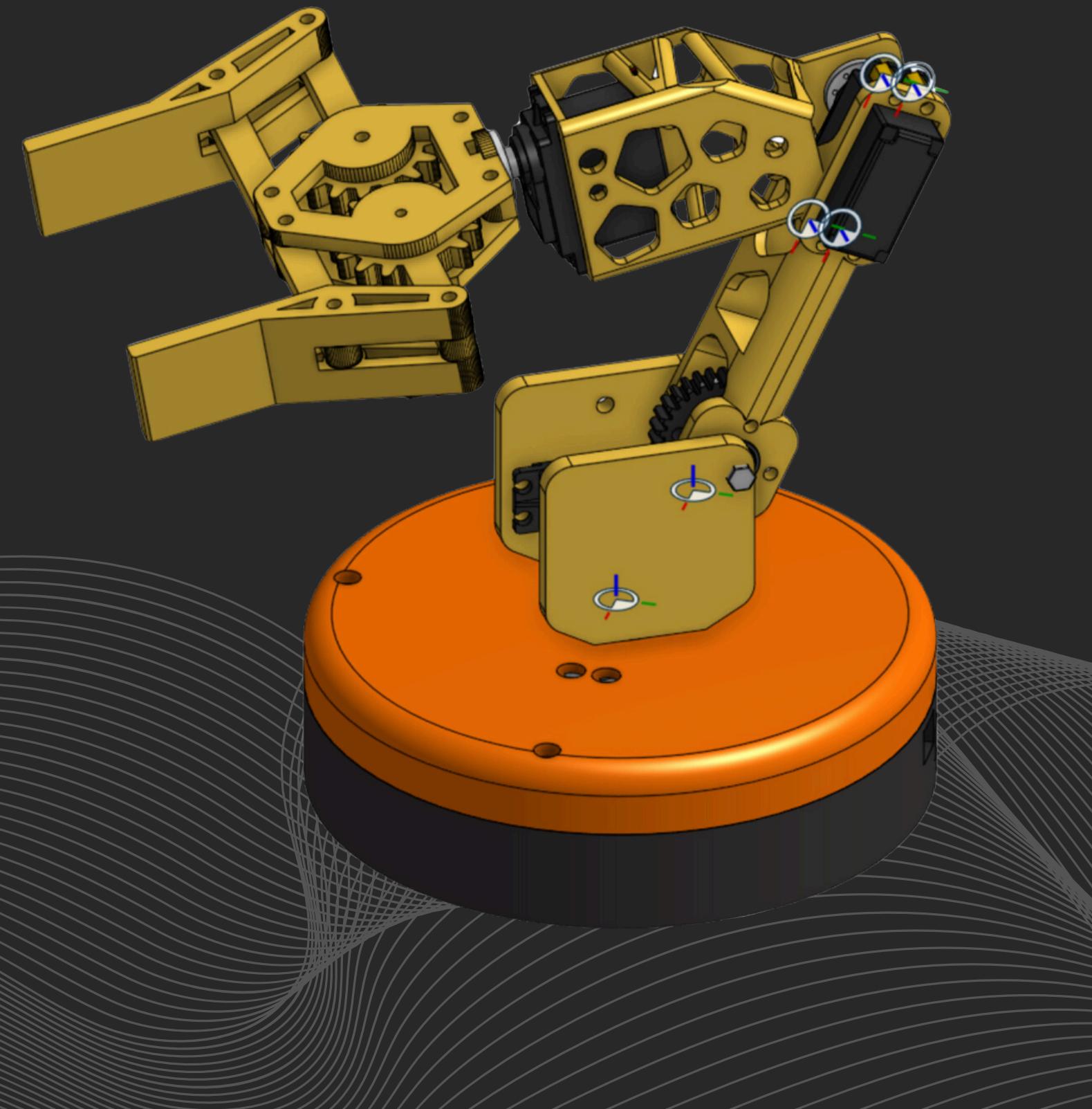


# HANDY-BOT

Un braț robotic menit să-ți facă viața mai ușoară.

