Expérimentation digicode Arduino

Un projet de Lola de BOURGUES et Edouard DARROUSEZ UX3B

Sommaire

Introduction

Consignes

“La production attendue est la conception, la réalisation et la documentation d'un objet connecté exploitant une ou plusieurs ESP32.

Aussi, votre objet pourra supporter un usage sérieux ou réaliste. L'usage peut tout aussi bien être expérimental.

Ce qui est attendu dans ce projet n'est pas uniquement la réalisation du dispositif mais surtout sa documentation, c'est-à-dire un document contenant toutes les informations pour comprendre le projet et le reproduire.”

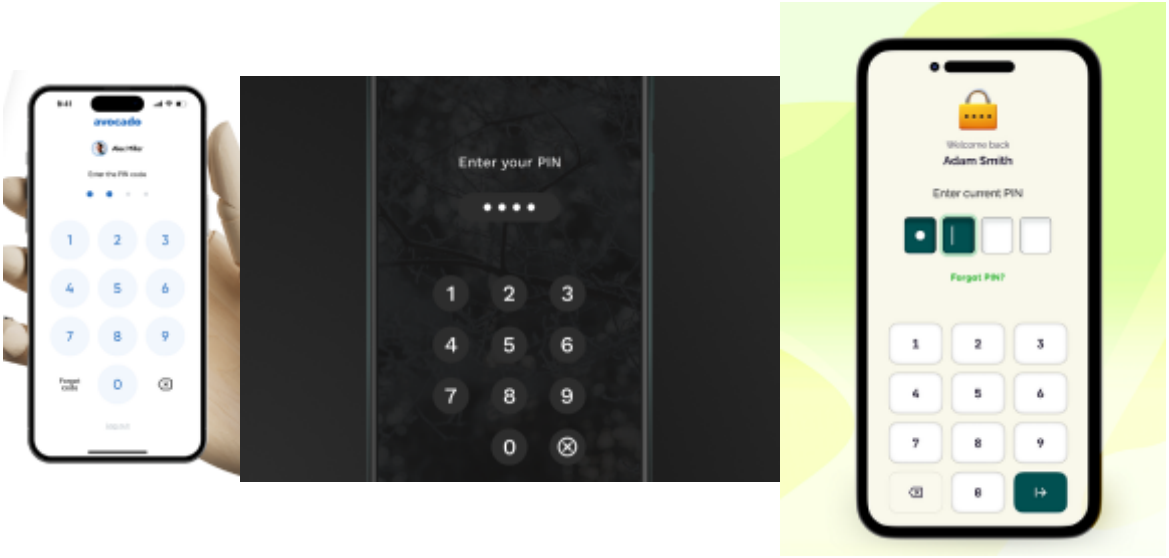
A propos

Nous avons opté pour une approche expérimentale en matière de conception, qui se matérialise actuellement par une page internet verrouillée nécessitant l'entré d'un mot de passe. Lorsqu'un utilisateur saisit le mot de passe correct, un signal est instantanément transmis à la carte arduino. On peut alors imaginer de nombreux usages comme ouvrir un verrou connecté...

Inspirations

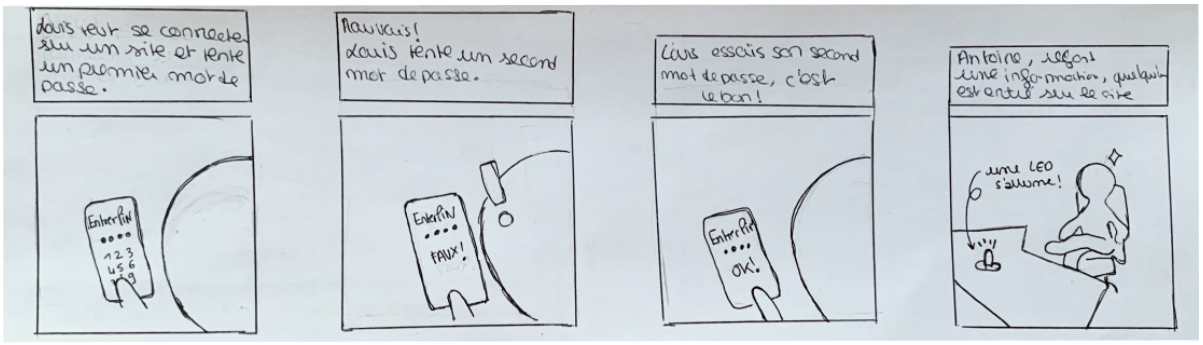
Pour ce projet, nous nous sommes inspirés de plusieurs domaines et technologies comme le “CAPTCHA” est un mécanisme de sécurité utilisé sur les sites web pour différencier les utilisateurs humains des programmes automatisés, appelés bots. Ou alors des systèmes de paiement, qui quand la transaction est validée envoient un message au terminal pour déclencher une action.

