|  |  |
| --- | --- |
|  | **DEPT OF COMPUTER AND COMMUNICATION ENGINEERING**  **IOT ARCHITECTURE AND PROTOCOLS LAB**  **MMH: ITAL328264** |
| **Group: Nguyễn Thành Trung 20161387**  **Huỳnh Văn Thiện 20161375**  **Đặng Đình Long Vũ 20161398** | |

# HTML (tag, features), CSS (style), Javascript [1]

## HTML

|  |  |
| --- | --- |
| Tag, features  Code  Kết quả | 1. Headings - Tiêu đề |
| Tags, features  Code  Kết quả | 1. Paragraphs - Đoạn văn |
| Tags, features  Code  Kết quả | 1. Styles - Định dạng chữ |
| Tags, features  Code  Kết quả | 1. Formating- định dạng chữ |
| Tags, features  Code  Kết quả | 1. Quotations - Đánh dấu các trích dẫn trong văn bản |
| Tags, features  Code  Kết quả | 1. Colors - định dạng màu nền, màu chữ và viền, có thể sử dụng giá trị màu để biểu thị màu. |
| Tags, features  Code  Kết quả | 1. CSS - kiểm soát màu sắc, phông chữ, kích thước văn bản, khoảng cách giữa các thành phần, cách các thành phần được định vị và bố trí, hình nền hoặc màu nền nào sẽ được sử dụng, các hiển thị khác nhau cho các thiết bị và kích thước màn hình khác nhau |

|  |  |
| --- | --- |
| Tags, features  Code  Kết quả | Links - cho phép người dùng nhấp chuột để truy cập từ trang này sang trang khác |
|  |  |

|  |  |
| --- | --- |
| 9.HTML images | img src = “….jpg”có chức năng tìm đường dẫn đến file hình ảnh và hiện lên web.  alt |
| Code |  |
| Kết quả |
| Khi mà tìm được đường dẫn file ảnh |  |
| Khi mà không tìm được đường dẫn file ảnh. Thì dòng alt sẽ hiện đặc điểm của ảnh. |  |
|  | Hình ảnh đầu tiên sử dụng thuộc tính chiều rộng (được đặt thành 128 pixel), nhưng kiểu trong phần đầu sẽ ghi đè lên thuộc tính đó và đặt chiều rộng thành 100%.  Hình ảnh thứ hai sử dụng thuộc tính style để đặt chiều rộng thành 128 pixel, điều này sẽ không bị ghi đè bởi kiểu trong phần đầu: |
| Code |  |
| Kết quả |  |
| Code sử dụng hình ảnh từ máy chủ khác. |  |
| Kết quả |  |

|  |
| --- |
|  |
| **10. HTML Tables**   1. HTML Tables cho phép các nhà phát triển sắp xếp dữ liệu theo hàng và cột. 2. Cú pháp:  * Mọi thứ ở giữa <td> và </td> là một ô:      * Mỗi hàng của bảng bắt đầu thẻ với <tr> và kết thúc với </tr>:      * Để tạo ô tiêu đề thì dùng <th> thay cho <td>:     c. Code mẫu |
|  |
|  |
|  |
| 1. HTML Lists   <ul> : Định nghĩa một danh sách không có thứ tự  <ol> : Xác định một danh sách có thứ tự  <li> : Xác định một mục danh sách  <dl> : Xác định danh sách mô tả  <dt> : Xác định một thuật ngữ trong danh sách mô tả  <dd> : Mô tả thuật ngữ trong danh sách mô tả  Code :    Kết quả: |
| **12. HTML Blocks & Inline**  Mọi yếu tố trong HTML đều có giá trị hiển thị mặc định, có 2 giá trị hiển thị: Block và Inline.   1. Yếu tố thuộc Block  * Các yếu tố thuộc loại Block luôn luôn bắt đầu với dòng mới, và trình duyệt tự động thêm một số khoảng trống trước và sau các yếu tố đó. * Có 2 loại Block được sử dụng phổ biến: <p> và <div>, <p> định nghĩa đoạn văn trong một tài liệu HTML, <div> định nghĩa một bộ phận hoặc một phần trong một tài liệu HTML     **Kết quả**    Kết quả của các yếu tố thuộc Block sẽ chiếm hết phần chiều rộng của hàng chứa văn Block. |
|  |
|  |
|  |
| 1. Các yếu tố Inline  * Các yếu tố Inline không bắt đầu ở hàng mới * Các yếu tố Inline chỉ lấy phần chiều rộng vừa đủ với văn bản * Yếu tố <span> xác định một vùng chứa nội dung chung là một yếu tố Inline     Kết quả    <p> là yếu tố Block nên chiếm hết phần chiều rộng của website, còn <span> là Inline nên chỉ chiếm một phần vừa đủ để chứa văn bản. |
| 1. HTML Iframes - Dùng để nhúng 1 wed khác vào trong wed của bạn     Cũng có thể sử dụng dòng lệnh:  style="height:500px;width:600px;" thay cho  height="500" width="600"   Kết quả:    Code chỉnh viền cho inframe:    Kết quả. |
| 14. HTML JavaScript  JavaScript làm cho các trang HTML có tính năng động và tương tác hơn  a. Thẻ <script> trong HTML   * Thẻ <script> HTML được dùng để xác định một tập lệnh từ phía người dùng * Để chọn yếu tố HTML, JavaScript thường sử dụng method: document.getElementById() |

|  |
| --- |
| Kết quả in ra là:     * JavaScript có thể đổi nội dung:     Kết quả |
| Kết quả 1:    Kết quả 2:     * JavaScript có thể chuyển styles:     Kết quả: |

|  |
| --- |
| Sau khi nhấn “Click Me”:     * JavaScript có thể thay đổi thuộc tính:       Tạo 2 nút nhấn có 2 element là light(0) và light(1), tạo hàm với light(sw) có biến pic, sw tương đương 0 với 1 khi nhấn nút Light On hay Light Off, dùng element để thay đổi biến pic. |

|  |
| --- |
| Kết quả:    Khi nhấn Light On: |

|  |  |
| --- | --- |
| 15.HTML HEAD | Phần tử <head> là nơi chứa siêu dữ liệu (dữ liệu về dữ liệu)  Phần tử <head> được đặt giữa thẻ <html> và thẻ <body>  Phần tử <title> là bắt buộc và nó xác định tiêu đề của tài liệu. |
|  |  |
| Kết quả |  |

|  |  |
| --- | --- |
|  | Phần tử <style> được sử dụng để xác định thông tin kiểu dáng cho một tài liệu |
| Code <style> |  |
| Kết quả |  |
| Code <meta> |  |
| Kết quả |  |

|  |
| --- |
| 16. HTML Layout  a. HTML Layout Elements  HTML có nhiều yếu tố ngữ nghĩa để xác định những phần khác nhau của trang web    b. Code mẫu |

|  |
| --- |
|  |

|  |
| --- |
| Kết quả: |

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

|  |  |
| --- | --- |
| Tags, features  Code  Kết quả | Symbols - thêm các ký hiệu, ký tự, biểu tượng, các biểu tượng không có trên bàn phím cũng có thể thêm |

|  |
| --- |
| 18. HTML Emojis  Emojis (Các biểu tượng cảm xúc) là các ký tự trong bộ ký tự UTF-8: 😄 😍 💗  Rất nhiều ký tự UTF-8 không thể gõ trên bàn phím, nhưng chúng có thể được hiển thị bằng cách sử dụng một chuỗi số (entity numbers): A là 65, B là 66, C là 67. Ta dùng <meta charset=”UTF-8”> để định nghĩa bộ ký tự.  Ví dụ:    a. Ký tự biểu tượng cảm xúc   * 😄 is 128516 * 😍 is 128525 * 💗 is 128151 |

|  |
| --- |
| Để hiển thị biểu tượng cảm xúc thì cần phải biết mã số của chúng. Như đoạn code trên, mã của mặt cười là 128512 thì ta dùng lện <p>&#128512, hiển thị trái tim là 128151    Danh sách mã: <https://www.w3schools.com/charsets/ref_emoji.asp> |

