

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KỸ THUẬT MÁY TÍNH  
—o0o—

Lập Trình Nhúng Căn Bản - CE211

BÁO CÁO CUỐI KỲ ĐỒ ÁN MÔN  
HỌC

## **GAME BATTLE SHIP**

**Giảng viên hướng dẫn:**

Trần Ngọc Đức

**Sinh viên thực hiện:**

Trần Minh Tân – 15520771

Bùi Anh Khoa – 15520364

Nguyễn Trần Kha - 15520329

## Mục lục nội dung

1. Giới thiệu.....	4
1.1. Kiến trúc tổng quan.....	4
1.2. Chip Nuvoton ARM Cortex –M0.....	4
1.3. ESP8266.....	4
1.4. Webserver.....	5
2. Battle Ship .....	5
2.1. Chương trình nhúng trên gamepad .....	5
2.1.1. Gửi chuỗi điều khiển lên webserver .....	5
2.1.2. Nhận giá trị thông báo trả về và xử lý rung .....	6
2.2. Xử lý truyền nhận server – client .....	7
2.3. Xử lý sự kiện trên webserver .....	8
2.4. Giao diện web .....	10
3. Kiểm tra .....	12
4. Kết luận .....	12

## Mục lục hình ảnh

<b>Hình 1.</b> Kiến trúc tổng quan của hệ thống.....	4
<b>Hình 2.</b> NUC140VE3CN.....	4
<b>Hình 3.</b> ESP8266.....	4
<b>Hình 4.</b> Lưu đồ thuật toán trên gamepad.....	5
<b>Hình 5.</b> Lưu đồ thuật toán hiệu chỉnh chuỗi được gửi lên webserver thông qua ESP8266.....	6
<b>Hình 6.</b> Lưu đồ thuật toán nhận giá trị trả về và xử lý rung .....	7
<b>Hình 7.</b> Lưu đồ thuật toán xử lý truyền nhận giữa client – server.....	7
<b>Hình 8.</b> Các bước xử lý sự kiện trên webserver .....	8
<b>Hình 9.</b> Lưu đồ thuật toán hàm tách dữ liệu.....	9
<b>Hình 10.</b> Lưu đồ thuật toán hàm xử lý sự kiện nút nhấn .....	10
<b>Hình 11.</b> Giao diện login .....	11
<b>Hình 12.</b> Giao diện signup.....	11
<b>Hình 13.</b> Giao diện trong game .....	11

## 1. Giới thiệu

### 1.1. Kiến trúc tổng quan



**Hình 1.** Kiến trúc tổng quan của hệ thống

Kiến trúc của hệ thống gồm 3 phần chính:

- Nuvoton ARM Cortex –M0 thực hiện xử lý các sự kiện nút nhấn (di chuyển, đặt tàu, bắn tàu), điều khiển rung khi tàu bị bắn trúng, gửi và nhận giá trị thông qua ESP8266 bằng giao tiếp UART.
- Module wifi ESP8266 thực hiện việc gửi các giá trị nhận được từ Nuvoton ARM Cortex –M0 đến webserver và ngược lại thông qua giao thức TCP.
- Webserver thực hiện xử lý các giá trị tương ứng với các sự kiện nút nhấn của Nuvoton ARM Cortex –M0 nhận được từ ESP8266 và hiển thị lên giao diện web.

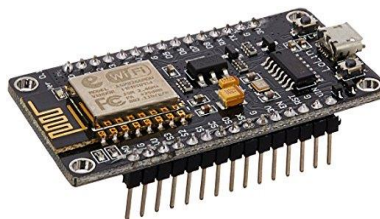
### 1.2. Chip Nuvoton ARM Cortex –M0

Nuvoton ARM Cortex -M0 là dòng chip 32-bit, dòng vi điều khiển này được thiết kế nhúng tối ưu hóa cho các ứng dụng vi xử lý MCU. Dòng ARM Cortex-M0 là dòng vi điều khiển lõi ARM có kích thước nhỏ nhất, tiêu thụ điện năng thấp nhất và có kiến trúc được sắp xếp hợp lý tương thích với việc sử dụng tools nạp của các hãng khác để phát triển các ứng dụng.



**Hình 2.** NUC140VE3CN

### 1.3. ESP8266



**Hình 3.** ESP8266

ESP8266 là module wifi năng lượng thấp, rất phù hợp cho các ứng dụng liên quan đến IoT.