[Arduino Hardware wallet - Using Arduino / Project Guidance - Arduino Forum]
[Arduino-Based Bitcoin Candy Vending Machine - Hackster.io]



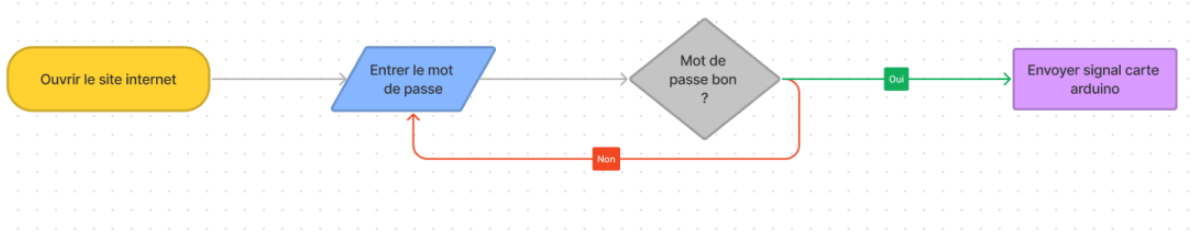
Détails

Schéma d'usage



Technique

Logigramme



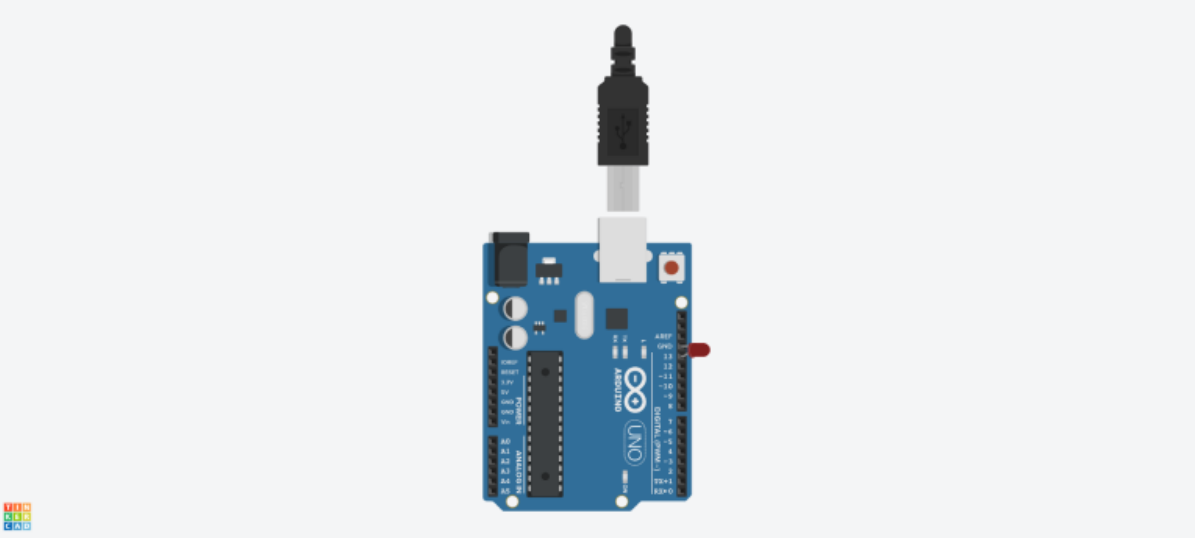
Matériel nécessaire

- 1*Carte arduino
- 1*Routeur
- 1*Led

Réalisation

Schéma électronique

Le schéma de montage de notre réalisation est assez simple, car, nous testons la réception du message uniquement avec l'allumage d'une Led.