|  |  |
| --- | --- |
| 19.HTML URL Encode | Trình duyệt web yêu cầu các trang từ máy chủ web bằng cách sử dụng URL.  Bộ định vị tài nguyên thống nhất (URL) được sử dụng để đánh địa chỉ một tài liệu (hoặc dữ liệu khác) trên web. |
| Test chuyển đổi URL |  |
| Kết quả |  |
|  | * **lược đồ** - xác định **loại** dịch vụ Internet (phổ biến nhất là **http hoặc https** ) * **tiền tố** - xác định **tiền tố** tên miền (mặc định cho http là **www** ) * **tên miền - xác định tên miền** Internet (như w3schools.com) * **cổng** - xác định **số cổng** tại máy chủ (mặc định cho http là **80** ) * **path** - xác định **đường dẫn** tại máy chủ (Nếu bỏ qua: thư mục gốc của trang web) * **tên tệp** - xác định tên của tài liệu hoặc tài nguyên   Ví dụ về mã hóa ASCII  Trình duyệt sẽ mã hóa đầu vào, theo bộ ký tự được sử dụng trong trang.  Bộ ký tự mặc định trong HTML5 là UTF-8. |

|  |
| --- |
| 20. HTML Forms  HTML form được sử dụng để thu thập dữ liệu đưa vào bởi người dùng. Dữ liệu vào của người dùng thường được gửi đến server để xử lý.  Element <form> được sử dụng để tạo HTML form cho dữ liệu người dùng    Element <input> là một element form được sử dụng nhiều nhất, có thể được hiển thị bằng nhiều cách, dựa vào thuộc tính type:    <input type = “text”> xác định một trường dữ liệu vào đơn cho dữ liệu vào là văn bản  <label> định nghĩa nhãn cho nhiều form element  <input type=”submit”> định nghĩa nút nhấn để nộp form dữ liệu tới form xử lý  Ví dụ:    Kết quả:    Sau khi nhấn Submit: |

|  |
| --- |
| 1. Graphics(canvas) - vẽ đồ họa trên trang web |

|  |
| --- |
| SVG- vẽ đồ họa dựa trên vecto định dạng |

|  |  |
| --- | --- |
| Canvas |  |

|  |
| --- |
| 22. HTML Video   * Element HTML video được sử dụng để phát video trên trang web * Thuộc tính <controls> thêm vào video để điều khiển video, như phát, dừng, và âm lượng. * Element <source> cho phép phân tích chi tiết files video cái mà trình duyệt lấy từ nó.   Ví dụ: |

|  |  |
| --- | --- |
| 23.HTML Audio | Thuộc tính này controls thêm các điều khiển âm thanh, như phát, tạm dừng và âm lượng.  Phần tử này <source>cho phép bạn chỉ định các tệp âm thanh thay thế mà trình duyệt có thể chọn. Trình duyệt sẽ sử dụng định dạng được nhận dạng đầu tiên.  Văn bản giữa thẻ <audio>và </audio>sẽ chỉ được hiển thị trong các trình duyệt không hỗ trợ <audio>phần tử này. |
| Code |  |
| Kết quả |  |
| Code (chèn nhạc tự phát nhưng tắt tiếng) |  |
| Kết quả |  |

|  |
| --- |
| 1. HTML Plug-ins   Plug-in là các chương trình máy tính mở rộng chức năng tiêu chuẩn của trình duyệt .  Element <object> định nghĩa các object được nhúng vào cùng với tài liệu HTML.  Ví dụ:    Kết quả: |

|  |
| --- |
| - Hoặc là hình ảnh:    Kết quả: |

|  |  |
| --- | --- |
| 25.HTML Youtube | Để phát video lên trang web. |
| Code |  |
| Kết quả |  |
|  | Ngoài ra còn có thể làm cho video tự phát và tắt tiếng bằng đoạn code  src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&mute=1"> </iframe>  Thêm loop=1để video của bạn lặp lại mãi mãi.  Giá trị 0 (mặc định): Video sẽ chỉ phát một lần.  Giá trị 1: Video sẽ lặp lại (mãi mãi).  iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1"> </iframe> |

|  |
| --- |
| 26. HTML Geolocation   * HTML Geolocation API được sử dụng để xác định vị trí của người dùng * Điều này có thể ảnh hưởng đến quyền riêng tư nên cần sự chấp thuận từ người dùng * Method getCurrentPosition() được sử dụng để quay lại vị trí của người dùng   Ví dụ:      Kết quả:    Sau khi nhấn Try It: |
| Kiểm tra xem định vị địa lý có được hỗ trợ, nếu có thì chạy method getCurrentPosition(), nếu không thì hiện thông báo cho người dùng. Nếu getCurrentPosition() chạy thành công, nó trả về một đối tượng tọa độ cho hàm được chỉ định trong tham số showPosition. Hàm showPosition xuất ra Vĩ độ và Kinh độ |

## CSS

|  |
| --- |
| 1. **CCS Color**   CSS hỗ trợ hơn 140 tên màu, giá trị HEX, RGB, giá trị RGBA, giá trị HSL, giá trị HSLA và độ mờ.  - **Giá trị màu RGBA** được chỉ định bằng: rgba(red, green, blue, alpha). Tham số alpha là một số nằm trong khoảng từ 0,0 (hoàn toàn trong suốt) đến 1,0 (hoàn toàn mờ).  +Cú pháp sử dụng: rgba(red, green, blue, alpha) tham số alpha là một số nằm trong khoảng từ 0,0 (hoàn toàn trong suốt) đến 1,0 (hoàn toàn mờ).  Vd:    - **Màu HSL:** HSL là viết tắt của Hue, Saturation và Lightness.  Giá trị màu HSL được chỉ định bằng: hsl(màu sắc, độ bão hòa, độ sáng).   * Hue là một mức độ trên bánh xe màu (từ 0 đến 360): * 0 (hoặc 360) là màu đỏ * 120 là màu xanh lá cây * 240 là màu xanh * Độ bão hòa là giá trị phần trăm: 100% là màu đầy đủ. * Độ nhẹ cũng là một phần trăm; 0% là màu tối (đen) và 100% là màu trắng. |

|  |
| --- |
| - Màu sắc HSLA  Giá trị màu HSLA là phần mở rộng của giá trị màu HSL với kênh alpha - kênh này chỉ định độ mờ cho màu.  Giá trị màu HSLA được chỉ định bằng:hsla(màu sắc, độ bão hòa, độ sáng, alpha), trong đó tham số alpha xác định độ mờ. Tham số alpha là một số nằm trong khoảng từ 0,0 đến 1,0 .    - Opacity  Thuộc tính CSS opacity đặt độ mờ cho toàn bộ phần tử (cả màu nền và văn bản sẽ mờ/trong suốt).  Giá opacity trị thuộc tính phải là một số trong khoảng từ 0,0 đến 1,0 . |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2. Background  Color : Lựa chọn màu cho hình nền web. Ở đây là màu xanh lá | | |  | |
| Kết quả : | | |  |
| Có thể tạo nền màu cho mỗi dòng mà mình muốn khi set up cho liên kết của dòng đấy nền màu mà mình muốn chọn. | | |  |
|  | |  |
|  | |  |
| Opacity là mức độ nhạt của chữ 1 là mức cao nhất chữ rõ bình thường. khi < 1 thì càng về o mức nhạt của chữ sẽ tăng lên. | | <!DOCTYPE html>  <html>  <head>  <style>  div {    background-color: green;  }  div.first {      opacity: 0.1;  }  div.second {    opacity: 0.3;  }  div.third {    opacity: 0.6;  }  </style>  </head>  <body>  <h1>Transparent Boxes</h1>  <p>When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:</p>  <div class="first">    <h1>Mức nhạt nhất</h1>  </div>  <div class="second">    <h1>Đậm dần rồi</h1>  </div>  <div class="third">    <h1>Còn mờ</h1>  </div>  <div>    <h1>Rõ nhất rồi (default)</h1>  </div>  </body>  </html> |
| Kết quả | |  |
| Image : Tạo hình nền cho giao diện web với 1 hình ảnh mà mình chọn. | | <!DOCTYPE html>  <html>  <head>  <style>  body {    background-image: url("Hinh-nen-dien-thoai-bau-troi-1.jpg");  }  </style>  </head>  <body>  <h1>Hello World!</h1>  <p>This page has an image as the background!</p>  </body>  </html> |
| Kết quả | |  |
| Tạo hình nền cho 1 dòng mà mình muốn chỉ định. | | <!DOCTYPE html>  <html>  <head>  <style>  p {    background-image: url("Hinh-nen-dien-thoai-bau-troi-1.jpg");  }  </style>  </head>  <body>  <h1>Hello World!</h1>  <p>Chỉ có dòng này là có nền nè </p>  </body>  </html> | |
| Kết quả | |  |
| Hình nền lặp lại theo chiều ngang và đối với chiều dọc thì không được. | |  |
| Kết quả: | |  |