ESP8266 hỗ trợ thu/phát wifi, giao thức TCP/IP và ESP8266 cũng bao gồm một MCU để hỗ trợ việc tính toán.

#### 1.4. Webserver

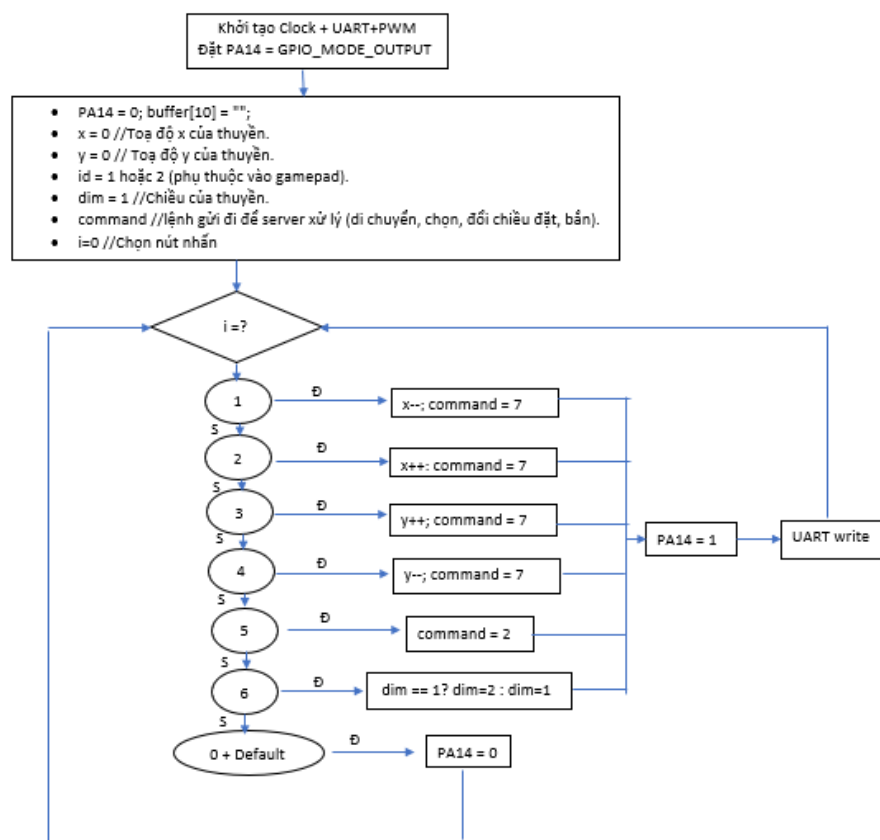
Các công cụ và ngôn ngữ được dùng để tạo webserver cho game battle ship gồm có: Apache, php, mysql, python, css và html, trong đó:

- Apache là một phần mềm web server có nhiệm vụ tiếp nhận request từ trình duyệt người dùng sau đó chuyển giao cho PHP xử lý và gửi trả lại cho trình duyệt.
- MySQL cũng tương tự như các hệ quản trị cơ sở dữ liệu khác (Postgress, Oracle, SQL server...) đóng vai trò là nơi lưu trữ và truy vấn dữ liệu.
- PHP được sử dụng để truy vấn trên database và gửi trả kết quả hiển thị lên giao diện.
- Python dùng để xử lý các sự kiện nhận được từ ESP8266 và lưu trạng thái vào database.
- Css và html dùng để tạo giao diện web.