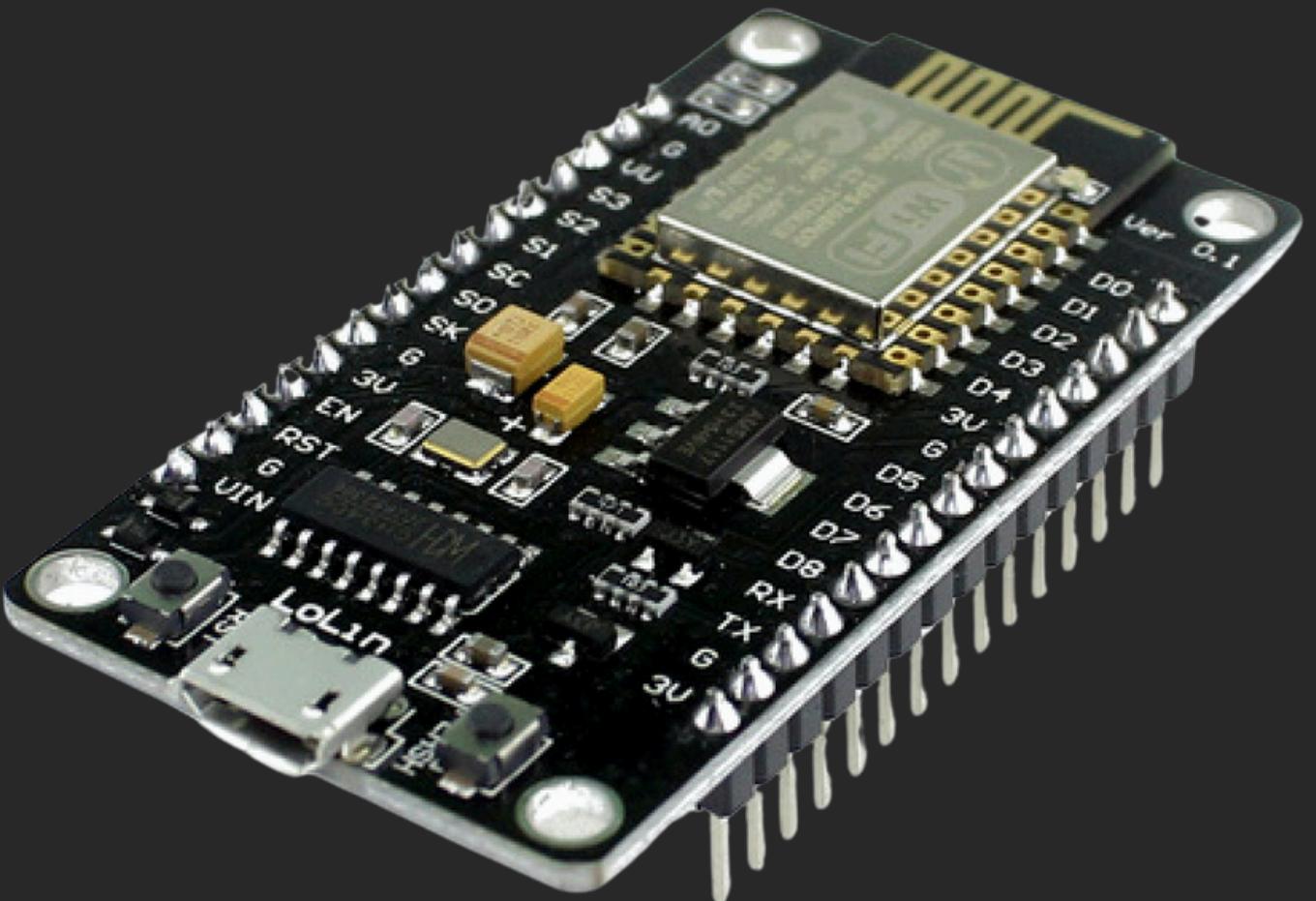
# PREZENTARE GENERALĂ

- Proiectul nostru presupune un brăt robotic, numit HandyBot. Aceasta are rolul de a înlocui forța de lucru umana cu una robotizată care nu obosește niciodată
- Este capabil să efectueze sarcini de bază, cum ar fi prinderea, ridicarea și plasarea obiectelor, manuală sau automatizată.