|  |
| --- |
| **3 Border**  - CSS Borders  Thuộc tính border-style chỉ định loại đường viền sẽ hiển thị.  Các giá trị sau được cho phép:   * dotted- Xác định đường viền chấm * dashed- Xác định đường viền nét đứt * solid- Xác định một đường viền vững chắc * double- Xác định đường viền kép * groove- Xác định đường viền có rãnh 3D. Hiệu ứng phụ thuộc vào giá trị màu viền * ridge- Xác định đường viền có gờ 3D. Hiệu ứng phụ thuộc vào giá trị màu viền * inset- Xác định đường viền lồng vào 3D. Hiệu ứng phụ thuộc vào giá trị màu viền * outset- Xác định đường viền đầu 3D. Hiệu ứng phụ thuộc vào giá trị màu viền * none- Xác định không có đường viền * hidden- Xác định đường viền ẩn   Thuộc tính border-style có thể có từ một đến bốn giá trị (cho viền trên, viền phải, viền dưới và viền trái).  \    - CSS border-image Property  Thuộc tính CSS border-image cho phép bạn chỉ định một hình ảnh sẽ được sử dụng thay vì đường viền thông thường xung quanh một phần tử.  Gồm có ba phần:   1. Hình ảnh dùng làm đường viền 2. Cắt ảnh ở đâu 3. Xác định xem các phần ở giữa nên được lặp lại hay kéo dài |

|  |
| --- |
| - CSS border-image - Different Slice Values  Các giá trị lát cắt khác nhau sẽ thay đổi hoàn toàn hình thức của đường viền: |

|  |
| --- |
|  |

|  |
| --- |
| 1. **Margin**   - Margin - Individual Sides  CSS có các thuộc tính để xác định lề cho mỗi cạnh của một phần tử:   * margin-top * margin-right * margin-bottom * margin-left       - Margin - Shorthand Property  Để rút ngắn mã, có thể chỉ định tất cả các thuộc tính lề trong một thuộc tính.  Thuộc tính margin là thuộc tính viết tắt cho các thuộc tính lề riêng lẻ sau:   * margin-top * margin-right * margin-bottom * margin-left   Vì vậy, đây là cách nó hoạt động:  Nếu margin thuộc tính có bốn giá trị:   * **lề: 35px 60px 75px 90px;**   + lề trên là 35px   + lề phải là 60px   + lề dưới là 75px   + lề trái là 90px       Nếu thuộc tính margin có ba giá trị:   * **lề: 15px 40px 85px;**   + lề trên là 15px   + lề phải và trái là 40px   + lề dưới là 85px       - The auto Value  Có thể đặt thuộc tính auto để căn lề giữa phần tử theo chiều ngang trong vùng chứa của nó.  Phần tử sau đó sẽ chiếm chiều rộng được chỉ định và khoảng trống còn lại sẽ được chia đều cho lề trái và lề phải.      - The inherit Value |

|  |
| --- |
| 1. CSS Padding   - **Padding**  được sử dụng để tạo không gian xung quanh nội dung của phần tử, bên trong bất kỳ đường viền được xác định nào.      - Padding - Individual Sides  CSS có các thuộc tính để chỉ định phần đệm cho mỗi bên của một phần tử:   * padding-top * padding-right * padding-bottom * padding-left   Tất cả các thuộc tính đệm có thể có các giá trị sau:   * chiều dài - chỉ định phần đệm bằng px, pt, cm, v.v. * % - chỉ định phần đệm theo % chiều rộng của phần tử chứa * kế thừa - chỉ định rằng phần đệm phải được kế thừa từ phần tử cha       - Padding - Shorthand Property  Để rút ngắn mã, có thể chỉ định tất cả các thuộc tính đệm trong một thuộc tính.  Thuộc tính padding là thuộc tính tốc ký cho các thuộc tính đệm riêng lẻ sau:   * padding-top * padding-right * padding-bottom * padding-left   Vì vậy, đây là cách nó hoạt động:  Nếu thuộc tính padding có bốn giá trị:   * **phần đệm: 25px 50px 75px 100px;**   + phần đệm trên cùng là 25px   + phần đệm bên phải là 50px   + phần đệm phía dưới là 75px   + phần đệm bên trái là 100px       - Padding and Element Width  Thuộc tính CSS width chỉ định độ rộng của vùng nội dung của phần tử. Vùng nội dung là phần bên trong phần đệm, đường viền và lề của một phần tử . |

|  |
| --- |
| 1. **CSS HEIGHT**   CSS height là thuộc tính được sử dụng để đặt chiều cao của một phần tử.      Các thuộc tính heightvà width có thể có các giá trị sau:   * auto- Đây là mặc định. Trình duyệt tính toán chiều cao và chiều rộng * length- Xác định chiều cao/chiều rộng tính bằng px, cm, v.v. * %- Xác định chiều cao/chiều rộng theo phần trăm của khối chứa * initial- Đặt chiều cao/chiều rộng về giá trị mặc định * inherit- Chiều cao/chiều rộng sẽ được kế thừa từ giá trị gốc của nó       - Setting max-width  Thuộc tính max-width được sử dụng để đặt chiều rộng tối đa của một phần tử.   max-width có thể chỉ định bằng các giá trị độ dài , như px, cm, v.v. hoặc theo phần trăm (%) của khối chứa hoặc được đặt thành không (đây là mặc định. Có nghĩa là không có chiều rộng tối đa).  Nếu vì lý do nào đó bạn sử dụng cả thuộc tính width và max-width trên cùng một phần tử và giá trị của thuộc tính width lớn hơn thuộc tính max-width; thuộc tính max-width sẽ được sử dụng (và thuộc tính width sẽ bị bỏ qua). |

|  |
| --- |
| 1. **Outline**   là một đường được vẽ bên ngoài đường viền của phần tử.      CSS có các thuộc tính phác thảo sau:   * outline-style * outline-color * outline-width * outline-offset * Outline   Kiểu phác thảo CSS  Thuộc tính outline-style chỉ định kiểu của đường viền và có thể có một trong các giá trị sau:   * dotted- Xác định một đường viền chấm * dashed- Xác định đường viền nét đứt * solid- Xác định một đường viền liền * double- Xác định một đường viền kép * groove- Xác định đường viền có rãnh 3D * ridge- Xác định đường viền có gờ 3D * inset- Xác định một phác thảo nội dung 3D * outset- Xác định một phác thảo đầu 3D * none- Xác định không có phác thảo * hidden- Xác định một phác thảo ẩn |

|  |
| --- |
| **8. CSS Text**   1. Text Color  * Color dùng để thiết lập màu cho đoạn văn bản. Giá trị của màu có thể ghi theo tên màu, giá trị HEX của màu, giá trị RGB. * Ví dụ:      * Kết quả:      1. Background Color  * “background-color” dùng để đặt màu cho nền của văn bản * Ví dụ:        * Kết quả: |

|  |  |
| --- | --- |
| 9.CSS Font | Tùy chỉnh phông chữ sử dụng trong web. |
| Mỗi phần p1 p2 p3 đều det uop kiểu chữ khác nhau để nhận dạng. |  |
| Kết quả : |  |
| Các kiểu chữ thường dùng cho web. |  |
| 5 kiểu dự phòng thường  Dung khi các kiểu chữ khác không phù hợp sử dụng trong web |  |
| Kiểu chữ của wed |  |
| Kết quả : |  |
| Kích thước, độ dài của chữ |  |
| Kết quả : |  |
| Code biến thể của phông chữ (trong vd là in hoa chữ nhỏ) |  | |
| Kết quả |  |
| Tùy chỉnh kích thước của chữ trong web |  |
| Kết quả |  |
| Thay vì ghi trực tiếp pixels thì ta sử dụng em (em=pixels/16). |  |
| Kết quả : |  |

|  |
| --- |
| **10. CSS Icons**   * Các icon có thể dễ dàng được thêm vào trang HTML bằng cách sử dụng thư viện icon.   a. Font Awesome Icons   * Để sử dụng Font Awesome Icons, vào fontawesome.com, đăng nhập và lấy mã (code) dán vào trong phần <head> trên trang HTML:   <script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>   * Ví dụ:      * Kết quả:     b. Bootstrap Icons   * Dán đường link này vào phần <head> của trang HTML và thực hiện dán các thẻ như Font Awesome:   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">   * Ví dụ:        * Kết quả:     c. Google Icons   * Để sử dụng Google Icon, thêm đường link bên dưới vào phần <head> của trang HTML:   <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">   * Ví dụ:      * Kết quả: |