## 2. Battle Ship

### 2.1. Chương trình nhúng trên gamepad

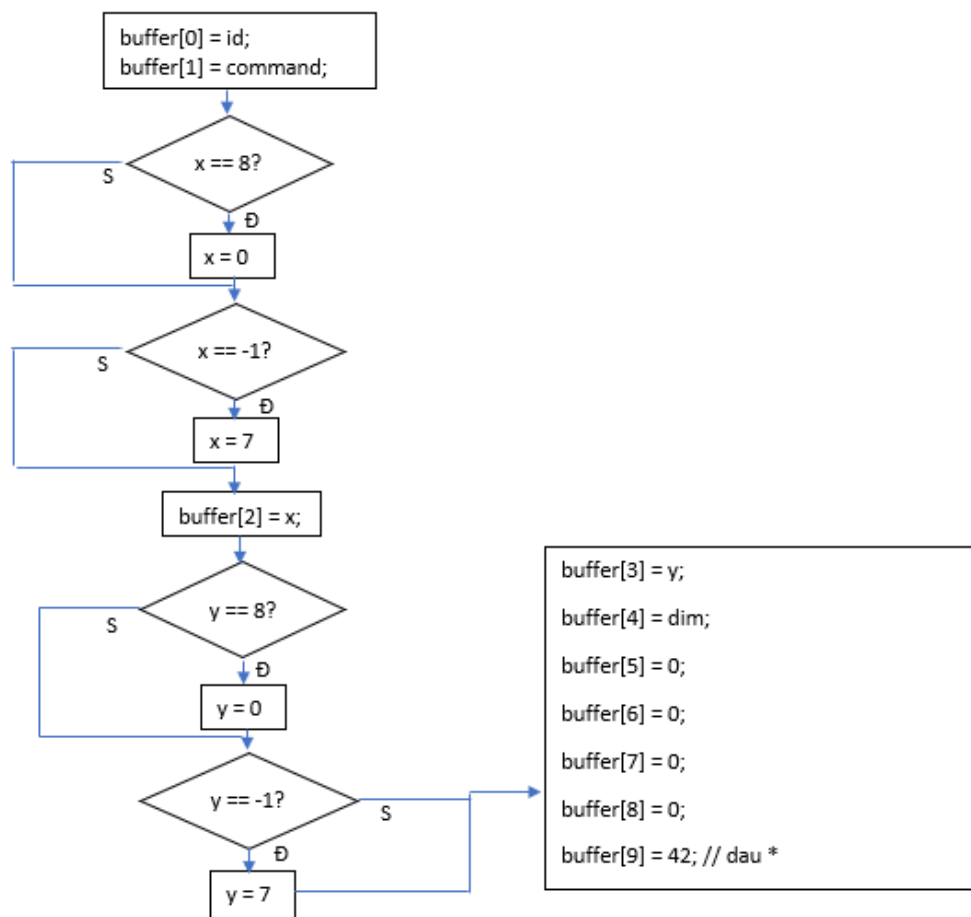
#### 2.1.1. Gửi chuỗi điều khiển lên webserver



**Hình 4.** Lưu đồ thuật toán trên gamepad

**Hình 4** mô tả thuật toán được sử dụng trên gamepad. Đầu tiên, xung clock và uart được khởi tạo, port PA14 nối với led được set là output. Sau đó, các giá trị điều khiển để gửi lên webserver thông qua ESP8266 được khởi tạo gồm:

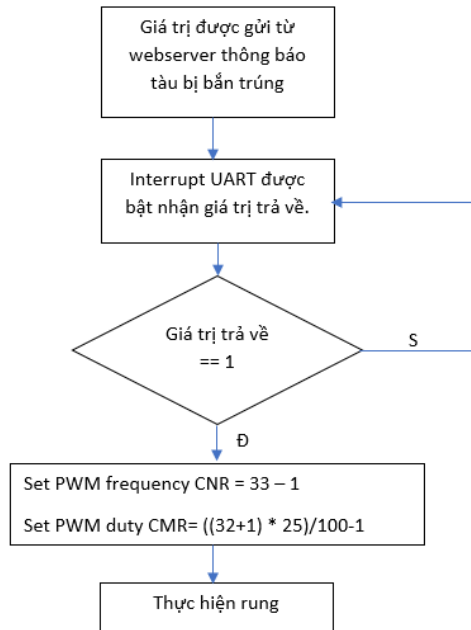
- ✓ Biến  $x$  chứa tọa độ x của thuyền.
- ✓ Biến  $y$  chứa tọa độ y của thuyền.
- ✓ Biến  $id$  là id của gamepad, được nạp cứng trong gamepad.
- ✓ Biến  $dim$  là chiều của thuyền (ngang hoặc dọc).
- ✓ Biến command là chứa lệnh cần thực hiện (lên, xuống, trái, phải, đặt thuyền, xoay, bắn). Nếu là các lệnh di chuyển thì command là 7, đặt thuyền hoặc bắn thuyền command là 2.
- ✓ Buffer[10] chứa chuỗi ký tự được gửi lên webserver thông qua ESP8266 khi thực hiện UART write. **Hình 5** mô tả các thành phần trong chuỗi buffer được gửi đi và hiệu chỉnh lại tọa độ cho chính xác khi các tọa độ vượt ra khỏi vùng biên (trên, dưới, trái và phải).