# ESP8266

- Brațul este controlat prin microcontrolerul ESP8266, care are un microcip Wi-Fi.
- Aceasta permite controlul și automatizarea brațului robotic de la distanță printr-o interfață web



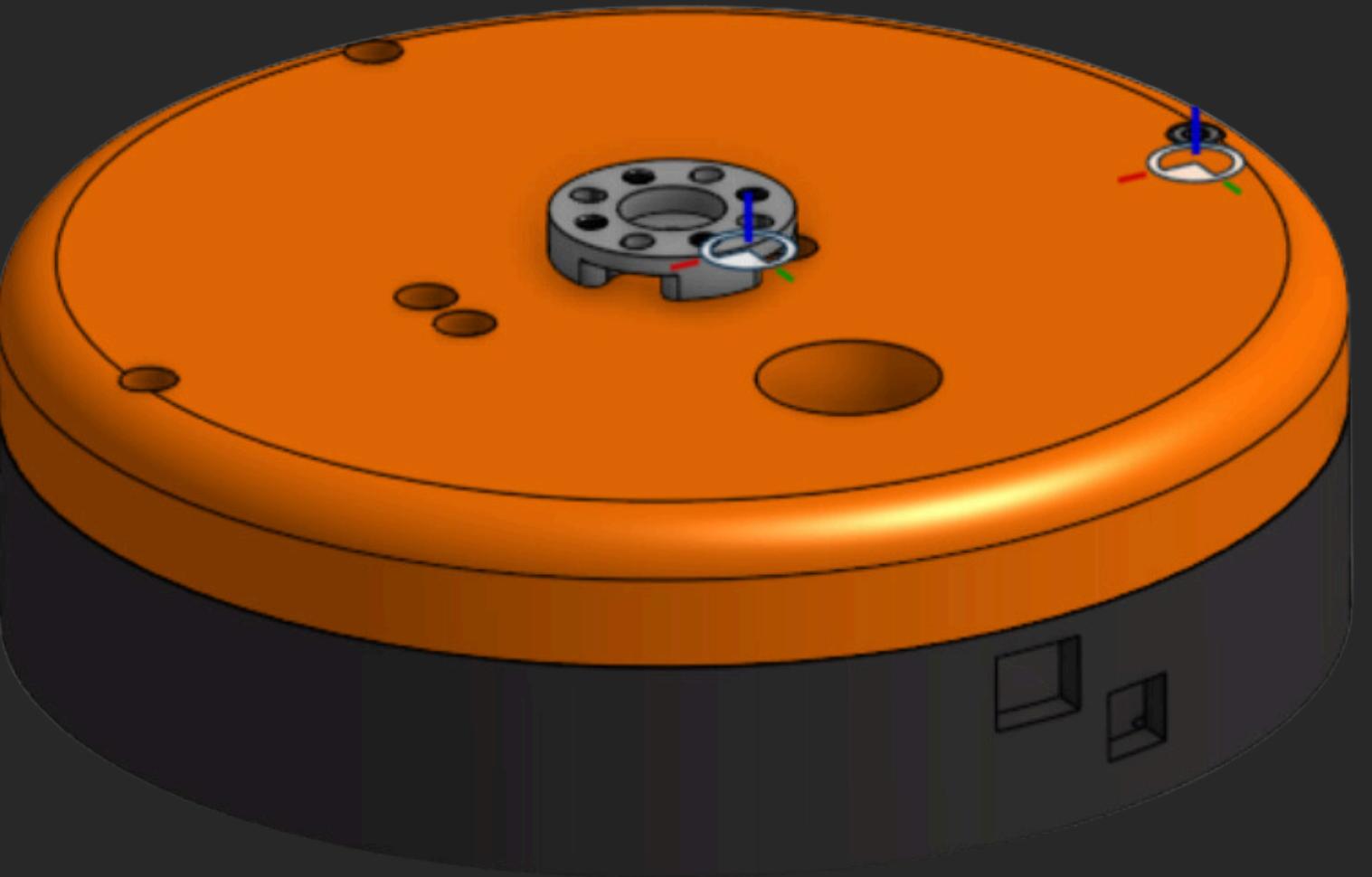
# PROIECTARE 3D

- Proiectarea 3D a proiectului a fost realizată integral în Onshape și poate fi găsită în link-ul atașat în documentație
- Ansamblul brațului se împarte în 3 părți: bază, brațul propriu-zis și gheară