|  |  |
| --- | --- |
| 11.CSS Lists  Tạo các danh sách menu torng giao diện web |  |
| Kết quả : |  |
| Tải file.gif dung để làm mục đánh dấu đầu hàng cho mỗi câu trong web. |  |
| Kết quả : |  | |
| Tạo background cho giao diện menu thêm sinh động. |  | |
|  |  | |
| Kết quả : |  | |

|  |
| --- |
| **12. CSS Tables**   * Bảng HTML có thể được cải thiện rất nhiều với CSS * Ví dụ:              * Kết quả:       a. Table Borders   * Ví dụ:        * Kết quả:     b. Full-Width Table   * Ví dụ:        * Kết quả: |

|  |  |
| --- | --- |
| 13.CSS Navigation Bar  Tạo các mục để dẫn đến đường link mà người tạo web thiết kế. |  |
| Kết quả : |  |
| Kết quả khi ấn vào link “Trang hướng dẫn làm cái này nè “ |  |

|  |  |
| --- | --- |
| Tạo khung cho mỗi đường dẫn thành 1 hình hộp và thi di chuyển chuột qua thì hộp đấy đổi màu |  |
|  |  |

|  |  |
| --- | --- |
| Kết quả giao diện: |  |
| Kết quả khi ấn và HTML |  |
| Kết quả khi ấn vào YOUTUBE |  |
|  |  |

|  |  |
| --- | --- |
| Tạo thêm 1 khung riêng biệt chứa văn bản cho giao diện và có thể văn bản dài vượt qua màn hình và để xem thêm phần bên dưới bị che mất bằng cách lăn chuột xuống |  |
| Kết quả : |  |

|  |  |
| --- | --- |
| Code dùng cho việc tạo thanh điều hướng theo phương ngang. |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Kết quả : |  |

|  |
| --- |
| **14. CSS Image Gallery**   * Ví dụ:              * Kết quả: |

|  |  |
| --- | --- |
| 15.CSS Website Playout | Để thiết kế bố cục của trang web |
| Dòng code thiết kết khung và kiểu chữ tiêu đề của web.  Thiết kế phần thanh điều hướng cho trang web theo hướng ngang. | <!DOCTYPE html>  <html lang="en">  <head>  <title>CSS Website Layout</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  \* {    box-sizing: border-box;  }  body {    margin: 0;  }  /\* Style the header \*/  .header {    background-color: #92dec6;    padding: 20px;    text-align: center;  }  /\* Style the top navigation bar \*/  .topnav {    overflow: hidden;    background-color: #333;  } |

|  |  |
| --- | --- |
| Tạo kiểu cho liên kết topnay.  Thay đổi màu khi di chuyển chuột  Chia 3 cột không bằng nhau  Cột trái và phải chiếm 25% diện tích khung.  Cột giữa chiếm 50% diện tích khung.  Xóa các floats sau các cột  Bố cục 3 cột chồng sát nhau.  Tạo kiểu cho footer. | /\* Style the topnav links \*/  .topnav a {    float: left;    display: block;    color: #f2f2f2;    text-align: center;    padding: 14px 16px;    text-decoration: none;  }  /\* Change color on hover \*/  .topnav a:hover {    background-color: #ddd;    color: black;  }  /\* Create three unequal columns that floats next to each other \*/  .column {    float: left;    padding: 10px;  }  /\* Left and right column \*/  .column.side {    width: 25%;  }  /\* Middle column \*/  .column.middle {    width: 50%;  }  /\* Clear floats after the columns \*/  .row::after {    content: "";    display: table;    clear: both;  }  /\* Responsive layout - makes the three columns stack on top of each other instead of next to each other \*/  @media screen and (max-width: 600px) {    .column.side, .column.middle {      width: 100%;    }  }  /\* Style the footer \*/  .footer {    background-color: #f1f1f1;    padding: 10px;    text-align: center;  }  </style>  </head> |

|  |  |
| --- | --- |
| Nội dung của phần tiêu đề và nội dung chính. | <body>  <div class="header">    <h1>TRƯỜNG THCS NGUYỄN VĂN A</h1>    <p>Ngôi trường dành cho tầng lớp quý tộc</p>  </div>  <div class="topnav">    <a href="#">Lớp 1</a>    <a href="#">Lớp 2</a>    <a href="#">Lớp 3</a>  </div>  <div class="row">    <div class="column side">      <h2>Buổi sáng</h2>      <p>Tập thể dục buổi sáng </p>      <p>Vào tiết 1 đến tiết 5 sau đó ăn trưa</p>    </div>    <div class="column middle">      <h2>Buổi chiều</h2>      <p>Học tiếp tục 5 tiết tiếp theo theo thời khóa biểu của mỗi lớp và sau đó sinh hoạt cuối ngày cùng với thầy cô trong lớp.</p>      <p>Học xong thì xõa thôi hehe . Nhưng nhớ về nhà phụ giúp gia đình đừng có mà nằm ường ra làm biếng</p>    </div>    <div class="column side">      <h2>Học phí</h2>      <p>Mỗi năm thu 60 tỉ vì được quay mv với messi :))))</p>    </div>  </div>  <div class="footer">    <p>Trích nguồn nguyên văn từ ai đó </p>  </div>  </body>  </html> |
| Kết quả : |  |

|  |
| --- |
| **16. CSS Rounded Corners**   * Lệnh border-radius xác định độ cong của góc của element * Ví dụ:          * Kết quả: |

|  |
| --- |
| 1. **CSS BACKGROUNDS**   - CSS Multiple Backgrounds  +CSS cho phép bạn thêm nhiều hình nền cho một phần tử thông qua background-image.  +Các hình nền khác nhau được phân tách bằng dấu phẩy và các hình ảnh được xếp chồng lên nhau, trong đó hình ảnh đầu tiên ở gần người xem nhất.    - CSS Background Size  Thuộc tính CSS background-size cho phép bạn chỉ định kích thước của hình nền.  Kích thước có thể được chỉ định theo độ dài, tỷ lệ phần trăm hoặc bằng cách sử dụng một trong hai từ khóa: contain or cover.        - Define Sizes of Multiple Background Images  Thuộc tính background-size cũng chấp nhận nhiều giá trị cho kích thước nền (sử dụng danh sách được phân tách bằng dấu phẩy) khi làm việc với nhiều nền.      - Full Size Background Image  Sử dụng hình nền bao phủ trên một cửa sổ trình duyệt.      - Hero Image  Bạn có thể sử dụng các thuộc tính nền khác nhau trên <div> để tạo hình ảnh chính (hình ảnh lớn có văn bản) và đặt nó ở nơi bạn muốn.        - CSS background-origin Property  Thuộc tính CSS background-origin chỉ định vị trí của hình nền.  Thuộc tính này nhận ba giá trị khác nhau:   * hộp viền - hình nền bắt đầu từ góc trên bên trái của đường viền * hộp đệm - (mặc định) hình nền bắt đầu từ góc trên bên trái của cạnh đệm * hộp nội dung - hình nền bắt đầu từ góc trên bên trái của nội dung         - CSS background-clip Property  Thuộc tính CSS background-clip chỉ định vùng vẽ của nền.  Thuộc tính này nhận ba giá trị khác nhau:   * hộp viền - (mặc định) nền được vẽ ở cạnh ngoài của đường viền * hộp đệm - nền được vẽ vào cạnh ngoài của phần đệm * hộp nội dung - nền được vẽ trong hộp nội dung |

|  |
| --- |
| **18. CSS Buttons**  a. Button Colors   * Ví dụ:          * Kết quả:     b. Button Sizes   * Ví dụ:        * Kết quả:     c. Hoverable Buttons   * Nút nhấn sẽ thay đổi khi di chuyển chuột vào * Ví dụ:               d. Disabled Buttons   * Dùng opacity để làm mờ tạo hiệu ứng không cho phép * Dùng cursor với giá trị not-allowed để hiển thị không cho phép khi di chuyển chuột vào * Ví dụ:        * Kết quả:     e. Button Width   * Dùng Width để thiết lập độ rộng của nút nhấn * Ví dụ:        * Kết quả:     f. Button Groups   * Loại bỏ lề và thêm float:left vào mỗi nút nhấn để tạo ra một nhóm nút nhấn * Ví dụ:        * Kết quả: |