Code

```
// Charger la librairie Wifi
#include <WiFiS3.h>

//Entrer les informations de connexion au Wifi
const char* ssid = "HUAWEI Edouard";
const char* password = "DGMYX34XK2M1";

// Définir le port du serveur sur 80
WiFiServer server(80);

// variable pour stocker les requêtes HTTP
String header;

// Variable pour stocker l'état de la sortie
String outputState = "off";

// Variable pour la pin de sortie
const int outputPin = 13;

// Current time
unsigned long currentTime = millis();
// Previous time
unsigned long previousTime = 0;
// Define timeout time in milliseconds (example: 2000ms = 2s)
const long timeoutTime = 2000;

void setup() {
  Serial.begin(115200);
  // Initialize the output variables as outputs
  pinMode(outputPin, OUTPUT);
  // Set outputs to LOW
```

```
  digitalWrite(outputPin, LOW);

  // Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}

void loop() {
  WiFiClient client = server.available();  // Listen for incoming clients

  if (client) {  // If a new client connects, (un client est un device qui se
connecte au serveur.)
    currentTime = millis();
    previousTime = currentTime;
    Serial.println("New Client.");                                          //
print a message out in the serial port
    String currentLine = "";                                                //
make a String to hold incoming data from the client
    while (client.connected() && currentTime - previousTime <= timeoutTime) {  //
loop while the client's connected
      currentTime = millis();
      if (client.available()) {  // if there's bytes to read from the client,
        char c = client.read();  // read a byte, then
        Serial.write(c);         // print it out the serial monitor
        header += c;
        if (c == '\\n') {  // if the byte is a newline character
          // if the current line is blank, you got two newline characters in a
row.
          // that's the end of the client HTTP request, so send a response:
          if (currentLine.length() == 0) {
            // HTTP headers always start with a response code (e.g. HTTP/1.1 200
OK)
            // and a content-type so the client knows what's coming, then a blank
line:
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

            // turns the GPIOs on and off == si dans le barre d'adresse, il y a
/mdp-true, met la variable outputState sur "On"
            if (header.indexOf("GET /mdp-true") >= 0) {
              Serial.println("GPIO on");
              outputState = "on";
            }
```

```
            // si home dans barre d'adresse, outputState = "off"
            if (header.indexOf("GET /home") >= 0) {
              Serial.println("GPIO off");
              outputState = "off";
            }

            //Vérifier si outputState est "on"
            if (outputState == "on") {
              digitalWrite(outputPin, HIGH);
              //envoyer au client la page web
              client.println("<!DOCTYPE html><html lang=\\"fr\\"><head> <meta
charset=\\"UTF-8\\"> <meta name=\\"viewport\\" content=\\"width=device-width,
initial-scale=1.0\\"> <title>Code correct</title></head><body> <h1>Code bon !
</h1> <a href=\\"/home\\"><button>retour</button></a></body></html>");
            }
            //Vérifier si outputState est "off"
            if (outputState == "off"){
              digitalWrite(outputPin, LOW);
              //envoyer au client la page web
```

```
            client.println("<!DOCTYPE html><html lang=\"fr\"><head> <meta
charset=\"UTF-8\"> <meta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0\"> <title>Arduino connect screen</title> <script src=\"
<https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js\>"></script>
<style>@import url('<https://fonts.googleapis.com/css2?
family=Inter:wght@100..900&display=swap>'); :root{--indicator-inactive: #CBD2E0;
--dark: #2D3648;}header{display: flex; justify-content: center; margin-bottom:
1rem;}main{display: flex; flex-direction: column; align-items: center; gap:
2.5rem;}footer{margin-top: 2rem; display: flex; justify-content: center; flex-
direction: column;}h1{margin: 0; color: var(--dark); font-family: Inter; font-
size: 1.25rem; font-style: normal; font-weight: 700; line-height:
150%;}button{text-decoration: none; background-color: white; border: none;
margin: 0; width: auto;}.logo-arduino{width: 6rem; height:
6rem;}.indicator{width: 0.825rem; height: 0.825rem; background-color: var(--
indicator-inactive); border-radius: 0.5rem;}.digit-row{display: flex; align-
items: flex-start; gap: 1.5rem;}.digit{display: inline-flex; flex-direction:
column; align-items: flex-start; gap: 2rem;}.digit-number-text{color: var(--
dark); text-align: center; /* Header - Medium */ font-family: Inter; font-size:
1.5rem; font-style: normal; font-weight: 600; line-height: 150%; transition: all
300ms;}.digit-number-layout{display: flex; width: 5rem; height: 3rem; padding:
1.1875rem 0rem; justify-content: center; align-items: center; border-radius:
2.5rem; margin: 0;}.steps{display: inline-flex; justify-content: center; align-
items: center; gap: 1.5rem;}.forgot-pass{color: var(--dark); text-align: center;
font-family: Inter; font-size: 0.875rem; font-style: normal; font-weight: 700;
line-height: 150%; opacity: 0.6;}.active-number{color: aqua; box-flex-group:
pink;}.new-input{visibility: hidden;}.indicator-active{background-color:
blue;}.indicator-true{background-color: #1ADE5D;}</style> <script>$(function(){
$(\"button\").click(function(){ $(this).addClass(\"active-number\");
$(this).removeClass(\"digit-number-text\"); window.setTimeout(function()
{$(\"button\").removeClass(\"active-number\");
$(\"button\").addClass(\"digit-number-text\");},500); return;});}); </script>
<script>let generatedPin=\"1249\"; let enteredPin=\"\"; let entryCount=0;
function addNumber(num){enteredPin +=num;
document.getElementById('enteredPin').value=enteredPin; console.log(enteredPin);
entryCount ++; console.log(entryCount); if (enteredPin===generatedPin)
{console.log(\"Mot de passe trouvé\");
$(\".indicator\").addClass(\"indicator-true\"); setTimeout(clearInput, 500);
window.location.href='/mdp-true';}else{console.log(\"mot de passe faux\");}if
(entryCount==1){console.log(\"1 entrée\"); $(\"#indicator-
1\").addClass(\"indicator-active\");}if (entryCount==2){console.log(\"2
entrée\"); $(\"#indicator-2\").addClass(\"indicator-active\");}if
(entryCount==3){console.log(\"3 entrée\"); $(\"#indicator-
3\").addClass(\"indicator-active\");}if (entryCount==4){console.log(\"4
entrée\"); $(\"#indicator-4\").addClass(\"indicator-active\");
setTimeout(clearInput, 500);}return;}function clearInput(){enteredPin=\"\";
entryCount=0; document.getElementById('enteredPin').value=enteredPin;
console.log('clear'); $('.indicator').removeClass('indicator-active');
$('.indicator').removeClass('indicator-true'); return;}</script></head><body>
<header> <img src=\"
<https://cdn.shopify.com/s/files/1/0438/4735/2471/files/Arduino_RGB_Pantone_9a224
c8c-5d1d-4e5a-8e26-db3aec5ea7db.png?v=1637755799\>" alt=\"logo arduino\"
class=\"logo-arduino\"> </header> <main> <h1>Enter Pin</h1> <section
class=\"steps\"> <article class=\"indicator\" id=\"indicator-1\"></article>
<article class=\"indicator\" id=\"indicator-2\"></article> <article
class=\"indicator\" id=\"indicator-3\"></article> <article
class=\"indicator\" id=\"indicator-4\"></article> </section> <section
```

```
class=\\"digit\\"> <article class=\\"digit-row\\"> <button class=\\"digit-number-
text digit-number-layout\\" onclick=\\"addNumber(1)\\">1</button> <button
class=\\"digit-number-text digit-number-layout\\"
onclick=\\"addNumber(2)\\">2</button> <button class=\\"digit-number-text digit-
number-layout\\" onclick=\\"addNumber(3)\\">3</button> </article> <article
class=\\"digit-row\\"> <button class=\\"digit-number-text digit-number-layout\\"
onclick=\\"addNumber(4)\\">4</button> <button class=\\"digit-number-text digit-
number-layout\\" onclick=\\"addNumber(5)\\">5</button> <button class=\\"digit-
number-text digit-number-layout\\" onclick=\\"addNumber(6)\\">6</button>
</article> <article class=\\"digit-row\\"> <button class=\\"digit-number-text
digit-number-layout\\" onclick=\\"addNumber(7)\\">7</button> <button
class=\\"digit-number-text digit-number-layout\\"
onclick=\\"addNumber(8)\\">8</button> <button class=\\"digit-number-text digit-
number-layout\\" onclick=\\"addNumber(9)\\">9</button> </article> <article
class=\\"digit-row\\"> <button class=\\"digit-number-layout\\"></button> <button
class=\\"digit-number-text digit-number-layout\\"
onclick=\\"addNumber(0)\\">0</button> <button class=\\"digit-number-layout\\">
</button> </article> </section> </main> <footer> <p class=\\"forgot-
pass\\">Forgot your Pin Code ?</p><input type=\\"text\\" id=\\"enteredPin\\"
class=\\"new-input\\" readonly> </footer></body></html>");
            }


            // The HTTP response ends with another blank line
            client.println();
            // Break out of the while loop
            break;
          } else {  // if you got a newline, then clear currentLine
            currentLine = "";
          }
        } else if (c != '\\r') {  // if you got anything else but a carriage
return character,
          currentLine += c;      // add it to the end of the currentLine
        }
      }
    }
    // Clear the header variable
    header = "";
    // Close the connection
    client.stop();
    Serial.println("Client disconnected.");
    Serial.println("");
  }
}
```

