

**Documentatie si raport**

**Proiect PrIOT**

**-Wifi Browser Controlled Robotic Arm-**

Cîrcioroabă Anca-Maria 342C1

***Conținutul Documentului***

1. Introducere
2. Arhitectură

### ****Introducere****

### **Scopul proiectului**

### Acest proiect urmărește realizarea unui braț robotic controlat printr-o interfață web, utilizând un microcontroler Arduino și un modul WiFi ESP8266. Scopul principal este de a demonstra o soluție practică de integrare IoT (Internet of Things) pentru controlul precis al dispozitivelor mecatronice la distanță.

### **Obiective**

* Implementarea unui braț robotic cu 4 servomotoare.
* Dezvoltarea unei interfețe web pentru controlul mișcărilor brațului robotic.
* Integrarea unei soluții de notificare a stării de funcționare și raportarea erorilor.
* Analiza topologiei rețelei și alegerea protocolului de comunicare optim.

### ****Arhitectura****

### **Descriere generala a sistemului**

Sistemul este alcătuit din următoarele componente principale:

1. **Hardware**:
   * Braț robotic cu 4 servomotoare (rotire, înclinare, extensie și control al cleștelui).
   * Microcontroler Arduino UNO pentru controlul mișcărilor servomotoarelor.
   * Modul WiFi ESP8266 utilizat pentru comunicația cu rețeaua.
2. **Software**:
   * Scripturi JavaScript pentru gestionarea comenzilor trimise de utilizator prin interfața web.
   * Cod Arduino pentru procesarea comenzilor și controlul servomotoarelor.
   * Interfață web pentru introducerea și gestionarea comenzilor.
3. **Rețea**:
   * Conexiune WiFi pentru transmiterea comenzilor și recepționarea notificărilor.

### **Fluxul de date**

1. Utilizatorul interacționează cu interfața web pentru a seta unghiurile de rotație ale servomotoarelor.
2. Comenzile sunt transmise prin rețeaua WiFi către ESP8266.
3. ESP8266 transmite comenzile către Arduino pentru procesare.
4. Arduino controlează servomotoarele conform comenzilor primite.
5. Răspunsurile de stare sunt trimise de Arduino către ESP8266 și afișate în interfața web.

### **Topologie utilizata**

Sistemul utilizează o topologie **star** (stea), în care ESP8266 acționează ca un nod intermediar între microcontroler (Arduino) și rețeaua WiFi locală.

**Componentele topologiei:**

1. **Router WiFi** – Punct central de acces pentru rețea.
2. **Modul ESP8266** – Se conectează la rețea ca stație (station mode) și servește ca punte între rețea și Arduino.
3. **Arduino UNO** – Primește date de la ESP8266 prin UART.
4. **Dispozitiv client (PC, tabletă, telefon)** – Trimite comenzi prin interfața web.

### 

### **Alegerea protocoalelor de comunicare**

**Protocoalele utilizate**

1. **HTTP (HyperText Transfer Protocol)**
   * Folosit pentru comunicarea dintre interfața web și modulul ESP8266.
   * Simplu de implementat și suficient pentru acest proiect.
2. **UART (Universal Asynchronous Receiver-Transmitter)**
   * Utilizat pentru transferul de date între ESP8266 și Arduino.
   * Asigură o comunicare serială rapidă și fiabilă.

### ****Implementare****

#### **Pașii de configurare a hardware-ului**

1. **Componente utilizate**:
   * Braț robotic cu 4 servomotoare pentru rotire, înclinare, extensie și controlul cleștelui.
   * Microcontroler Arduino UNO pentru procesarea comenzilor și controlul servomotoarelor.
   * Modul WiFi ESP8266 pentru transmiterea și recepționarea datelor prin rețea.
2. **Conectarea hardware-ului**:
   * Conectați fiecare servomotor la pinii de control PWM ai Arduino-ului (de exemplu, pinii 3, 5, 6, 9 pentru PWM).
   * Modulul ESP8266 este conectat la Arduino printr-o conexiune UART (SoftwareSerial, pinii 10 și 11 pe Arduino).
   * Alimentarea servomotoarelor și a modulului ESP8266 trebuie să fie realizată utilizând o sursă externă, capabilă să furnizeze suficient curent (de exemplu, 5V cu minim 2A).
3. **Testare inițială**:
   * Folosiți un program simplu pe Arduino pentru a verifica funcționarea fiecărui servomotor (de exemplu, rotirea între 0° și 180°).
   * Configurați ESP8266 pentru conexiunea WiFi utilizând comanda AT+CWJAP.

#### **Pașii de configurare a software-ului**

1. **Cod Arduino**:
   * Implementați controlul servomotoarelor în funcție de comenzile recepționate prin ESP8266.
   * Verificați comenzile și trimiteți răspunsuri adecvate (ex. "OK" pentru succes, "ERROR" pentru probleme hardware).
2. **Codul JavaScript**:
   * Realizați interfața web utilizând HTML și JavaScript.
   * Utilizați funcții AJAX pentru a trimite cereri către ESP8266.
   * Asigurați tratarea erorilor și afișarea notificărilor pentru utilizator.
3. **Configurarea ESP8266**:
   * Stabiliți IP-ul static al modulului și configurați-l pentru a funcționa în modul stație (AT+CWMODE=1).
   * Configurați un server HTTP pe ESP8266 pentru a recepționa comenzile trimise din interfața web.

#### **Sistemul de alertare și notificare**

* **Funcționalitate**:
  + Interfața web afișează notificări cu stările actuale, succesul comenzilor și eventualele erori.
  + Mesajele de eroare sunt vizibile în timp real (ex. resetarea hardware-ului detectată de ESP8266).
* **Implementare**:
  + În JavaScript, folosiți funcții precum addNotification pentru a adăuga mesaje în zona de notificări.
  + Adăugați statistici de succes, timeout-uri și resetări utilizând un grafic generat cu Chart.js.

### ****Vizualizarea si procesare de date****

#### **Metoda de procesare a datelor**

1. **Recepția datelor**:
   * Datele sunt transmise de ESP8266 către interfața web sub formă de mesaje HTTP. Fiecare mesaj conține starea curentă (ex. "OK" sau "ERROR").
2. **Procesarea datelor**:
   * În funcție de răspunsurile recepționate, aplicația web procesează stările și actualizează statistici în timp real:
     + Numărul de comenzi trimise cu succes.
     + Numărul de timeout-uri.
     + Numărul de resetări hardware.
3. **Stocarea datelor**:
   * Utilizatorii pot salva comenzile într-un fișier text pentru referință sau încărcare ulterioară.

#### **Afișarea datelor într-o interfață intuitivă**

* **Zona de notificări**:
  + Fiecare comandă și starea asociată sunt afișate imediat într-o zonă de notificări vizibilă pe interfața web.
* **Grafic de statistici**:
  + Generat cu Chart.js, graficul afișează statisticile comenzilor (succese, erori, resetări) într-o manieră vizuală.

### ****Securitate****

#### **Măsuri implementate**

1. **Criptare**:
   * **Nivel HTTPs**:
     + Hostarea fisierului de html pe Github Pages care adauga automat https (SSL security)
2. **Restricționarea accesului**:
   * Dezactivare opțiunea de acces la ESP8266 din afara rețelei locale.