|  |
| --- |
| 19. CSS The object-position  Thuộc tính CSS object-positionđược sử dụng để chỉ định cách định vị <img> hoặc <video> trong vùng chứa của nó.  sử dụng object-fit: cover;để giữ tỷ lệ khung hình và điền vào kích thước nhất định.      Giả sử phần hình ảnh được hiển thị không được định vị như chúng ta mong muốn. Để định vị hình ảnh, chúng ta sẽ sử dụng thuộc tính object-position. |

|  |
| --- |
| **20. CSS Multiple Columns**   * Khi văn bản quá dài thì sẽ được hiển thị theo nhiều cột giống như các bài báo   a. Tạo Multiple Columns   * Dùng lệnh column-count để xác định số cột mà văn bản sẽ chia * Ví dụ:        * Kết quả:     b. Tạo khoảng cách giữa các cột   * Dùng column-gap để xác định khoảng cách giữa các cột * Ví dụ:        * Kết quả:     c. CSS Column Rules   * Column-rule tạo ra đường ngăn cách giữa các cột * Ví dụ:        * Kết quả:     d. Column Width   * Column-width xác định độ rộng mỗi cột * Ví dụ:        * Kết quả: |

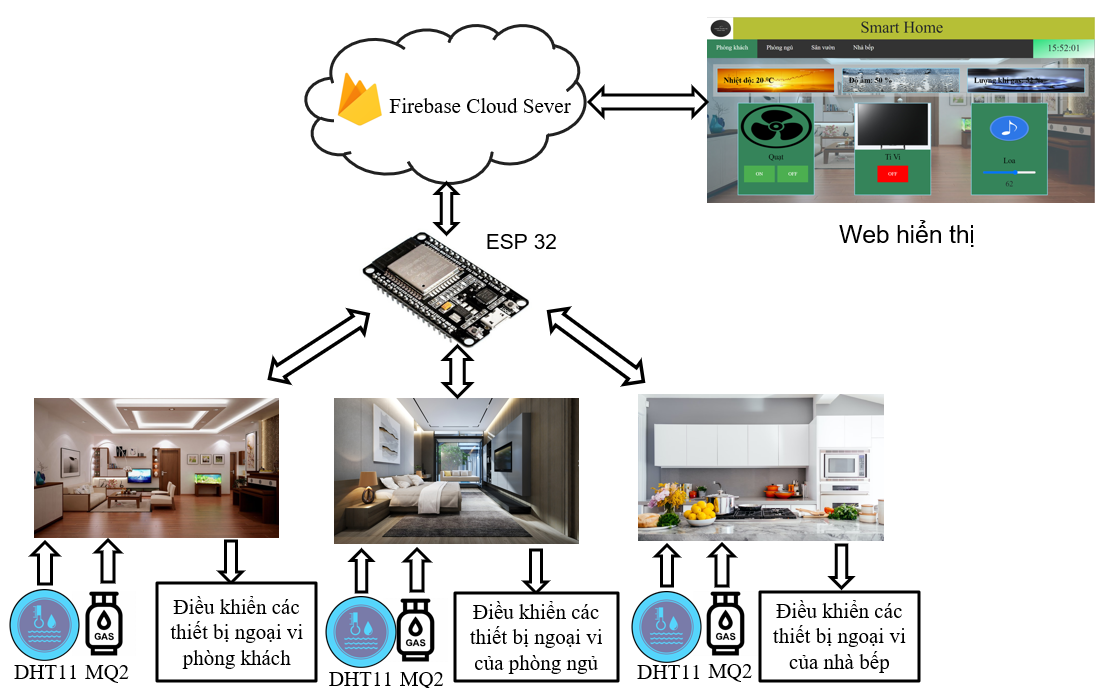
# THIẾT KẾ HỆ THỐNG

## 1. SƠ ĐỒ KHỐI CỦA HỆ THỐNG

### 1.1. Yêu cầu của hệ thống:

* Điều khiển bật tắt từ xa các thiết bị trong mỗi phòng của ngôi nhà như Tivi, quạt, đèn, lò vi sóng,… nhờ giao tiếp với Vi điều khiển ESP 32 và truyền tính hiệu lên nền tảng firebase và wed để quản lý ngôi nhà của mình.
* Đo đạt các thông số của môi trường xung quang căn nhà như : Nhiệt độ, Độ ẩm và Lượng khí Gas thông qua vi điều khiển ESP 32 để mà cập nhật và truyền tải các thông số lên nền tảng firebase và cũng như hiện lên wed để cảnh báo và đề phòng cho ngôi nhà tránh nguy cơ như : cháy nổ, ngộ độc khí,....
* Thiết kế giao diện wed sao cho người dùng có thể sử dụng một cách dễ dàng và bên cạnh đó bố cục và các số liệu cũng phải thật chính xác, hợp lý và ưu nhìn.

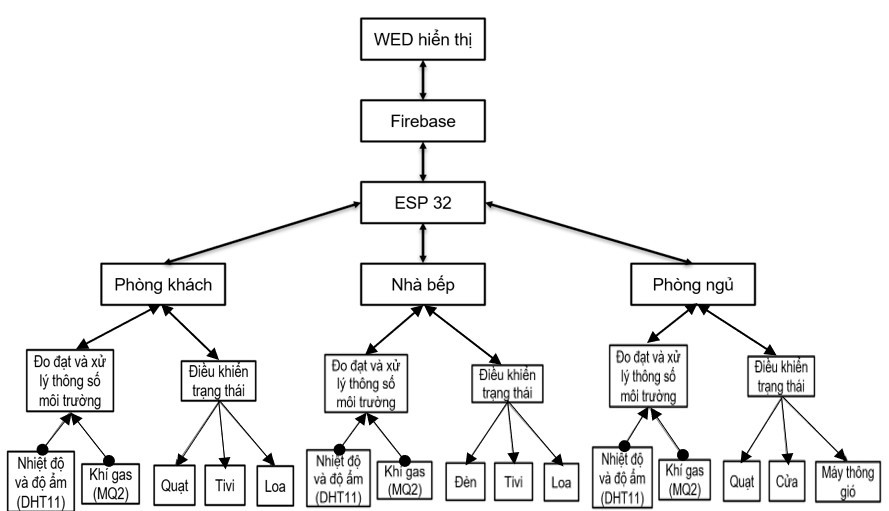
### 1.2. Sơ đồ phác thảo của hệ thống :



Hình 1 Phác thảo sơ đồ của hệ thống ngôi nhà thông minh

Dựa vào hình 1 cho ta thấy được mỗi căn phòng điều được lắp cảm biến DHT11 và MQ2 dùng để đo nhiệt độ, độ ẩm và khí gas của môi trường gửi cho ESP32. Bên cạnh đó thông qua bộ vi điều khiển ( ESP32 ) để gửi các tín hiệu dùng để điều khiển các thiết bị ngoại vi. Toàn bộ các thông tin như: Nhiệt độ, độ ẩm, khí gas và trạng thái của các thiết bị của mỗi phòng sẽ được gửi về vi điều khiển để xử lý và gửi về google firebase để qua đó chuyển đổi để hiện các thông số trên lên wed của người dùng quan sát.

### 1.3. Sơ đồ cây của hệ thống :



Hình 2 Sơ đồ cây chi tiết của hệ thống ngôi nhà thông minh.

Dựa vào hình 2 cho chúng ta thấy được chi tiết của mô hình nhà thông minh với việc lắp đặt các cảm biến gửi về wed để hiển thị và các thiết bị trong mỗi phòng được điều khiển từ xa thông qua vi điều khiển để gửi lên firebase và hiển thị lên wed cho người dùng dễ dàng quan sát và thao tác điều khiển từ xa.

