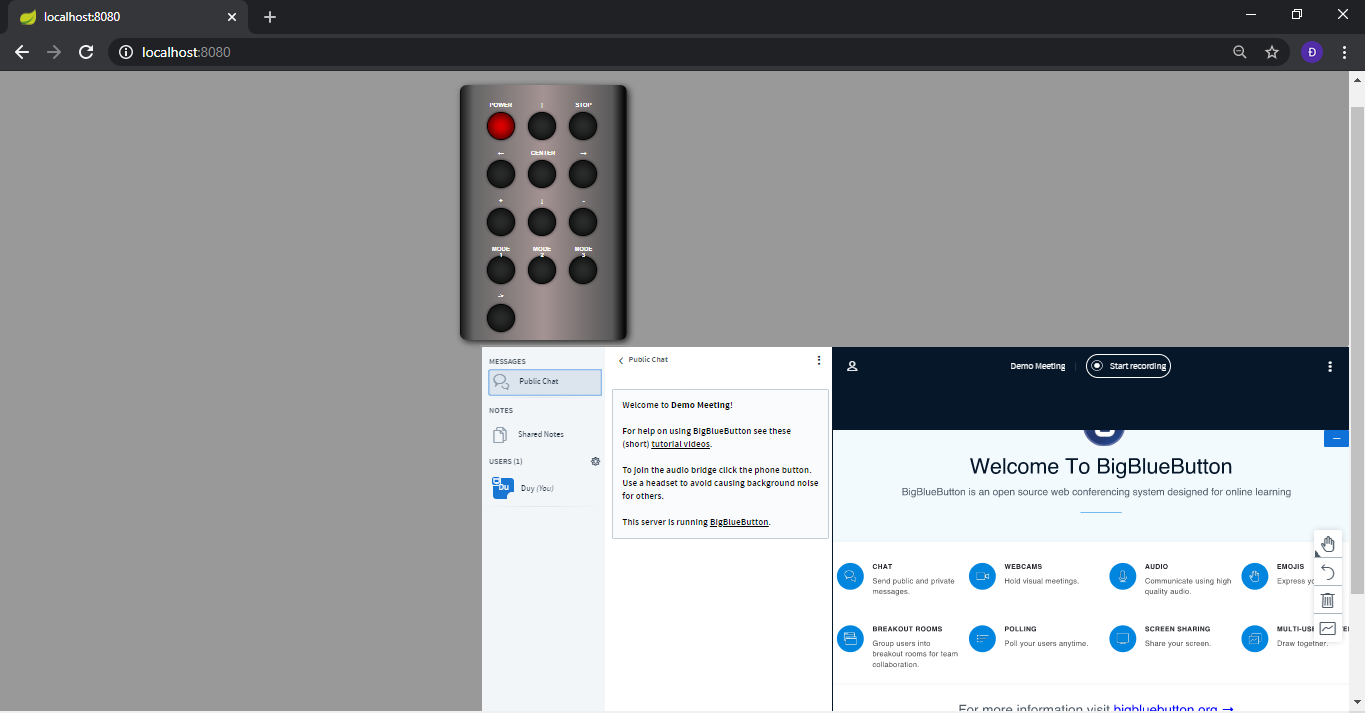


Hình 6-1.Xây dựng server Websocket

Xây dựng websocket client:



Hình 6-2.Xây dựng websocket client



Hình 6-2.Giao diện webpage

## Truyền hình ảnh:

# CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết Luận

Qua đề Luận văn tốt nghiệp đã giúp em có nhiều kiến thức và kinh nghiệm:

* Hiểu rõ lợi ích của việc xây dựng hệ thống điều khiển Camera từ xa hiện nay:
  + Tiết kiệm chi phí, thời gian cho các bệnh viện.
  + Hiệu quả cao hơn so với bộ điều khiển rời khi phải điều khiển từ khoảng cách xa
  + Khả năng mở rộng quy mô quan sát khi có nhiều bên cùng tham gia quan sát và điều khiển
* Hiểu về cách xây dựng và hoạt động khi xây dựng một Layout Webpage cơ bản
* Hiểu biết về ngôn ngữ lập trình Java, Java Script và một số ngôn ngữ khác
* Xây dựng hệ thống điều khiển trên nền tảng HTML

Kết quả đạt được của việc xây dựng hệ thống điều khiển Camera từ xa :

