

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | آردوینوکنترل | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | امیر آزادی |  | | | |
|  | | | | 5/4/1403**—**اصول رباتيکز**—**استاد : دکتر محمد زارع **How to create a website communicating Arduino by using PHP** |  | | | |
|  | | |  | | |  | | |

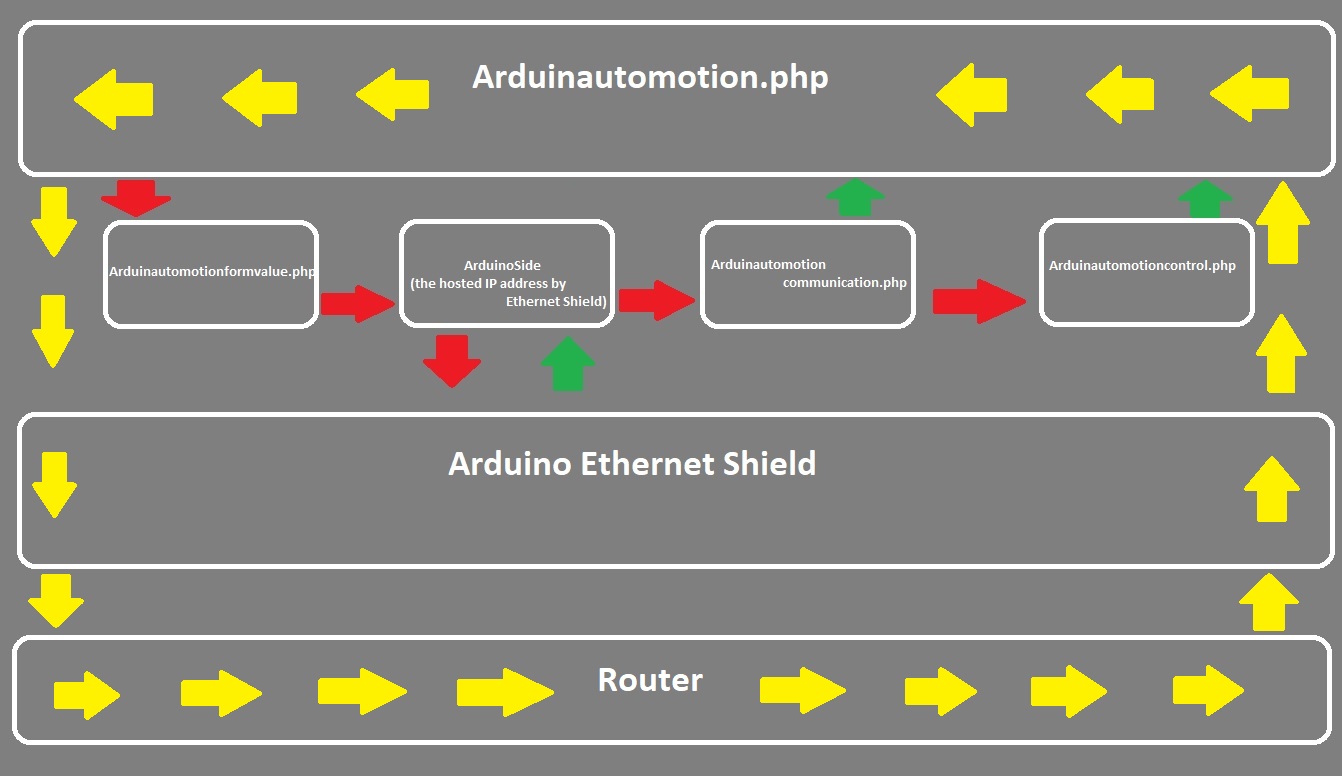
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | چکیده | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  |  | پروژه من ایجاد یک وبسایت حرفه‌ای برای کنترل آکواریوم و سیستم آبیاری خانه است و همچنین دریافت اطلاعات خانه وقتی که در خانه نیستم.  برای این کار، پروژه را ایجاد کردم که شامل ۴ صفحه HTML مختلف) در سرور محلی Apache من است. گاهی اوقات نیاز دارم که فیلتر داخلی را خاموش کنم تا سطح اکسیژن در آکواریوم افزایش نیابد یا از راه دور به ماهی‌ها غذا بدهم. شش گزینه برای کنترل آکواریوم و یک گزینه برای فعال‌سازی سیستم آبیاری اضافه کرد  علاوه بر این، می‌توانید به راحتی اطلاعاتی مانند دما و رطوبت (از سنسور DHT11) یا میزان گاز متان در خانه (از سنسور MQ4) را از طریق Arduinautomotion و Arduino Ethernet Shield دریافت کنید.  وبسایت Arduinautomotion رابط کاربری ساده‌ای برای دریافت و ارسال داده‌ها به و از آردوینو دارد.  برای درک کامل محتوای پروژه، نیاز به دانش HTML، CSS، JavaScript، PHP و jQuery است. این پروژه فقط یک نمایش از پروژه است، ولی اگر این زبان‌ها را بلد باشید، می‌توانید تمامی صفحات وب را به صورت فایل‌های کد PHP پیدا کنی | | |  |  |  |
| Decorative | | | | |
|  |  |  |  |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decorative | |  |  | | |  |  | |
|  | | روند انجام پروژه | | | | |  | |
|  | |  |  | | |  |  | |
|  | **ارتباط با آردوینو:**   * **یک صفحه وب دیگر به نام ArduinoSide.php روی Arduino Ethernet Shield میزبانی می‌شود و از طریق آدرس IP مشخص قابل دسترسی است.**   **ارسال و دریافت داده‌ها:**   * **از متد POST برای ارسال داده‌های امن بین صفحات PHP استفاده می‌شود.** * **از متد GET برای ارتباط با Arduino Ethernet Shield استفاده می‌شود.** * **داده‌ها بین صفحات وب با استفاده از PHP $\_SESSION ذخیره و منتقل می‌شوند.** | | | | | | |  |
|  |  | | |  |  | | |  |
|  | **کنترل تنظیمات شبکه و دسترسی**  برای دسترسی به Localhost از طریق Wi-Fi، از IP محلی و پورت فورواردینگ استفاده می‌شود.  **اتصال سخت‌افزارها**:  سنسورها و رله‌ها به پین‌های آردوینو متصل می‌شوند و ارتباط از طریق کابل ADSL بین Arduino Ethernet Shield و روتر برقرار می‌شود. | | |  | **آکواریوم و سیستم آبیاری**  از فرم‌های HTML برای ارسال دستورات به آردوینو استفاده می‌شود. داده‌ها از Arduinautomotionformvalue.php دریافت و به Arduinautomotioncontrol.php ارسال می‌شوند.  وضعیت سنسورها و دستگاه‌ها در صفحه اصلی (Arduinautomotion.php) نمایش داده می‌شود. | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | PAGE 3 |  |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Code **ArduinoSide :**     |  | | --- | | #include <SPI.h>  #include <Ethernet.h>  //Initial DHT11 library and create an object named dht to communicate with the sensor.  #include "DHT.h"  DHT dht;  // Enter a MAC address and IP address for your controller below.  // The IP address will be dependent on your local network:  byte mac[] = {    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED  // Your MAC address has to be here.  };  IPAddress ip(192, 168, 1, 21);  // Initialize the Ethernet server library  // with the IP address and port you want to use  // (port 80 is default for HTTP):  EthernetServer server(80);  // Define which pin connects to which object such as a air pump in your aquarium.  int pump = 3;  int lamp = 4;  int feed = 5;  int heater = 6;  int filterex =7;  int filterin =8;  int irrigation =9;  // Determine a string to collect the commands came from Arduiaautomotion.php.  String readString ;  void setup() {    // Open serial communications and wait for port to open:    Serial.begin(9600);    // Determine the relay module's pins.     pinMode(pump,OUTPUT);     pinMode(lamp,OUTPUT);     pinMode(feed,OUTPUT);     pinMode(heater,OUTPUT);     pinMode(filterex,OUTPUT);     pinMode(filterin,OUTPUT);     pinMode(irrigation,OUTPUT);     // Give HIGH value for the default.     digitalWrite(pump,HIGH);     digitalWrite(lamp,HIGH);     digitalWrite(feed,HIGH);     digitalWrite(heater,HIGH);     digitalWrite(filterex,HIGH);     digitalWrite(filterin,HIGH);     digitalWrite(irrigation,HIGH);     //Connect Dht11 to data pin 2.     dht.setup(2);    // Start the Ethernet connection and the server:    Ethernet.begin(mac, ip);    server.begin();    Serial.print("server is at ");    Serial.println(Ethernet.localIP());  }  void loop() {   // This makes Arduino board to wait until the temperature and the humidity values are accurate.   delay(dht.getMinimumSamplingPeriod());   // The datas from DHT11 Sensor.    float humidity = dht.getHumidity();      float temperature = dht.getTemperature();      float fahrenheit = dht.toFahrenheit(temperature);    // Integer values of analog pins'readings.    int ldrValue=analogRead(A0);    int flameValue=analogRead(A1);    int MQ4Value=analogRead(A2);    int HygValueA=analogRead(A3);      int HygValueB=analogRead(A4);    int HygValueC=analogRead(A5);      // Listen for incoming clients    EthernetClient client = server.available();    if (client) {      Serial.println("new client");      // An http request ends with a blank line      boolean currentLineIsBlank = true;      while (client.connected()) {        if (client.available()) {          char c = client.read();          // Add each data came from Arduinautomotion.php to String named readString.          if (readString.length() < 135) {              readString += c;            //Serial.print(c);           }          Serial.println(readString);          // If the line ends, initial a HTTP Request.          