**Hình 5.** Lưu đồ thuật toán hiệu chỉnh chuỗi được gửi lên webserver thông qua ESP8266

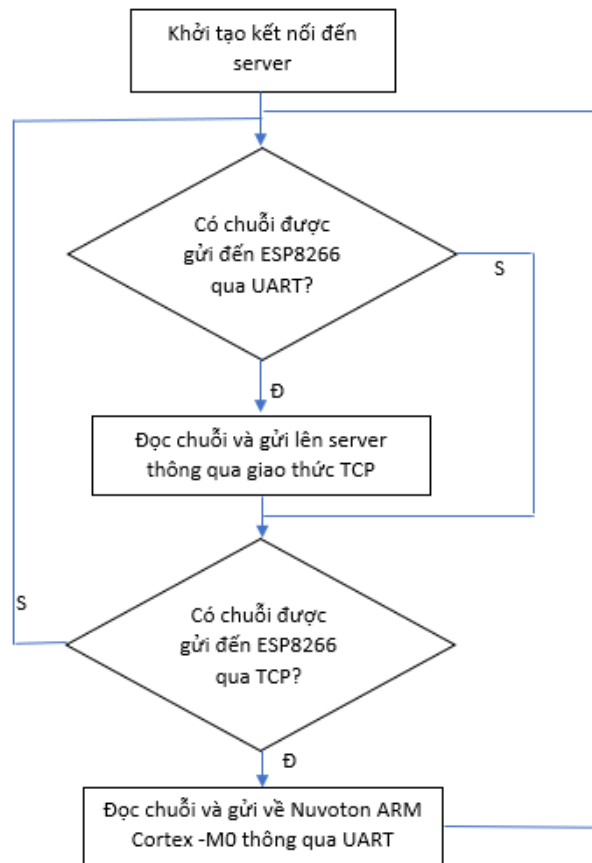
### 2.1.2. Nhận giá trị thông báo trả về và xử lý rung

Khi tàu bị bắn trúng, webserver sẽ gửi giá trị về cho gamepad thông qua ESP8266 và xử lý rung.



**Hình 6.** Lưu đồ thuật toán nhận giá trị trả về và xử lý rung

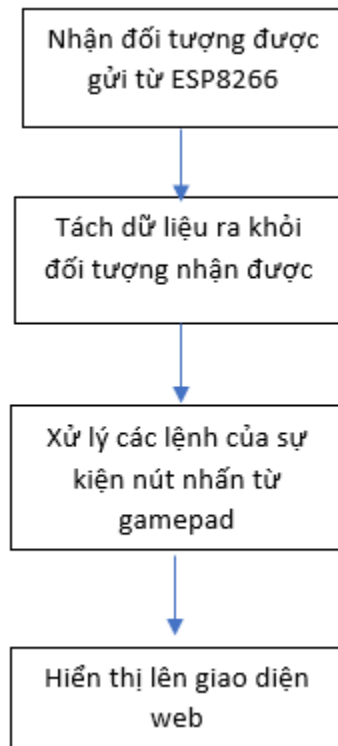
## 2.2. Xử lý truyền nhận server – client



**Hình 7.** Lưu đồ thuật toán xử lý truyền nhận giữa client – server

**Hình 7** mô tả thuật toán xử lý truyền nhận giữa client và server. Đầu tiên, kết nối giữa ESP8266 và server được khởi tạo. Khi có một chuỗi được gửi đến ESP8266 qua UART, chuỗi sẽ được đọc và gửi lên server. Ngược lại khi có một chuỗi được gửi đến ESP8266 từ server thông qua giao thức TCP, chuỗi sẽ được đọc và gửi đến chip Nuvoton thông qua kết nối UART.