* Ưu điểm :
  + Xây dựng hệ thống điều khiển trên nền tảng mã nguồn mở miễn phí nên tiết kiệm một khoảng chi phí rất lớn cho việc đầu tư cơ sở hạ tầng trang thiết bị từ các công ty khác.
  + Hệ thống sẵn sàng, đã được tối ưu hóa mô phỏng lại chiếc điều khiển có sẵn, dễ dàng sử dụng cho các bác sĩ, y tá tham gia quan sát hội chuẩn.
  + Hê thống sử dụng được trên các thiết bị như máy tính để bàn, điện thoại, máy tính bảng.
  + Khả năng kết nối số lượng nhiều thiết bị tốt, nhiều thiết bị có thể cùng tham gia kết nối và điều khiển cùng một Camera
* Khuyết điểm :
  + Hệ thống còn độ trễ không mong muốn do tốc độ mạng, cấu trúc tập lệnh gửi đi, cách xây dựng cấu trúc chưa tối ưu hóa hệ thống
  + Kết nối từ người dùng cuối đến thiết bị đôi khi sẽ xảy ra lỗi không kết nối được, không tự connect lại khi bị mất kết nối.

## Hướng phát triển:

Một số tồn tại của hệ thống cần được khắc phục :

* Hiện tại hệ thống còn sử dụng kiến trúc cơ bản, các kết nối chưa được tối ưu để giảm độ trễ khi truyền câu lệnh .
* Layout còn đơn giản, chưa thực thi hết tất cả các lệnh mà mới chỉ một số lệnh cơ bản.
* Các chọn hệ thống streaming phức tạp, nhiều lỗi xảy ra do xung đột môi trường và ngôn ngữ xây dựng.
* Hệ thống chưa lên được Internet để có thể điều khiển từ khoảng cách địa lý xa, mới chỉ demo được trên hệ thống Local.

Giải pháp cải thiện :

* Sử dụng một ngôn ngữ khác khi lập trình như Python. Cấu trúc hiện giờ của tập lệnh đang sử dụng là gửi lệnh điều khiển, delay rồi gửi lệnh stop. Do cấu trúc của Javascript là gửi động thời các lệnh nên không thể kiểm soát lệnh nào sẽ được bắt trước và sau, vừa gây nên độ trễ, vừa gây nên việc sai lệch do bắt sai thứ tự lệnh.
* Cải thiện Layout, cập nhật thêm các lệnh và chức năng, mở rộng mô hình và cải thiện giao diện người dùng.
* Có nhiều giải pháp để cải thiện hệ thống streaming như : sử dụng OPS, hoặc xây dựng phương tiện streaming sử dụng ngôn ngữ Python. Hệ thống stream hiện tại đang xây dựng trên ngôn ngữ Java, cồng kềnh và khó để đồng bộ được với hệ thống đang xây dựng, cách đơn giản nhất để kết hợp cả việc điều khiển và quan sát trên cùng một Webpage là nhúng Iframe và cũng xảy ra lỗi phải sửa do sai khác môi trường React và ReactJS.
* Cải thiện hệ thống điều khiển từ xa bằng các công cụ miễn phí như Heruko để đăng ký một tên địa chỉ internet miễn phí hoặc dùng các công cụ có trả phí như Google Cloud, Amazon Web Services hay Microsoft Azure hoặc giải pháp NAT.