Code page web

```
<!-- setup page HTML -->
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Arduino connect screen</title>
    <!-- import jquery -->
```

```
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<!-- Style CSS -->
    <style>
    @import url('https://fonts.googleapis.com/css2?
family=Inter:wght@100..900&display=swap');

        :root{
            --indicator-inactive: #CBD2E0;
            --dark: #2D3648;
        }

        header{
            display: flex;
            justify-content: center;
            margin-bottom: 1rem;
        }

        main{
            display: flex;
            flex-direction: column;
            align-items: center;
            gap: 2.5rem;
        }

        footer{
            margin-top: 2rem;
            display: flex;
            justify-content: center;
            flex-direction: column;
        }

        h1{
            margin: 0;
            color: var(--dark);
            font-family: Inter;
            font-size: 1.25rem;
            font-style: normal;
            font-weight: 700;
            line-height: 150%;
        }

        button{
            text-decoration: none;
            background-color: white;
            border: none;
            margin: 0;
            width: auto;
        }

        .logo-arduino{
            width: 6rem;
            height: 6rem;
        }

        .indicator{
```

```
        width: 0.825rem;
        height: 0.825rem;
        background-color: var(--indicator-inactive);
        border-radius: 0.5rem;
    }

    .digit-row{
        display: flex;
        align-items: flex-start;
        gap: 1.5rem;
    }

    .digit{
        display: inline-flex;
        flex-direction: column;
        align-items: flex-start;
        gap: 2rem;
    }

    .digit-number-text{
        color: var(--dark);
        text-align: center;
        /* Header - Medium */
        font-family: Inter;
        font-size: 1.5rem;
        font-style: normal;
        font-weight: 600;
        line-height: 150%;
        transition: all 300ms;
    }

    .digit-number-layout{
        display: flex;
        width: 5rem;
        height: 3rem;
        padding: 1.1875rem 0rem;
        justify-content: center;
        align-items: center;
        border-radius: 2.5rem;
        margin: 0;
    }

    .steps{
        display: inline-flex;
        justify-content: center;
        align-items: center;
        gap: 1.5rem;
    }

    .forgot-pass{
        color: var(--dark);
        text-align: center;
        font-family: Inter;
        font-size: 0.875rem;
        font-style: normal;
        font-weight: 700;
```

```
            line-height: 150%;
            opacity: 0.6;
        }

        .active-number{
            color: aqua;
            box-flex-group: pink;
        }

        .new-input{
            visibility: hidden;
        }

        .indicator-active{
            background-color: blue;
        }

        .indicator-true{
            background-color: #1ADE5D;
        }
    </style>

    <!-- Script JS changer style boutons -->
    <script>

              $(function() {                        //run when the DOM is ready
        $("button").click(function() {  //use a class, since your ID gets mangled
            $(this).addClass("active-number");      //add the class to the
clicked element
            $(this).removeClass("digit-number-text");
            window.setTimeout(function(){
                $("button").removeClass("active-number");
                $("button").addClass("digit-number-text");
            },500);
            return;
  });
});
    </script>
    <!-- Script JS Digicode -->
    <script>
        // Définir variables
        let generatedPin = "1249";
        let enteredPin = "";
        let entryCount = 0;

        // Définition fonction addNumber
        function addNumber(num) {
            enteredPin += num;
            document.getElementById('enteredPin').value = enteredPin;
            console.log(enteredPin);
            entryCount ++;
            console.log(entryCount);
            //vérifier si le code entré est le même que celui prévu
            if (enteredPin === generatedPin) {
                console.log("Mot de passe trouvé");
                $(".indicator").addClass("indicator-true");
```

```
                setTimeout(clearInput, 500);
                window.location.href = '/mdp-true';
            } else {
                console.log("mot de passe faux");
            }
            // Vérifier le nombre de chiffres entrés
            if (entryCount == 1){
                console.log("1 entrée");
                $("#indicator-1").addClass("indicator-active");
            }
            if (entryCount == 2){
                console.log("2 entrée");
                $("#indicator-2").addClass("indicator-active");
            }if (entryCount == 3){
                console.log("3 entrée");
                $("#indicator-3").addClass("indicator-active");
            }
            if (entryCount == 4){
                console.log("4 entrée");
                $("#indicator-4").addClass("indicator-active");
                setTimeout(clearInput, 500);
            }
            return;
        }

        // Fonction pour remettre à zéro le digicode
        function clearInput() {
            enteredPin = "";
            entryCount = 0;
            document.getElementById('enteredPin').value = enteredPin;
            console.log('clear');
            $('.indicator').removeClass('indicator-active');
            $('.indicator').removeClass('indicator-true');
            return;
        }

    </script>
<!-- Scructure HTML de la page web -->
</head>
<body>
    <header>
        <img
src="https://cdn.shopify.com/s/files/1/0438/4735/2471/files/Arduino_RGB_Pantone_9
a224c8c-5d1d-4e5a-8e26-db3aec5ea7db.png?v=1637755799" alt="logo arduino"
class="logo-arduino">
    </header>
    <main>
          <h1>Enter Pin</h1>
          <section class="steps">
            <article class="indicator" id="indicator-1"></article>
            <article class="indicator" id="indicator-2"></article>
            <article class="indicator" id="indicator-3"></article>
            <article class="indicator" id="indicator-4"></article>
          </section>
          <section class="digit">
            <article class="digit-row">
```

```
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(1)">1</button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(2)">2</button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(3)">3</button>
            </article>
            <article class="digit-row">
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(4)">4</button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(5)">5</button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(6)">6</button>
            </article>
            <article class="digit-row">
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(7)">7</button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(8)">8</button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(9)">9</button>
            </article>
            <article class="digit-row">
                <button class="digit-number-layout"></button>
                <button class="digit-number-text digit-number-layout"
onclick="addNumber(0)">0</button>
                <button class="digit-number-layout"></button>
            </article>
          </section>
    </main>
    <footer>
        <p class="forgot-pass">Forgot your Pin Code ?</p>
        <input type="text" id="enteredPin" class="new-input" readonly>
    </footer>
</body>
</html>
```