### 2.3. Xử lý sự kiện trên webserver



**Hình 8.** Các bước xử lý sự kiện trên webserver

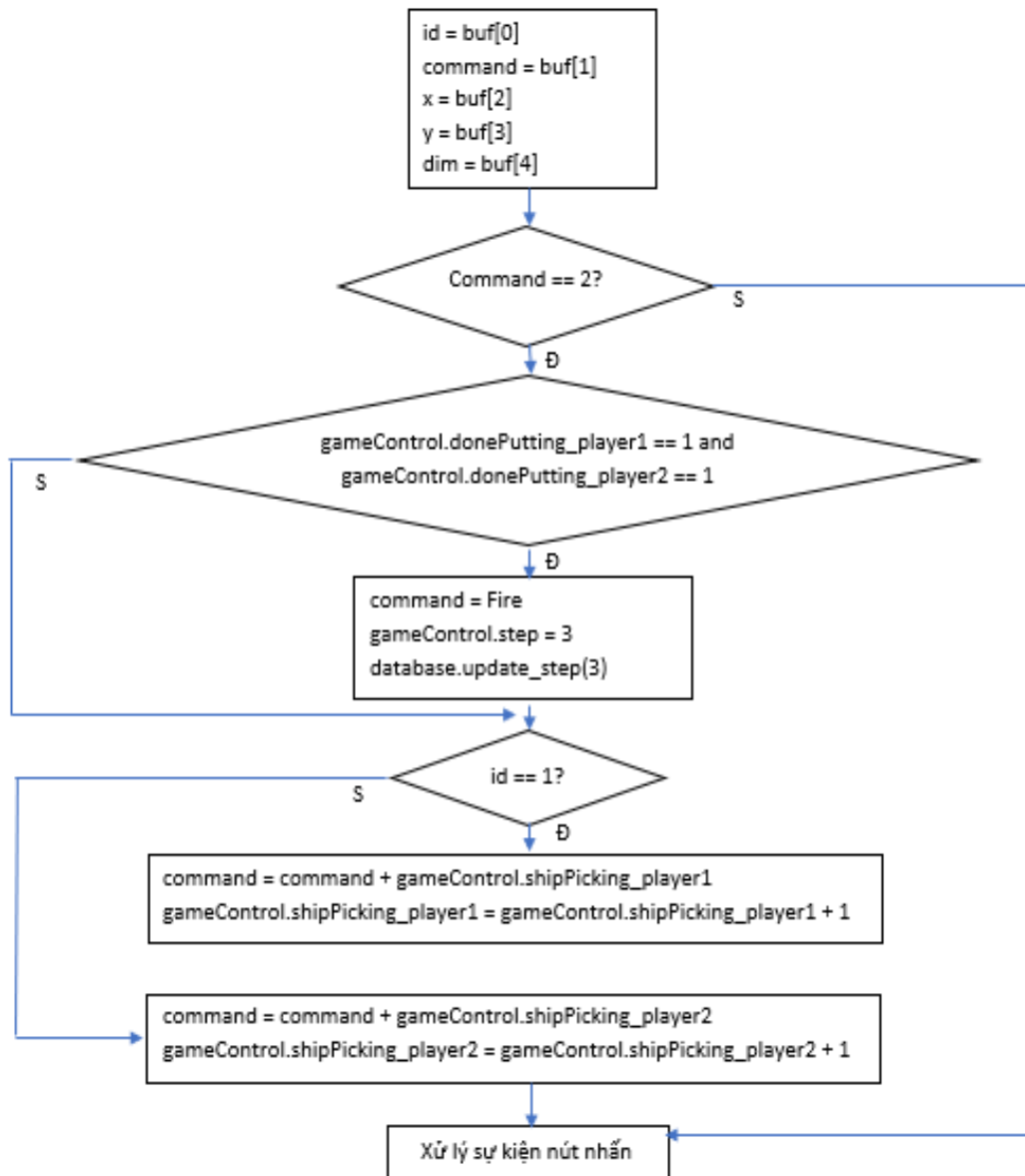
Xử lý sự kiện trên webserver gồm ba bước chính: Nhận đối tượng được gửi từ ESP8266, tách dữ liệu ra khỏi đối tượng nhận được và xử lý các lệnh của sự kiện nút nhấn từ gamepad.

Khi nhận được đối tượng được gửi từ ESP8266, đối tượng sẽ được đưa vào hàm *separate(buf)* để tách dữ liệu ra khỏi đối tượng. Tại đây, chuỗi sẽ được tách ra để lấy dữ liệu *id*, *command*, *x*, *y*, *dim* được gửi từ gamepad và kiểm tra xem tàu đã được đặt hay chưa, nếu tàu chưa đặt thì giá trị *command=2* sẽ là đặt tàu, còn nếu tàu đã được đặt và đang bắn thì giá trị *command=2* sẽ được biến đổi để thực hiện chức năng bắn tàu. Lưu đồ thuật toán của hàm *separate(buf)* được mô tả như trong **Hình 9**. Trong đó, *gameControl.donePutting\_player1 == 1* và *gameControl.donePutting\_player2 == 1* dùng để kiểm tra xem hai người chơi đã hoàn thành việc đặt tàu hay chưa? Nếu đã hoàn thành thì gán *command = Fire*, đổi *step* từ đặt tàu sang bắn và cập nhật lại trạng thái của *step* trong database. Ngược lại thì tăng số tàu đã đặt lên 1 (tối đa mỗi người chơi được đặt 3 chiếc tàu với kích thước từ nhỏ đến lớn).