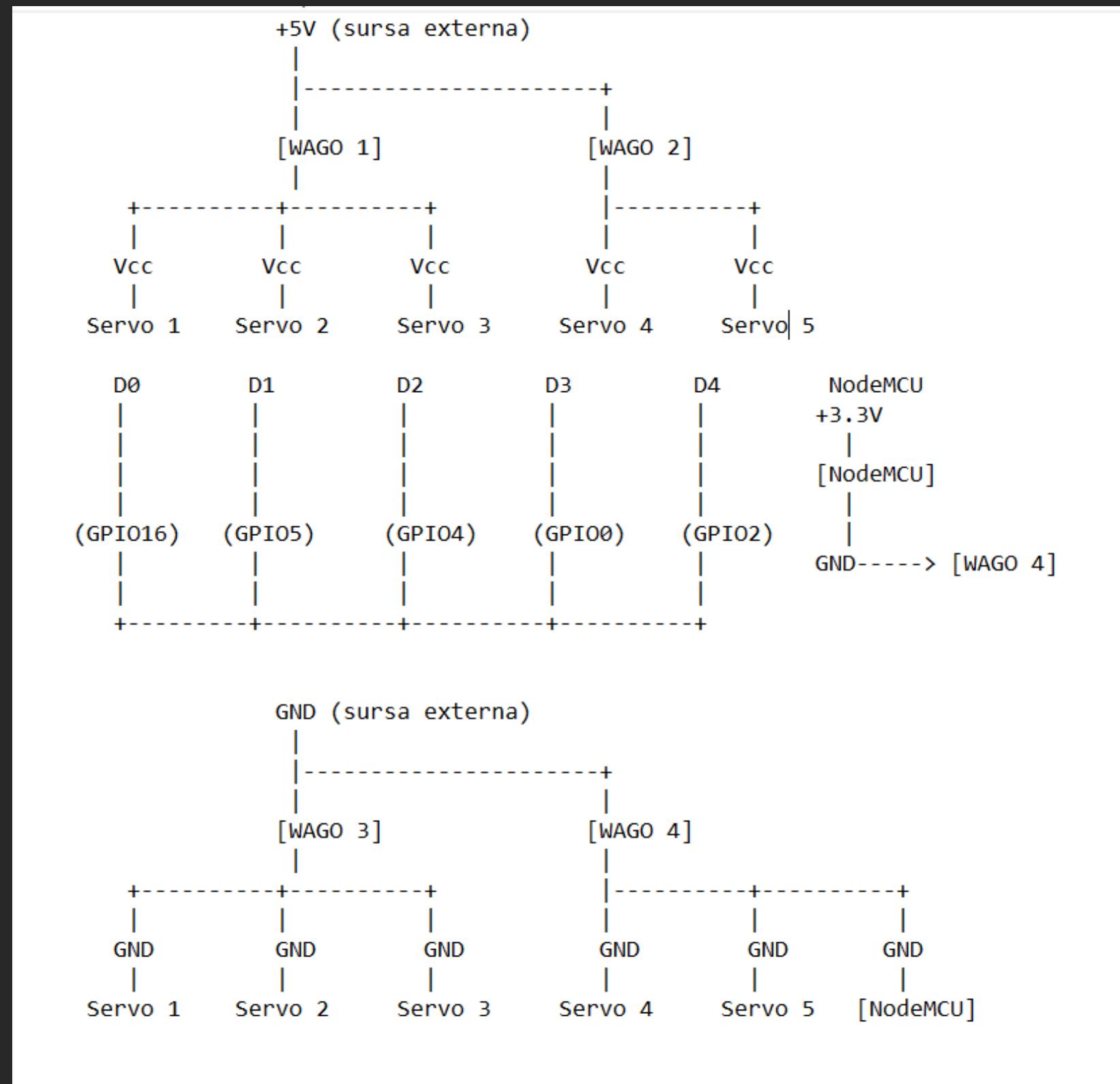


# BAZA

- Aceasta adăpostește ESP-ul 8266 pus pe un breadboard cu 170 de pini și 4 conectoroare Wago cu 5 poli pentru a conecta servo-urile în paralel la sursă
- Mai mult de atât în capac este prins servo-ul de rotație a bazei brațului