if (c == '\n') {            // send a standard http response header            client.println("HTTP/1.1 200 OK");            client.println("Content-Type:text/html");            client.println("Connection: close");  // the connection will be closed after completion of the response            client.println();            client.println("<!DOCTYPE HTML>");            client.println("<html style=""background-color:black;"">");            client.println("<head>");            client.println("<title>ArduinoSide</title>"); // Header for the web page.            client.println("<link rel=""icon"" href=""http://www.clker.com/cliparts/D/W/n/h/n/W/house-logo-hi.png""></link>"); // Describe the icon of the web page.            client.println("</head>");            client.println("<body>");            // http://your IPAddress/your site? is the way to access the server.            client.println("<form action=""http://192.168.1.20/Arduinautomotioncommunication.php?"" method=""get""  target=""\_blank"">"); // The hidden HTML form for sending the information.                      client.println("<input type=""hidden""  name=""temperatureCelsius"" "); // The easiest way to send data to the server is to use the Get Method, but it is not secure.However, the Post Method can be used for security.            client.println("value=");            client.println(temperature);            client.println(">");                      client.println("<input type=""hidden""  name=""temperatureFahrenheit"" ");            client.println("value=");            client.println(fahrenheit);            client.println(">");                      client.println("<input type=""hidden""  name=""humidity"" ");            client.println("value=");            client.println(humidity);            client.println(">");                      client.println("<input type=""hidden""  name=""flame"" ");            client.println("value=");            client.println(flameValue);            client.println(">");                      client.println("<input type=""hidden""  name=""MQ4sensor"" ");            client.println("value=");            client.println(MQ4Value);            client.println(">");                      client.println("<input type=""hidden""  name=""LDRsensor"" ");            client.println("value=");            client.println(ldrValue);            client.println(">");                      client.println("<input type=""hidden""  name=""plant1H"" ");            client.println("value=");            client.println(HygValueA);            client.println(">");                      client.println("<input type=""hidden""  name=""plant2H"" ");            client.println("value=");            client.println(HygValueB);            client.println(">");                      client.println("<input type=""hidden""  name=""plant3H"" ");            client.println("value=");            client.println(HygValueC);            client.println(">");            client.println("<input type=""submit"" value=""Communication"" style=""width:500px;height:200px;font-weight:bold;background-color:black;color:green;"">"); // When the Communication button is clicked, the information is sent to the server from the chosen IPAddress.            client.println("</form>");            client.println("</body>");            client.println("</html>");              break;          }        }      }      // Give the web browser time to receive the data.      delay(1);      // Close the connection:      client.stop();      // Open the objects which are selected by Arduinautomotion.php (such as  internal filter or external filter in the Aquarium or irrigation system in the Plant).             if (readString.indexOf("?airpump=pumpIsOn") >0){                 digitalWrite(pump, LOW);             }             else if (readString.indexOf("?airpump=pumpIsOff") >0){                 digitalWrite(pump, HIGH);             }             if (readString.indexOf("&lamp=lampIsOn") >0){                 digitalWrite(lamp, LOW);             }             else if (readString.indexOf("&lamp=lampIsOff") >0){                 digitalWrite(lamp, HIGH);             }             if (readString.indexOf("&feed=feedIsOn") >0){                 digitalWrite(feed, LOW);             }             else if (readString.indexOf("&feed=feedIsOff") >0){                 digitalWrite(feed, HIGH);             }             if (readString.indexOf("&heater=heaterIsOn") >0){                 digitalWrite(heater, LOW);             }             else if (readString.indexOf("&heater=heaterIsOff") >0){                 digitalWrite(heater, HIGH);             }              if (readString.indexOf("&filterex=exIsOn") >0){                 digitalWrite(filterex, LOW);             }             else if (readString.indexOf("&filterex=exIsOff") >0){                 digitalWrite(filterex, HIGH);             }             if (readString.indexOf("&filterin=inIsOn") >0){                 digitalWrite(filterin, LOW);             }             else if (readString.indexOf("&filterin=inIsOff") >0){                 digitalWrite(filterin, HIGH);             }             if (readString.indexOf("&irrigation=start") >0){                 digitalWrite(irrigation, LOW);             }             else if (readString.indexOf("&irrigation=stop") >0){                 digitalWrite(irrigation, HIGH);             }             readString="";      Serial.println("client disconnected");    }  } | |  |  | |
|  |  |  | |  |
|  |  |  |  | |