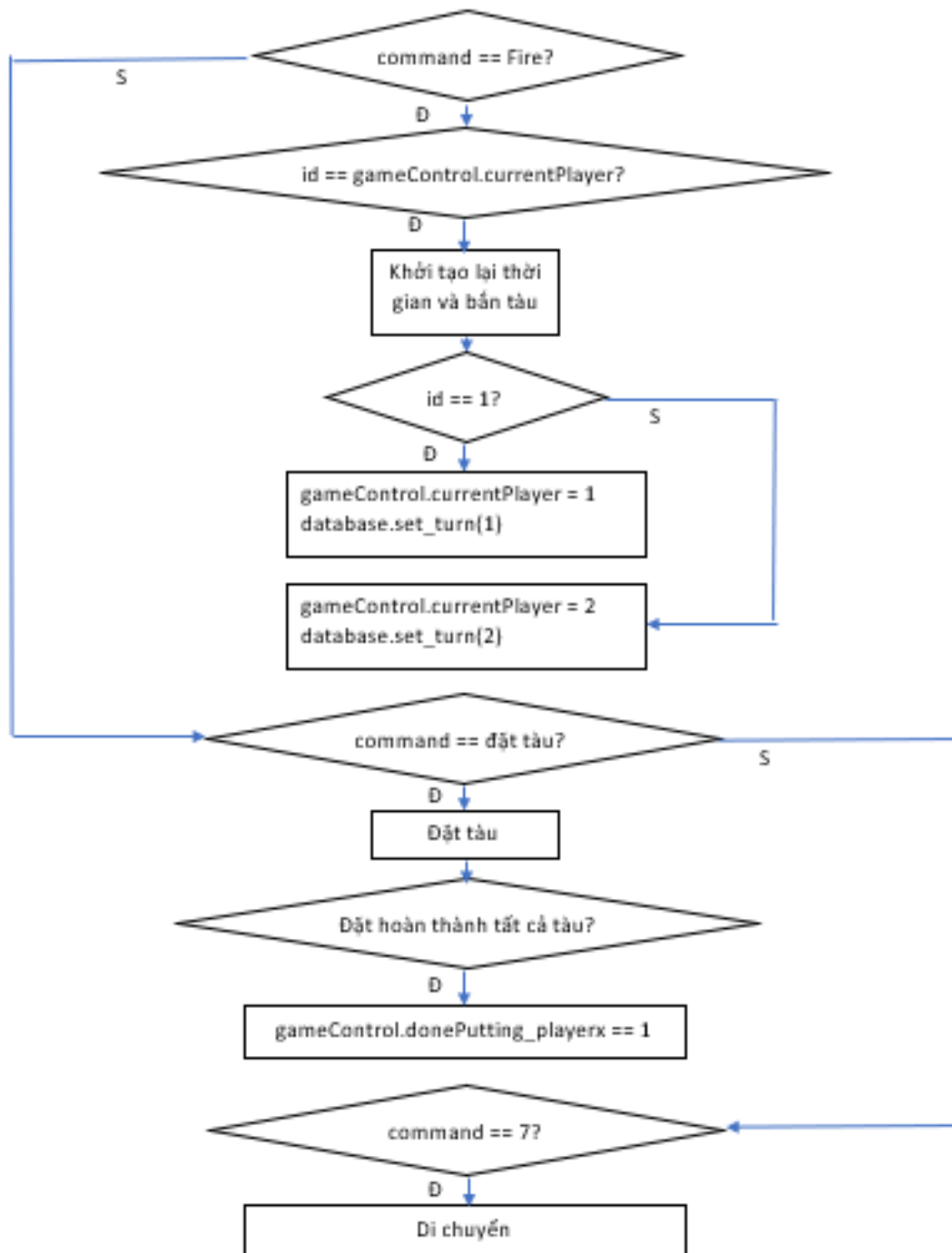
Sau khi tách dữ liệu ra khỏi đối tượng nhận được, hàm *process(buf)* sẽ được gọi để xử lý các lệnh của sự kiện nút nhấn nhận được từ gamepad. Tại đây, hàm sẽ thực hiện các chức năng như di chuyển, bắn tàu, đặt tàu và cập nhật trạng thái vào database. Sau đó, ngôn ngữ php