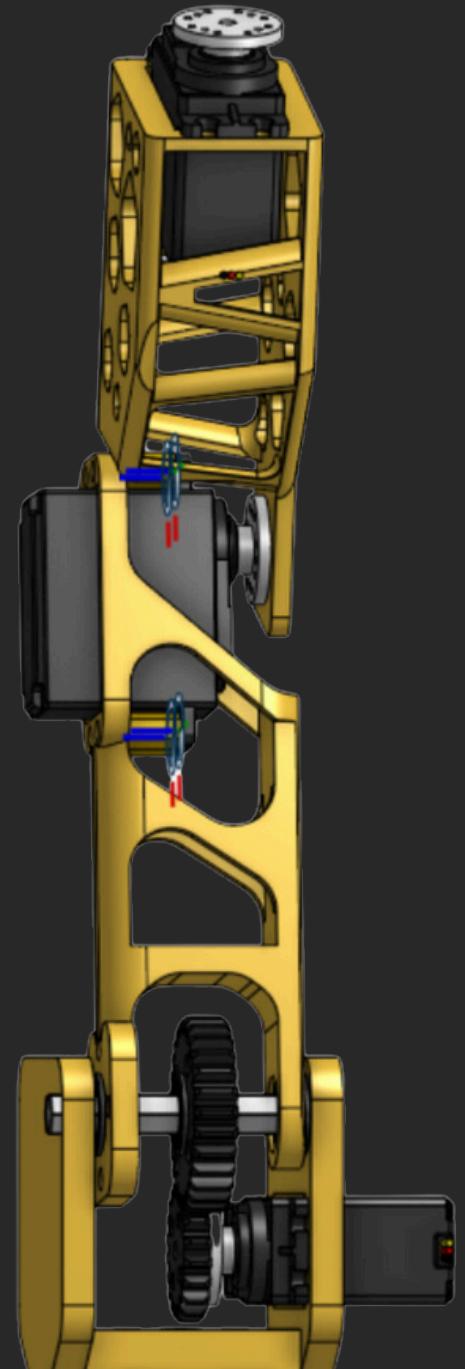


# CIRCUITUL ELECTRIC



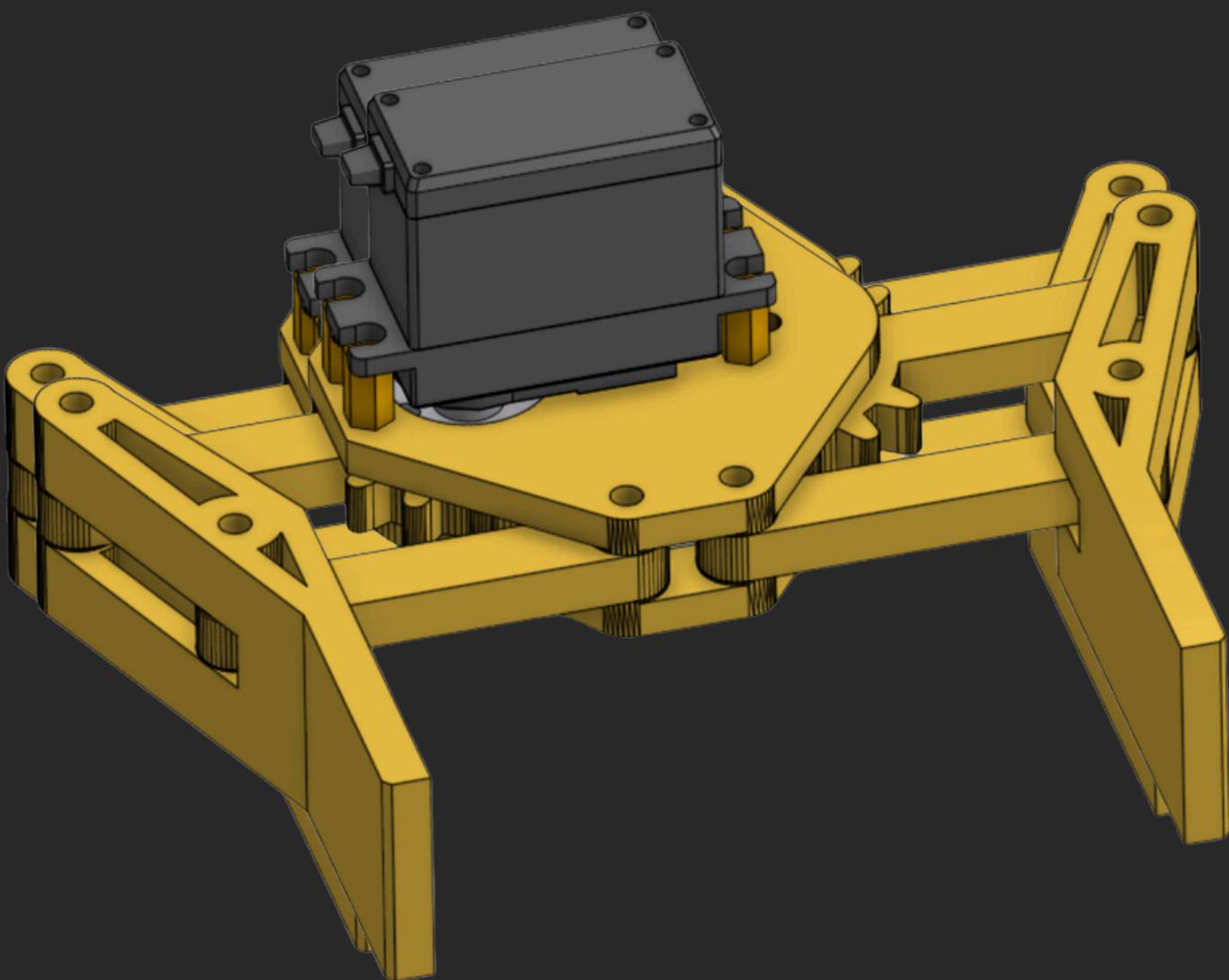
# BRĂȚUL

- Brățul este controlat de 3 motoare servo. Acestea împreună cu cel din bază permit mișcării lui Handy Bot 4 grade de libertate
- La baza brățului am folosit pe servo un raport de transmisie de 28:20 pentru a avea mai mult cuplu



# Gheara

- Gheara este controlată de un sigur servo datorită angrenajelor interne care conferă o mișcare simultană și paralelă
- Aceasta se poate roti la 180 de grade fiind atașată direct pe un motor servo



# <PROGRAMARE>

Programarea a fost făcută în:

- C++ pentru controlarea propriu-zisă a servomotoarelor;
- HTML, CSS, JS pentru interfața web.