|  |  |  |  |
| --- | --- | --- | --- |
|  | در این قسمت شما تنها کد **ArduinoSide مشاهده کردید و برای جلوگیری از پیچیدگی و طولانی شدن متن ارایه بقیه کد ها که شامل html css php هستند را در گیت هاب به آدرس github.com/amirazadi1380**  **آپلود کرده ام.)**  **نکات ضروری در تعریف توابع و بخش‌های مهم کد**   1. **اتصال به شبکه اترنت:**    * **برای اتصال Arduino به شبکه، باید آدرس MAC و IP مشخص شود. سپس با استفاده از دستوراتی، ارتباط اترنت و سرور آغاز می‌شود.** 2. **تعریف و پیکربندی پین‌های ورودی و خروجی:**    * **پین‌هایی که به رله‌ها و سنسورها متصل هستند، به عنوان ورودی یا خروجی تعریف می‌شوند. همچنین وضعیت پیش‌فرض آنها تعیین می‌شود (به عنوان مثال، HIGH برای غیرفعال بودن رله‌ها** 3. **خواندن داده‌ها از سنسورها:**    * **در حلقه اصلی کد (loop)، مقادیر سنسورها خوانده می‌شود. این شامل دما و رطوبت از DHT11 ، و مقادیر مختلف از سنسورهای آنالوگ مانند LDR، سنسور شعله، و سنسور گاز MQ4 می‌شود.** 4. **برقراری ارتباط با کلاینت‌ها:**    * **وقتی یک کلاینت به سرور اترنت متصل می‌شود، درخواست‌های HTTP دریافت و پردازش می‌شود. سپس داده‌های سنسورها به عنوان پاسخ به کلاینت ارسال می‌شود.** 5. **کنترل دستگاه‌ها بر اساس دستورات دریافتی:**    * **دستورات ارسالی از وبسایت برای کنترل دستگاه‌ها (مانند پمپ هوا، لامپ، و سیستم آبیاری) پردازش می‌شود و بر اساس این دستورات، رله‌ها و دستگاه‌های متصل فعال یا غیرفعال می‌شوند.** | ( | |
|  |  | |  |
|  |  |  | |