### 1.4. Sơ đồ khối của hệ thống:

KHỐI NGUỒN

TRANG WEB

ĐÁM MÂY (GOOGLE FIREBASE)

KHỐI CẢM BIẾN (DHT11, MQ2)

KHỐI HIỂN THỊ LCD (LOA, MÁY THÔNG GIÓ)

KHỐI THIẾT BỊ NGOẠI VI (LED)

KHỐI NÚT NHẤN

KHỐI XỬ LÝ TRUNG TÂM

**Sơ đồ khối của hệ thống nhà thông minh.**

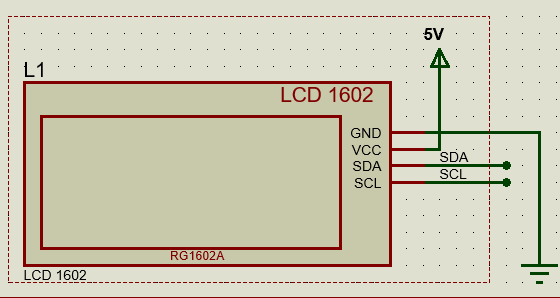
* **Chức năng của các khối**
* **Khối xử lý trung tâm :** Khối xử lý trung tâm đóng vai trò như bộ não của hệ thống, điều khiển và quản lý các hoạt động của toàn bộ hệ thống. Khối này liên lạc với tất cả các khối khác trong hệ thống để thực hiện các yêu cầu được đưa ra bởi hệ thống.
* **Khối nút nhấn:** Có thể điều chỉnh trạng thái on hoặc off của các thiết bị trong các phòng khác nhau để gửi lên Firebase cũng như là thể hiện lên wed.
* **Khối hiển thị**: Để hiển thị các thông tin như trạng thái của các thiết bị và thông số nhiệt độ , độ ẩm và khí gas từ những cảm biến gửi lên vi điều khiển (LCD 16x2).
* **Khối nguồn:** Để cung cấp nguồn cho toàn bộ mạch, ta có thể sử dụng nguồn cấp DC từ 7.5V đến 12V.
* **Khối cảm biến :** Dùng để đo đạc và xử lý các thông số nhiệt độ, độ ẩm và khí gas của môi trường xung quang phòng.
* **Khối thiết bị ngoại vi :** Thay thế các thiết bị trong nhà bằng đèn led để tiện thực hiện mô hình nhà thông minh.
* **Trang web :** Là giao diện để phục vụ cho người dùng dễ dàng quan xác và điều chỉnh các thiết bị ngoại vi từ xa.
* **ĐÁM MÂY (GOOGLE FIREBASE) :** là nền tảng phục vụ việc truyền và nhận các thông tin từ vi điều khiển để cập nhật lên trang wed.

## 2.THIẾT KẾ HỆ THỐNG PHẦN CỨNG.

### 2.1. Khối hiển thị.

* **Sơ đồ nguyên lý**

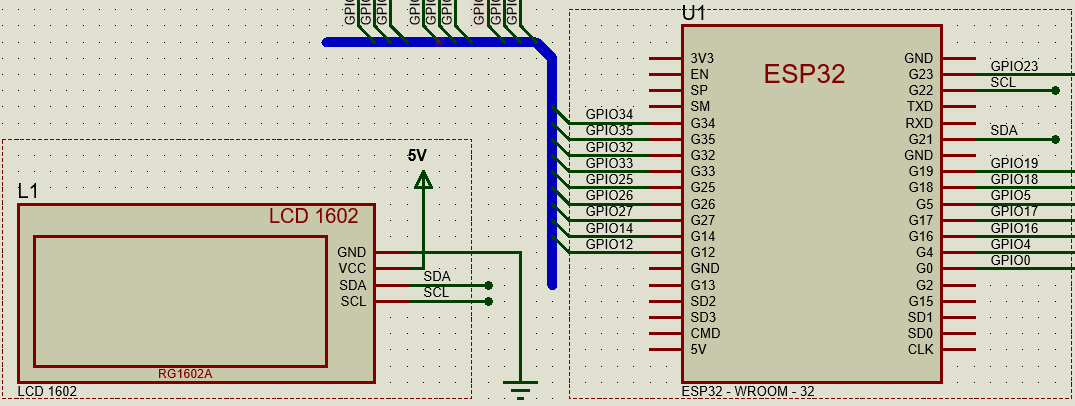
**Khối hiển thị**: Để hiển thị các thông tin như nhiệt độ, độ ẩm và lượng khí gas có trong mỗi phòng. Màn hình sử dụng LCD 16x2. Các thông tin này đã được xử lý từ khối xử lý trung tâm và được truyền tới màn hình LCD để hiển thị.



Hình 3 Sơ đồ nguyên lý khối hiển thị LCD 16x2

Nhìn vào hình 3 cho ta thấy khối hiển thị được sử dụng là LCD 16x2 giao tiếp qua I2C sử dụng với nguồn 5V và chân SDA và SCL để truyền và nhận dữ liệu từ vi điều khiển một cách đồng bộ.

* **Mô tả kết nối chân của LCD với vi điều khiển ESP32 :**



Hình 4 Sơ đồ kết nối chân khối hiển thị LCD 16x2

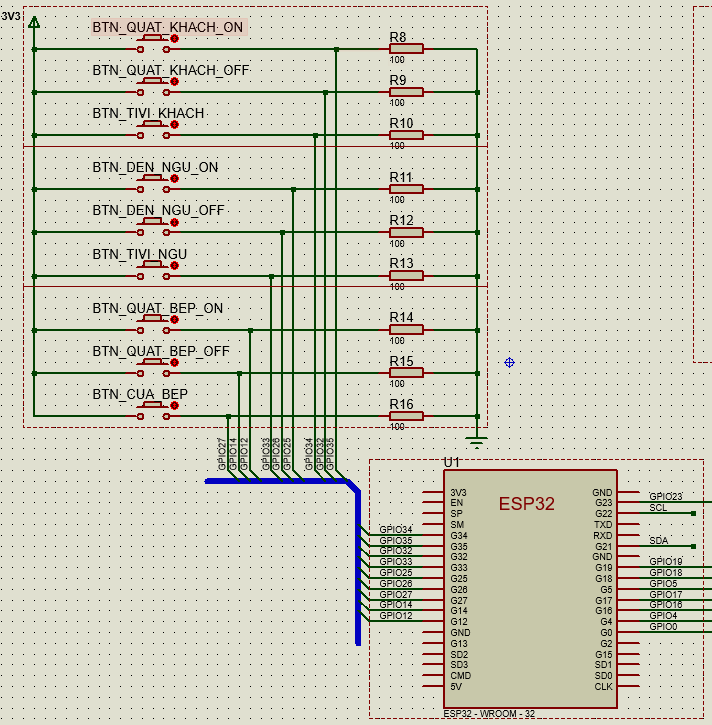
Dựa vào hình 4 :

* Chân SCL và SDA của chuẩn giao tiếp I2C được kết nối với chân G22 và G21 của vi điều khiển.
* Chân VCC được cấp nguồn 5V và chân GND được nối với chân GND của vi điều khiển.

### 2.2. Khối nút nhấn.

* **Sơ đồ nguyên lý**

**Khối nút nhấn :** Để vi điều khiển có thể nhận biết khi nào có nút nhấn được bấm, ta sử dụng các nút nhấn một trạng thái và hai trạng thái để tạo ra mức logic thay đổi để phục vụ cho mục đích điều khiển các thiết bị ngoại vi.



Hình 5 Sơ đồ nguyên lý khối nút nhấn

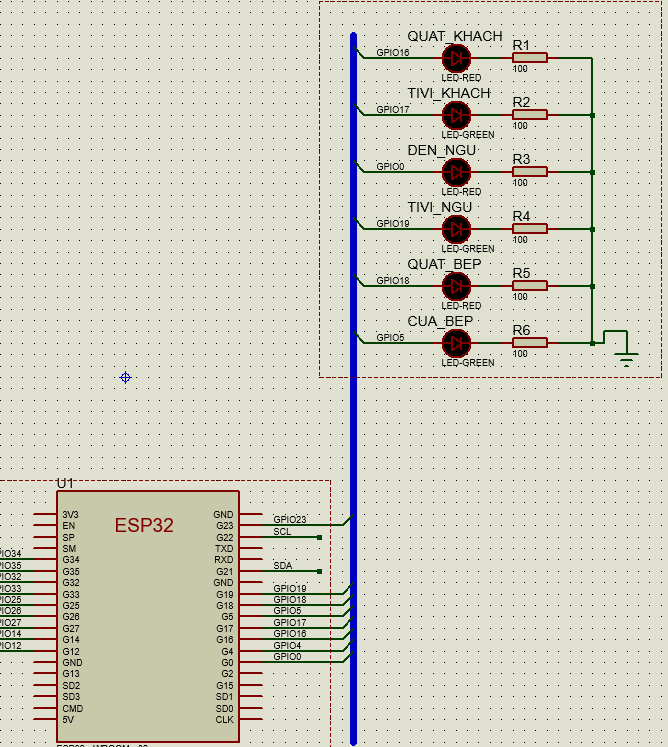
Hình 5 là sơ đồ nguyên lý của khối nút nhấn bao gồm : Các điện trở R8 đến R16 (100 Ω) được kết nối để kéo xuống và xác định mức thấp khi không có nút nào được nhấn. Kèm theo đấy là các nút nhấn :

* Phòng khách : Nút nhấn quạt trạng thái bật, tắt và nút Tivi phòng khách lần lượt nối vào chân G35, G32 và G34 của vi điều khiển.
* Phòng ngủ : Nút nhấn đèn ngủ trạng thái bật, tắt và nút Tivi phòng ngủ lần lượt nối vào chân G25, G26 và G33 của vi điều khiển.
* Nhà bếp : Nút nhấn quạt bếp trạng thái bật, tắt và nút điều khiển cửa bếp lần lượt nối vào chân G12, G14 và G27 của vi điều khiển.