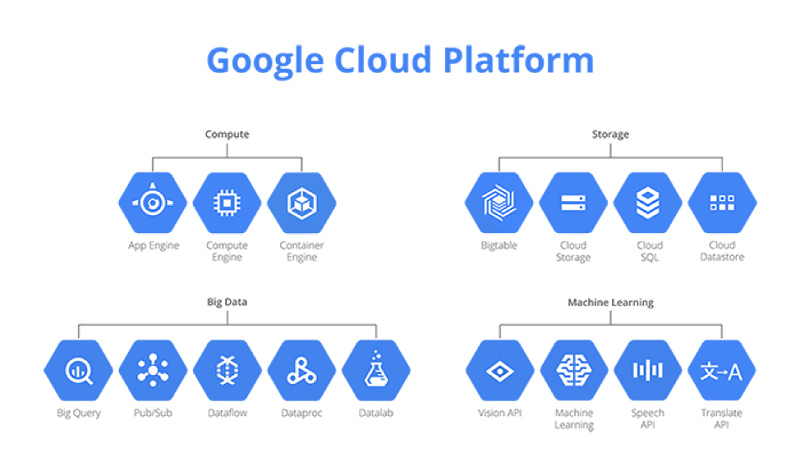
**Giải pháp Google Cloud**



Hình 7.1 Dịch vụ Google Cloud Platform

Google Cloud Platform (GCP)là một hệ thống các dịch vụ điện toán đám mây của Google. Google Cloud Platform gồm các dịch vụ lưu trữ phục vụ cho việc tính toán, lưu trữ, phát triển ứng dụng. Dịch vụ này được lưu trữ và chạy trên phần cứng của Google nhằm phục vụ cho các nhà phát triển phần mềm, quản trị viên điện toán đám mây, các chuyên gia công nghệ thông tin của doanh nghiệp, các nhà phát triển website thông qua mạng công cộng cũng như những mạng riêng.

Google Cloud Platform giúp giải quyết tất cả các vấn đề thiết yếu bao gồm Big Data, Storage, Compute Engine, Networking, Management, Developer Tools, Mobile, …. Do đó, doanh nghiệp chỉ không cần tốn công phát triển các hệ thống bảo mật.v.v.. mà không cần phải quan tâm đến hệ thống bên dưới.



Hình 7.2.Các dịch vụ và chức năng của Google Cloud Platform

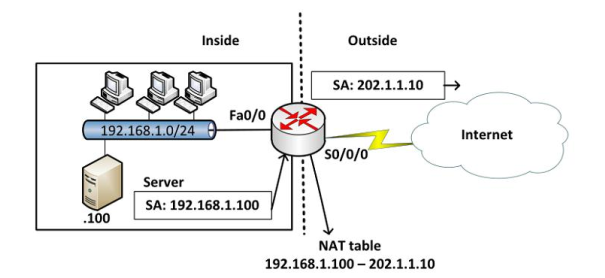
Trong đó, các dịch vụ Google có thể được dùng để phục vụ cho việc phát triển hệ thống điều khiển

* Google Compute Engine : lưu trữ các máy ảo, tồn tại hoàn toàn dưới dạng phần mềm
* Google App Engine : cung cấp cho các nhà phát triển các công cụ và ngôn ngữ như Python, PHP và nhiều ngôn ngữ khác để xây dựng và triển khai các dự án web trực tuyến trên đám mây. Đây là sự phát triển dựa trên sự phát triển của điện toán đám mây, để xây dựng, triển khai và phát triển ứng dụng từ xa.

**Giải pháp NAT**

NAT ([Network Address Translation](https://vi.wikipedia.org/wiki/Bi%2525C3%2525AAn_d%2525E1%2525BB%25258Bch_%2525C4%252591%2525E1%2525BB%25258Ba_ch%2525E1%2525BB%252589_m%2525E1%2525BA%2525A1ng)) là một kỹ thuật cho phép chuyển đổi từ một địa chỉ IP này thành một địa chỉ IP khác. Thông thường, NAT được dùng phổ biến trong mạng sử dụng địa chỉ cục bộ, cần truy cập đến mạng công cộng (Internet). Vị trí thực hiện NAT là router biên kết nối giữa hai mạng. Có ba dạng NAT chính là Static Nat, Dynamic Nat và Nat Overload.

Static NAT được dùng để chuyển đổi một [địa chỉ IP](https://www.totolink.vn/article/74-ip-la-gi-cach-xac-dinh-dia-chi-ip-tren-may-tinh.html) này sang một địa chỉ khác một cách cố định, thông thường là từ một địa chỉ cục bộ sang một địa chỉ công cộng và quá trình này được cài đặt thủ công, nghĩa là địa chỉ ánh xạ và địa chỉ ánh xạ chỉ định rõ ràng tương ứng duy nhất.   
Static NAT rất hữu ích trong trường hợp những thiết bị cần phải có địa chỉ cố định để có thể truy cập từ bên ngoài Internet. Những thiết bị này phổ biến là những Server như Web, Mail,...



Hình 7.3.Mô hình Static Nat.

Ưu điểm :

* Cục bộ, sẵn sàng. Do khác với Google Cloud ( máy chủ đặt ở Singapore) thì chỉ cần hệ thống mạng có thể triển khai được Nat là có thể phát triển, độ trễ thấp
* Miễn phí

Nhược điểm :

* Lỗ hổng bảo mật, Hacker
* Khả năng chịu tải truy cập không cao.