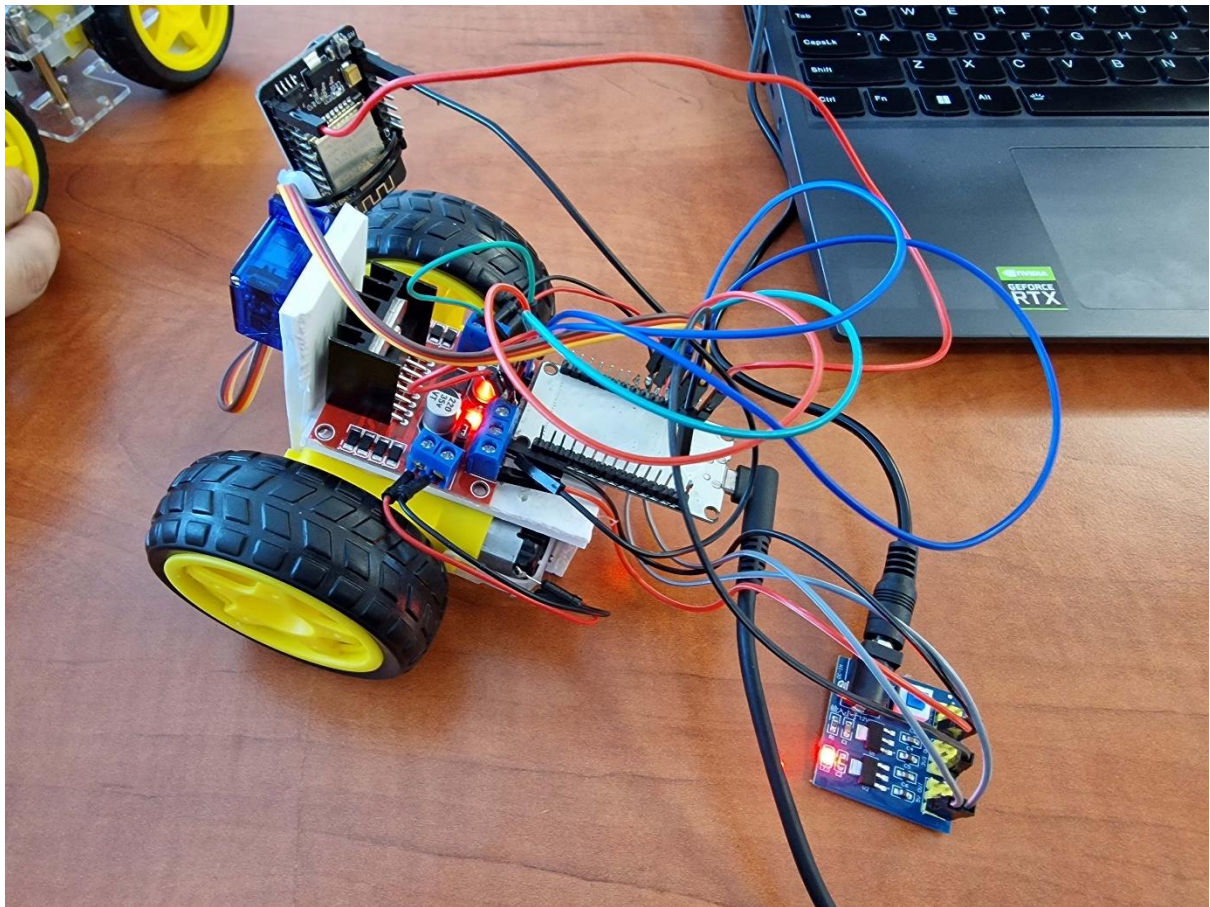


Project – walking robot with attached camera

Made by Olaru Constantin-Alexandru



The robot from this project is built to test the power of 6V DC engines, to test the power of a development board made by Espressif and what I could do with a camera. The links attached in this file are links from the online stores where I bought the components.

For the static part of the robot I used some PVC planks on which the components were glued, except the power supply module that got overheated and melted the glue.

The main board, an [ESP32 DevKIT V1](#), is where the main code was loaded, which I will explain later. Next to it, I attached a [engine driver module L298N](#) that drives the [DC engines](#). In the project, the speed of the engines could not be adjusted as I just wanted to make it move. Later, I regretted doing that as it would become unstable.

The mobile part of the robot consists of the two DC engines and a [mini servomotor SG90](#). The servo was used for moving the camera up and down.

The camera used is an [ESP32-CAM that came with the programmer](#) and the code for it will not be attached to this file because it's the default one from Espressif, and I just needed to select the camera from the defines.

The [power supply module](#) provides 3.3V, 5V and 12V. Initially, I used it for powering the whole project, but the power draw was too high for the module so I used it just for the common ground and the DC engines. The main board is powered through the microUSB port, and the camera is powered from the main board.

Now, let's do an analysis throughout the code.

I will just explain the important parts, and not the connected pins or the server initialization because the pins are selected based on the characteristics. I selected 4 pins that can output PWM signal for the connection with the engine driver module, the IP from the camera server was extracted by starting the camera and observing the logs.

The robot can move forward, backward, turn left, turn right or stop based on the functions `stopMotors`, `moveForward`, `moveBackward`, `turnLeft`, `turnRight`. Because the module is built with a H bridge, I just understood that in order to make it work I needed to use only one HIGH signal for each engine, or else I could burn it.

For this part of the code:

```
void setFaceDetection(bool enable) {  
    HTTPClient http;  
  
    String url = "http://" + String(camServerIP) +  
    "/control?var=face_detect&val=" + String(enable ? "1" : "0");  
  
    http.begin(url);  
  
    int httpCode = http.GET();  
  
    if (httpCode > 0) {  
        String payload = http.getString();  
        Serial.println(payload);  
    } else {  
        Serial.printf("Error on HTTP request: %s\n",  
            http.errorToString(httpCode).c_str());  
    }  
  
    http.end();  
}
```

The camera code has a specific part where it can detect faces. When the “Face detection” mode is enabled on the main server it will also enable it on the camera server.

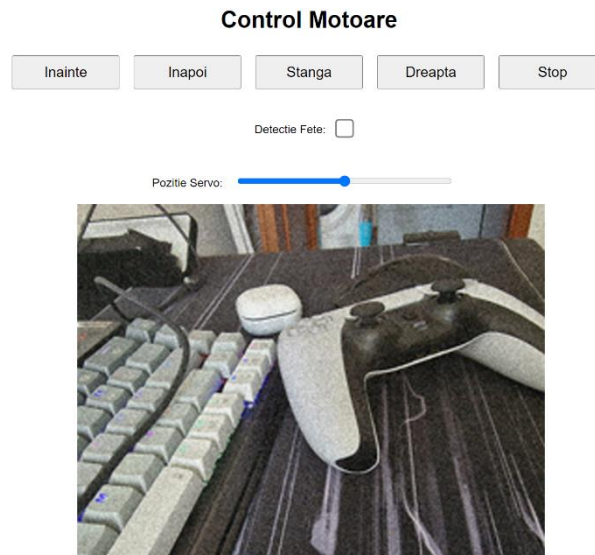
The setServoPosition adjusts the position of the head of the servo.

In the setupWebServer is the design of the online server built with HTML, CSS and some JavaScript. That was not entirely made by me, but I made the logic and the design in the web page.

The body of the setup function is adapted to show me the IP of the server in the logs.

As a conclusion and testing, I will say it works ok, the part with the adjustment of the speed would make it better and a better camera because the one I used is hard to attach.

The server looks like this:



In the photo you can see the movement buttons, face detection enabler, servo position slider and what you can see on the camera.