**آیتم های مورد نیاز برای پروژه کنترلر**

**اجزای سخت‌افزاری**

1. \*\*آردوینو\*\*

- Arduino UNO (×1)

- Arduino Ethernet Shield 2 (×1)

2. \*\*سنسورها\*\*

- سنسور دما و رطوبت DHT11 (×1)

- سنسور شعله (×1)

- سنسور گاز MQ4 (×1)

- LDR (×1)

- سنسور رطوبت خاک (Hygrometer) (×3)

3. \*\*ریله‌ها و بردها\*\*

- ریله 2 کاناله (×4)

- برد بورد کوچک (×1)

4. \*\*کابل‌ها و اتصالات\*\*

- کابل روتر ADSL (×1)

- دوشاخه برق (یک طرفه) (×7)

- پریز برق (یک طرفه) (×7)

5. \*\*منابع تغذیه\*\*

- باتری 9 ولت (×1)

- کابل باتری بشکه‌ای 9 ولت (×1)

**نرم‌افزارها و خدمات آنلاین**

1. \*\*سرور و ویرایشگر\*\*

- سرور HTTP Apache

- Notepad++

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | PAGE 4 |  |

|  |
| --- |
| **نتیجه : پروژه آردینو کنترل با استفاده از آردینو و شبکه اترنت، یک وبسایت حرفه‌ای برای کنترل آکواریوم و سیستم آبیاری خانگی را ایجاد می‌کند. این وبسایت به کاربر امکان می‌دهد که از راه دور اطلاعات خانه‌اش را دریافت کند و دستگاه‌ها را مانند فیلترها و پمپ‌ها از راه دور کنترل کند. با استفاده از تکنولوژی‌های HTML، CSS، JavaScript و PHP، این پروژه یک نمایشگر برای ارتباط با آردینو ارائه می‌دهد. تمامی داده‌ها از طریق LAN ارسال و دریافت می‌شوند و از دستورات و داده‌های سنسورها برای کنترل دقیق تر و مدیریت بهینه خانه استفاده می‌کند**  **لینک پروژ:**  [**https://www.hackster.io/kutluhan-aktar/how-to-create-a-website-communicating-arduino-by-using-php-ce5232**](https://www.hackster.io/kutluhan-aktar/how-to-create-a-website-communicating-arduino-by-using-php-ce5232)  **لینک کوپایلت :**  [**https://copilot.microsoft.com**](https://copilot.microsoft.com)    **لینک گیت هاب :**  [**https://github.com/amirazadi1380/robotic-project**](https://github.com/amirazadi1380/robotic-project) |