được sử dụng để truy vấn database và cập nhật trạng thái lên giao diện web (hiển thị tàu được đặt, hiển thị lửa khi bắn trúng tàu, đang di chuyển chọn ô, lượt người chơi...). Lưu đồ thuật toán của hàm ***process(buf)*** được mô tả như trong **Hình 10**. Trong đó, nếu lệnh bắn được thực hiện, chương trình sẽ kiểm tra xem lệnh bắn đó có phải được gửi từ người chơi tại lượt hiện tại hay không bằng cách so sánh biến  $id == gameControl.currentPlayer$  hay không? Nếu đúng lượt chơi thì chương trình sẽ thực hiện bắn và đổi lượt chơi, ngược lại chương trình không làm gì cả. Nếu người chơi đang thực hiện đặt tàu, sau khi đặt xong một chiếc tàu, chương trình sẽ kiểm tra xem tất cả các tàu của hai người chơi được đặt hoàn thành hay chưa? Nếu đã được đặt hoàn thành thì biến  $gameControl.donePutting\_playerx == 1$  (Với x là id của người chơi, mang giá trị 1 hoặc 2). Hoặc nếu người chơi thực hiện lệnh di chuyển, chương trình sẽ lưu tọa độ vị trí mà người chơi đang chọn, cập nhật ô đang chọn ở trạng thái chọn, cập nhật ô trước đó về trạng thái không chọn, sau đó cập nhật dữ liệu trong database và hiển thị lên giao diện web.