### 2.3. Khối thiết bị ngoại vi.

* **Sơ đồ nguyên lý**

**Khối thiết bị ngoại vi :** Thay thế các thiết bị ngoại vi bằng đèn led để dễ dàng mô phỏng mô hình nhà thông minh với trạng thái sáng đèn là thiết bị on và đèn tắt là thiết bị off.



Hình 6 Sơ đồ nguyên lý khối thiết bị ngoại vi (LED)

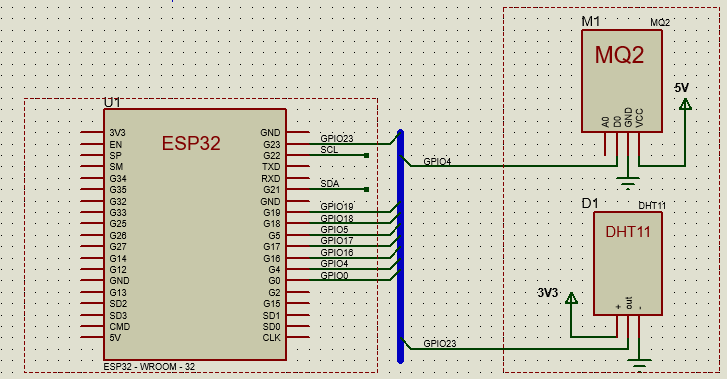
Hình 6 trên là sơ đồ nguyên lý của khối thiết bị ngoại vi bao gồm : Các điện trở R1 đến R6 (100 Ω) để bảo vệ đèn và tương ứng với đó là các led xem như là các thiết bị ngoại vi :

* Quạt và Tivi phòng khách lần lượt nối vào chân G16 và G17 của vi điều khiển.
* Đèn và Tivi phòng ngủ lần lượt nối vào chân G0 và G19 của vi điều khiển.
* Quạt và cửa nhà bếp lần lượt nối vào chân G18 và G5 của vi điều khiển.

### 2.4. Khối cảm biến.

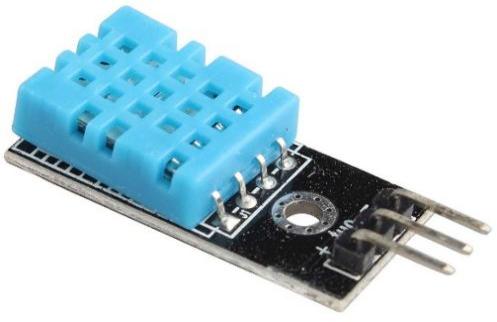
* **Sơ đồ nguyên lý**

**Khối cảm biến :** Sử dụng hai cảm biến chính là DHT11 dùng để đo đạt và thu nhận số liệu nhiệt độ và độ ẩm của môi trường và cảm biến MQ2 dùng để thông báo lượng khí gas bị rò rĩ trong môi trường làm hạn chế rủi ro cho căn phòng.



Hình 7 Sơ đồ nguyên lý khối cảm biến.

Hình 7 trên là sơ đồ nguyên lý của khối cảm biến bao gồm hai thiết bị cảm biến là DHT11 và MQ2 đều sử dụng nguồn 5V đầu ra của cảm biến MQ2 đã qua xử lý sẽ được gửi về thông qua việc nối với chân G4 của vi điều khiển tương tự như vậy đầu ra của cảm biến DHT11 được nối với chân G23 của vi điều khiển.

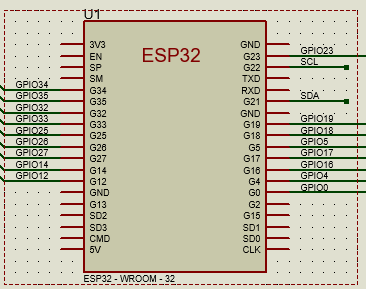
Hình 8a Cảm biến DHT11. Hình 8b Cảm biến MQ2.

Hình 8a và 8b là hình ảnh thực tế của cảm biến Nhiệt độ, độ ẩm (DHT11) và cảm biến khí gas ( MQ2 ) được sử dụng trong mô hình ngôi nhà thông minh.

### 2.5. Khối xử lý trung tâm.

* **Sơ đồ nguyên lý**

Khối xử lý trung tâm được thiết kế để kết nối và truyền thông với các khối khác, nhận và xử lý tín hiệu từ khối thiết bị ngoại vi và khối cảm biến để tính toán, xử lý và đưa ra số liệu, tín hiệu điều khiển để hiển thị lên LCD cũng như trang wed người dùng.



Hình 9 Sơ đồ nguyên lý khối xử lý trung tâm.

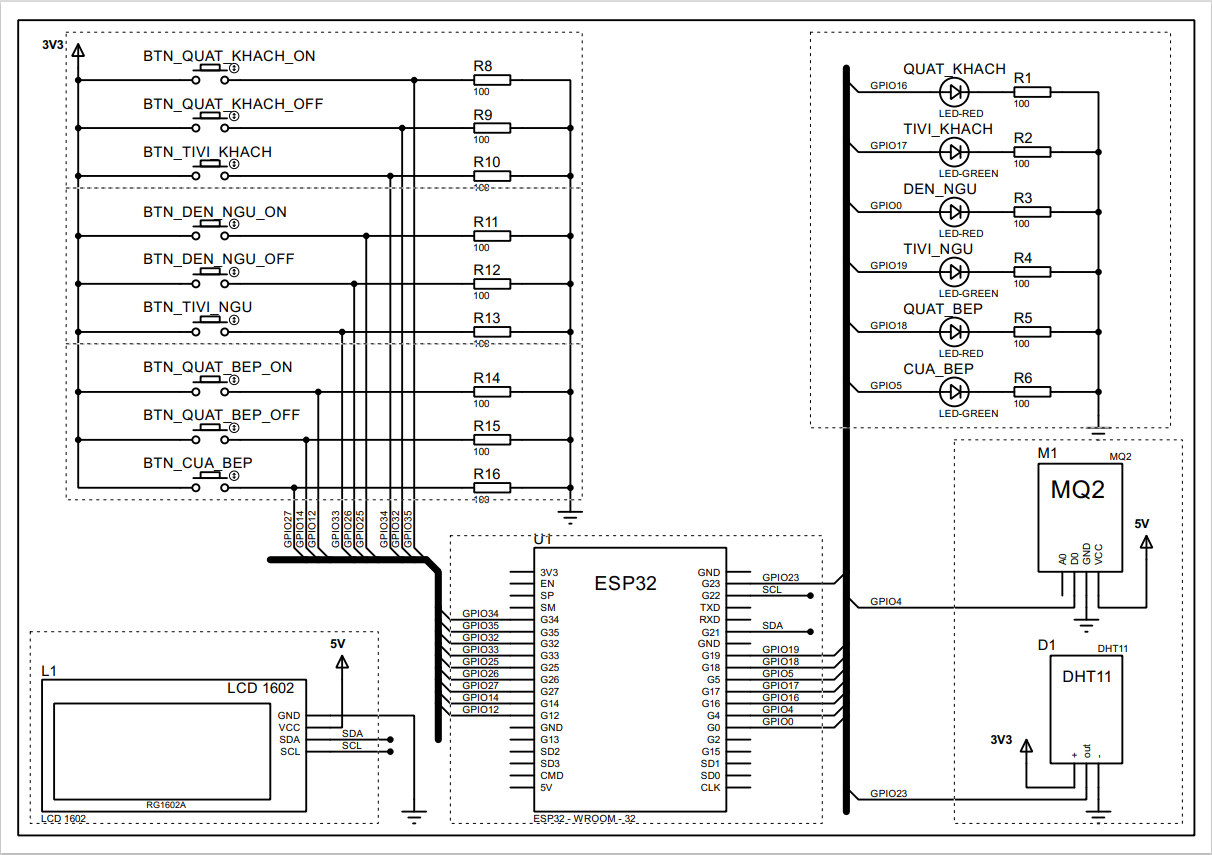
Hình 9 cho ta thấy vi điều khiển sử dụng trong mô hình là ESP32 được ví như bộ não hoạt động của cả một hệ thống. Kết nối các tín hiệu từ thiết bị ngoại vi, cảm biến gửi về nền tảng firebase và hiện các thông số môi trường cũng như trạng thái các thiết bị lên wed người dùng.