# FUNZIONI PRINCIPALI



```
void setup(){
    Serial.begin(9600);

    WiFi.softAP(ssid, password);
    WiFi.softAPConfig(local_ip, gateway, subnet);

    server.on("/", handleRoot);
    server.on("/button", handleButton);
    server.on("/slider", handleSlider);
    server.begin();
    Serial.println("HTTP server started");
    servoMainRotation.attach(D0, 500, 2500);
    servoSecondRotation.attach(D1, 500, 2500);
    servoMainVertical.attach(D3, 500, 2500);
    servoSecondVertical.attach(D4, 500, 2500);
    claw.attach(D2, 500, 2500);
}
```



```
void handleRoot() {
    String s = MAIN_page;
    server.send(200, "text/html", s);
}
```



```
void loop() {
    server.handleClient();
}
```

# FUNCȚIA DE AMORTIZARE



```
void smoothMove(Servo &servo, int start, int end, int steps)
{ float increment = (end - start) / float(steps);
  for (int i = 0; i <= steps; i++) {
    servo.write(start + i * increment);
    delay(15);
  }
}
```

# BUTOANE ȘI SLIDERE



```
void handleSlider() {
    if (server.hasArg("id") && server.hasArg("value") && aut == 1) {
        int sliderId = server.arg("id").toInt();
        int sliderValue = server.arg("value").toInt();

        Serial.print("Slider ID: ");
        Serial.print(sliderId);
        Serial.print(" Value: ");
        Serial.println(sliderValue);

        switch(sliderId){
            case 1: smoothMove(servoMainRotation, turnPos, sliderValue, 40); turnPos = sliderValue; break;
            case 4: servoMainVertical.write(sliderValue); break;
            case 3: smoothMove(servoSecondVertical, secondRotPos, sliderValue, 50); secondRotPos = sliderValue;
break;
            case 2: smoothMove(servoSecondRotation, mainRotPos, sliderValue, 50); mainRotPos = sliderValue;
break;
            default: Serial.print("No such slider.");break;
        }
    }
}
```



```
void handleButton() {
    if (server.hasArg("id")) {
        int buttonId = server.arg("id").toInt();
        Serial.print("Button ID: ");
        Serial.println(buttonId);

        // Perform action based on buttonId here
        switch (buttonId) {
            case 1:
                if(aut == 1) claw.write(0); // CLOSE
                break;
            case 2:
                if(aut == 1) claw.write(180); //
OPEN
                break;
            case 3:
                autonomous();
                break;
            default:
                // Handle invalid button ID
                break;
        }
    }
}
```

# PAGINA WEB

```
<center>
  <h1>Brat Robotic</h1><br>
  <p>Rotire Baza</p>
  <input type="range" min="0" max="180" value="90" onchange="updateSlider(1, this.value)">
  <input type="range" min="0" max="140" value="90" onchange="updateSlider(2, this.value)">
  <input type="range" min="43" max="160" value="90" onchange="updateSlider(3, this.value)">
  <input type="range" min="0" max="180" value="90" onchange="updateSlider(4, this.value)">
  <br>

  <button onclick="sendRequest(1)" class="button">Deschide</button>
  <button onclick="sendRequest(2)" class="button">Inchide</button>
  <button onclick="sendRequest(3)" class="buttonLast">Autonom</button>
</center>
```

```
<script>
  function updateSlider(sliderId, value) {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/slider?id=" + sliderId + "&value=" + value, true);
    xhr.send();
  }

  function sendRequest(buttonId) {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/button?id=" + buttonId, true);
    xhr.send();
  }
</script>
```

# APLICATIA DE CONTROL

## Braț Robotic

Rotire Bază



Ridicare Braț 1



Rizicare Braț 2



Rotire Gheără



Deschide

Inchide

Autonom

# PROBLEME CE POT APĂREA

- Interferențe cu rețeaua folosită pentru conectarea la braț;
- Defectarea angrenajelor din cauza uzurii în timp;
- Stagnarea brațului din cauza unei coliziuni nedorite cu un corp sau o persoană.

# ÎMBUNĂTĂȚIRI VIITOARE

- Printarea pieselor cu un filament mai puternic
- Adăugarea de senzori pentru recunoașterea obiectelor
- Optimizarea interfeței pentru o controlare mai ușoară și fluidă
- Un servo motor mai puternic pentru gheară
- Adăugarea încă unui unghi de libertate

MULȚUMIM PENTRU ATENȚIE!