Pour avoir accès au code :

https://github.com/edarrous/Arduino-web-digicode.git

Sources

Ahmed, A. (2024, avril 9). *A Step-by-Step Guide To Generate and Validate 4-Digit PINs In JavaScript*. Codeconvey. https://codeconvey.com/a-step-by-step-guide-to-generate-and-validate-4-digit-pins-in-javascript/

Arduino—HTTP Request | Arduino Tutorial. (s. d.-a). Arduino Getting Started. Consulté 28 mai 2024, à l'adresse https://arduinogetstarted.com/tutorials/arduino-http-request

Arduino—HTTP Request | Arduino Tutorial. (s. d.-b). Arduino Getting Started. Consulté 27 mai 2024, à l'adresse https://arduinogetstarted.com/tutorials/arduino-http-request

ESP32 Web Server—Arduino IDE | Random Nerd Tutorials. (s. d.). Consulté 28 mai 2024, à l'adresse https://randomnerdtutorials.com/esp32-web-server-arduino-ide/

Rui Santos (Réalisateur). (2018). *Build an ESP32 Web Server with Arduino IDE*. https://www.youtube.com/watch?v=ApGwxX6VVzk

Verstraete, A. (2018, janvier 6). *ESP32 Arduino HTTP Server : Serving HTML and JavaScript - techtutorialsx*. https://techtutorialsx.com/2018/01/06/esp32-arduino-http-server-serving-html-and-javascript/