**Hình 9.** Lưu đồ thuật toán hàm tách dữ liệu

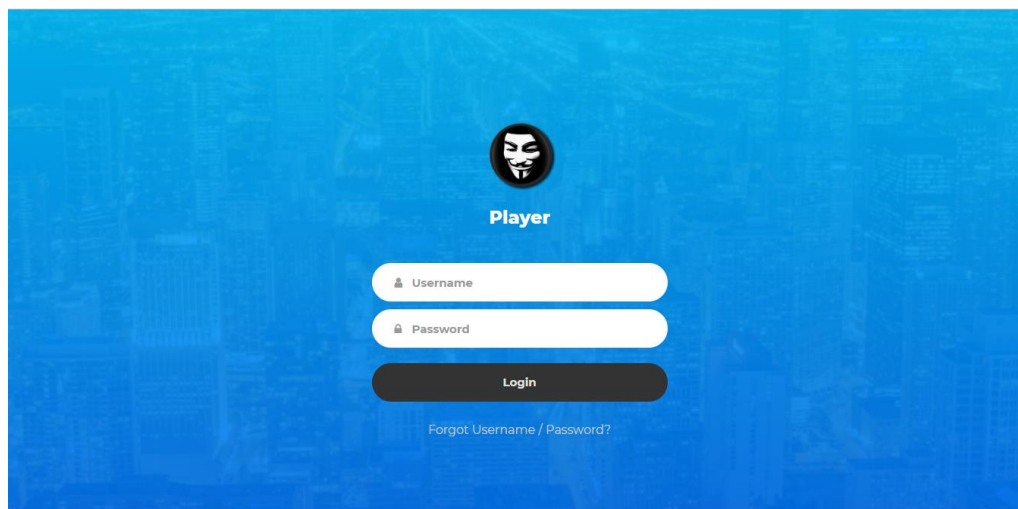


**Hình 10.** Lưu đồ thuật toán hàm xử lý sự kiện nút nhấn

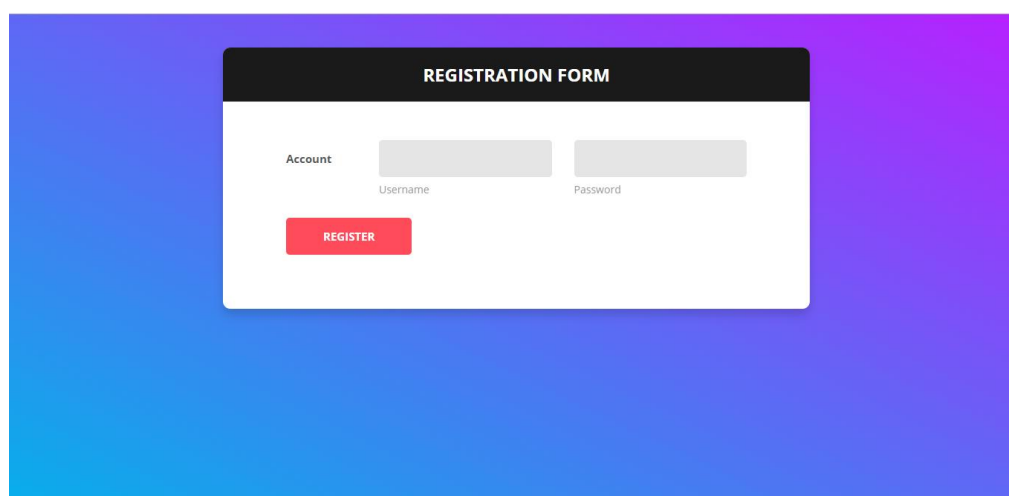
Nếu tàu của người chơi bị bắn trúng, webserver sẽ gửi một giá trị về gamepad với một đích thông báo rằng tàu người chơi đã bị bắn trúng và thực hiện xử lý rung trên gamepad.

#### 2.4. Giao diện web

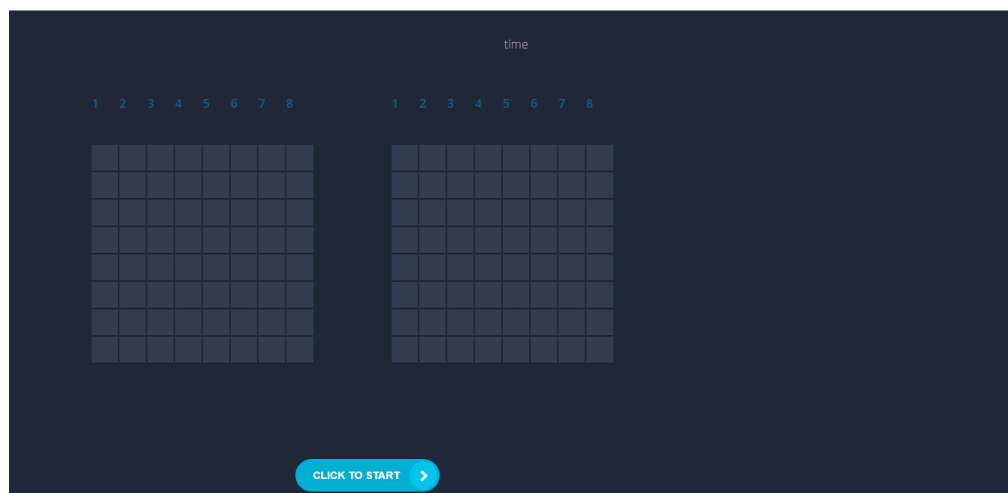
Nhóm sử dụng html và css để tạo ra giao diện web như login, signup, chọn phòng và bắn tàu.



**Hình 11.** Giao diện login



**Hình 12.** Giao diện signup



**Hình 13.** Giao diện trong game

### 3. Kiểm tra

Kết quả chạy chương trình được đính kèm trong link bên dưới:

[https://www.youtube.com/watch?v=iS6Vdb6D0DE&fbclid=IwAR1Qm4R2BcLC8EO2uEP M2HSTJogs8Q4mPJD11b-JxKRE0-5ucFi-ysrU\\_fg](https://www.youtube.com/watch?v=iS6Vdb6D0DE&fbclid=IwAR1Qm4R2BcLC8EO2uEP M2HSTJogs8Q4mPJD11b-JxKRE0-5ucFi-ysrU_fg)

### 4. Kết luận

Nhóm đã xây dựng thành công game battle ship với tín hiệu điều khiển được gửi từ gamepad đến webserver. Tuy nhiên, game battle ship nhóm xây dựng vẫn còn chưa linh động trong việc nhận id gamepad (do id được đặt cứng trong chương trình nhúng xuống gamepad: gamepad1 = id1, gamepad2 = id2), và chỉ giới hạn ở hai người chơi.

Từ game battle ship chỉ được điều khiển bởi gamepad, ta có thể phát triển thêm tính năng điều khiển trên các thiết bị khác để người dùng có thể trải nghiệm game trên các thiết bị di động hoặc trực tiếp trên web. Ngoài ra, ta có thể mở rộng số người chơi, không chỉ giới hạn ở hai người như trong game battle ship nhóm xây dựng.