* **Thông số của ESP32 :**

|  |  |
| --- | --- |
| Số lượng lõi | 2 (dual core) |
| Wi-Fi | 2,4 GHz lên đến 150 Mbits / s |
| Bluetooth | BLE (Bluetooth Low Energy) và legacy Bluetooth |
| Kiến trúc | 32 bit |
| Clock frequency | Lên đến 240 MHz |
| RAM | 512 KB |
| Số chân (Pin) | 30 hoặc 36 (phụ thuộc vào model) |

Bảng thông số của ESP 32

### 2.6 Sơ đồ kết nối toàn hệ thống



Hình 10. Sơ đồ kết nối toàn hệ thống

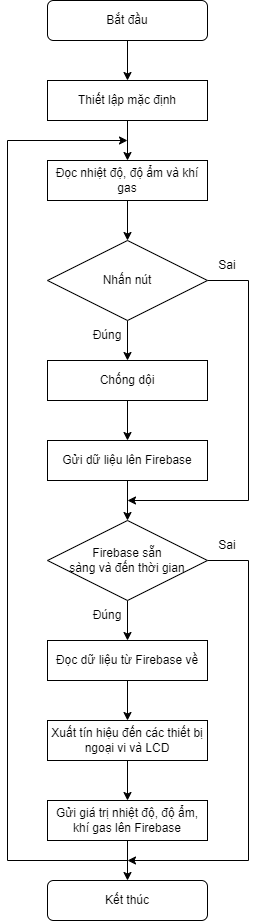
## 3. CHỨC NĂNG VÀ HOẠT ĐỘNG CỦA PHẦN MỀM

### 3.1 Yêu cầu của phần mềm.

* Hiển thị thời gian và các thông số của cảm biến, trạng thái của các thiết bị ngoại vi lên trên wed và lcd.
* Có chức năng điều chỉnh từ xa các thiết bị của từng phòng và cập nhật trạng thái lên wed.
* Có chức năng thu thập số liệu các yếu tố môi trường xung quang cụ thể là đo nhiệt độ và độ ẩm, khí gas và gửi chính xác lên nền tảng firebase và hiện lên wed.

### 3.2 Lưu đồ giải thuật của hệ thống nhà thông minh.

Để hiểu rõ hơn về cách thức hoạt động của hệ thống nhà thông minh để thực thi phần mềm của hệ thống thì chúng ta vẽ ra lưu đồ giải thuật của dự án :



Hình 11 Lưu đồ hoạt động của hệ thống.

Hình 11 là lưu đồ hoạt động của khối xử lý trung tâm (ESP32), thực hiện đọc và gửi dữ liệu lên Firebase:

* Đầu tiên thiết lập các chân ngõ vào, ra. Thiết lập giao tiếp với DHT11, MQ2 và LCD1602. Thiết lập kết nối đến Firebase thông qua Wi-Fi.
* Đọc giá trị nhiệt độ, độ ẩm từ DHT11 và khí gas của MQ2.
* Kiểm tra các nút nhấn, nếu có nhấn nút được nhấn thì sẽ thực hiện việc chống dội và gửi dữ liệu của nút nhấn đó lên Firebase ngay lập tức. Nếu không thì bỏ qua chống dội và gửi dữ liệu.
* Nếu Firebase sẵn sàng cho việc truyền nhận dữ liệu và đến thời gian thì ESP32 sẽ: đọc dữ liệu từ Firebase, xuất các tín hiệu đọc được đến các chân ngõ ra để điều khiển thiết bị và hiển thị trên LCD, gửi dữ liệu nhiệt độ, độ ẩm và khí gas lên Firebase(sau một khoảng thời gian thì ESP32 sẽ cập nhật dữ liệu 1 lần).

# 

# KẾT QUẢ HỆ THỐNG

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

Hình 12 – Kết quả phòng khách

Hình 12 – Kết quả phòng khách cho thấy: Quạt và tivi đang được bật và các thiết bị phòng khác đang tắt. Giá trị loa phòng khách (L.K) được hiển thị trên LCD và trên Web là 80, nhiệt độ 33.3 °C[](https://thuthuatnhanh.com/cach-viet-do-c-trong-excel-them-ky-tu-dac-biet/) và độ ẩm là 76%, trạng thái khí gas là bình thường.

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

Hình 13 Kết quả phòng ngủ

Hình 13 – Kết quả phòng ngủ cho thấy: Đèn và tivi phòng ngủ đang được bật, thiết bị phòng khách đang bật, thiết bị nhà bếp đang tắt. Giá trị loa phòng ngủ (L.N) được hiển thị trên LCD và trên Web là 50, nhiệt độ 33.8 °C và độ ẩm là 72%, trạng thái khí gas bình thường.

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

Hình 14 Kết quả nhà bếp

Hình 14 – Kết quả nhà bếp cho thấy: Quạt và cửa bếp đang được bật, thiết bị các phòng khác đang tắt. Giá trị máy thông gió (T.Gio) được hiển thị trên LCD và trên Web là 68, nhiệt độ 34.2 °C và độ ẩm là 71%, trạng thái khí gas đang được cảnh báo.



Hình 15 Thành viên nhóm

Link github của dự án nhà thông minh: <https://github.com/ThienHuynh2605/SmartHome_FinalProject.git>

# References

1. HTML tutorial(w3schools.com)<https://www.w3schools.com/html/html_intro.asp>
2. CSS tutorial(w3schools.com)<https://www.w3schools.com/css/css_intro.asp>
3. ESP32 - <https://dientutuonglai.com/esp32-la-gi.html>
4. DHT11 -<https://hshop.vn/products/cam-bien-do-am-nhiet-do-dht11>
5. MQ2 - <https://chotroihn.vn/module-cam-bien-khi-gas-mq-2>

Mục lục

[HTML (tag, features), CSS (style), Javascript [1] 1](#_Toc20238)

[1. HTML 1](#_Toc30682)

[2. CSS 40](#_Toc24206)

[THIẾT KẾ HỆ THỐNG 118](#_Toc6360)

[1. SƠ ĐỒ KHỐI CỦA HỆ THỐNG 118](#_Toc10680)

[1.1. Yêu cầu của hệ thống: 118](#_Toc22285)

[1.2. Sơ đồ phác thảo của hệ thống : 118](#_Toc4510)

[1.3. Sơ đồ cây của hệ thống : 119](#_Toc13040)

[1.4. Sơ đồ khối của hệ thống: 120](#_Toc20954)

[2.THIẾT KẾ HỆ THỐNG PHẦN CỨNG. 121](#_Toc19113)

[2.1. Khối hiển thị. 121](#_Toc1292)

[2.2. Khối nút nhấn. 122](#_Toc14112)

[2.3. Khối thiết bị ngoại vi. 124](#_Toc19991)

[2.4. Khối cảm biến. 125](#_Toc20174)

[2.5. Khối xử lý trung tâm. 126](#_Toc21573)

[2.6 Sơ đồ kết nối toàn hệ thống 128](#_Toc25526)

[3. CHỨC NĂNG VÀ HOẠT ĐỘNG CỦA PHẦN MỀM 128](#_Toc29137)

[3.1 Yêu cầu của phần mềm. 128](#_Toc12123)

[3.2 Lưu đồ giải thuật của hệ thống nhà thông minh. 129](#_Toc26124)

[KẾT QUẢ HỆ THỐNG 132](#_Toc15437)

[References 134](#_Toc1469)