RC Surveillance Car

by George-Codrin Preda (1221B)

Introduction

Résumé: The project is inspired by the Curiosity Mars rover and the growing need for remote-controlled surveillance devices. From bomb disposal robots to Mars rovers, we need robots that we can send on risky missions instead of humans. A robot does not need food or sleep and it can do many things that a human cannot: it can withstand harsh conditions such as extreme temperatures or high levels of radiation.

General description

The car is built using a robot car kit with 4 motors,4 wheels and connectors, an ESP32 cam module, an L298N motor driver module, 8×1.5V batteries, and the supports for the batteries, double-sided tape, jumper wires, and female to male and female to female wires. After assembling everything according to the diagram below, I took the ESP32 cam module out and assembled a circuit for the code upload using a breadboard, an Arduino Uno and a few jumper wires, the diagram for the code upload is below, in the Hardware Design section. After reconnecting the ESP32 cam module, I connected the batteries to the car. The car captures video footage using the camera present in the car and it sends the footage to my mobile phone using WebSocket through a Wi-Fi connection. The mobile application created can adjust the speed of the car and the light of the flashlight present on the cam module.

Block scheme:

Hardware Design

Components list:

- 4WD car kit
- ESP32 cam mod
- L298 motor driver mod
- 8×1.5V batteries
- 2xbattery supports
- breadboard
- Arduino Uno
- double-sided tape
- jumper wires, female to female wires, female to male wires

Electric scheme:

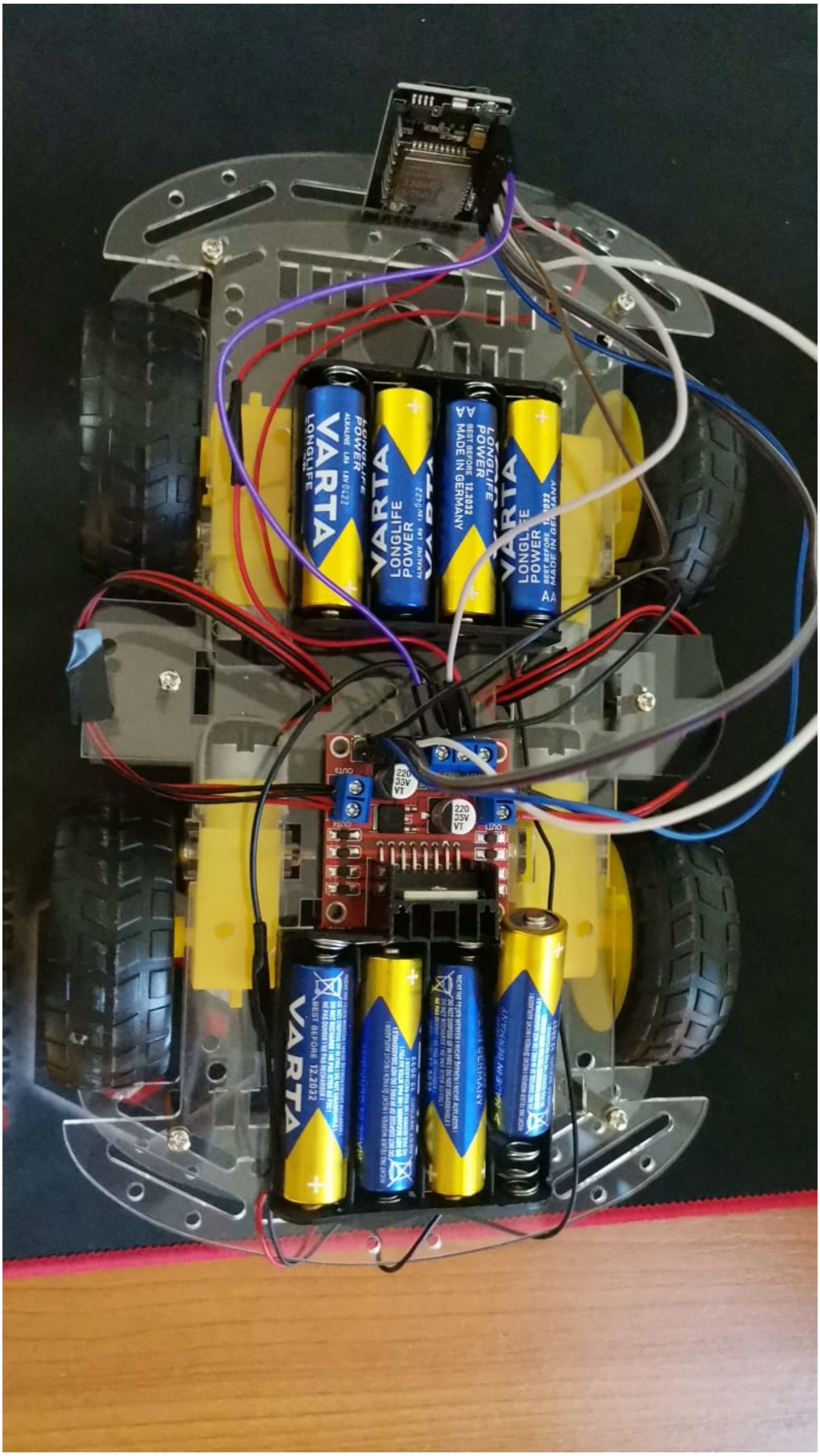
The Car

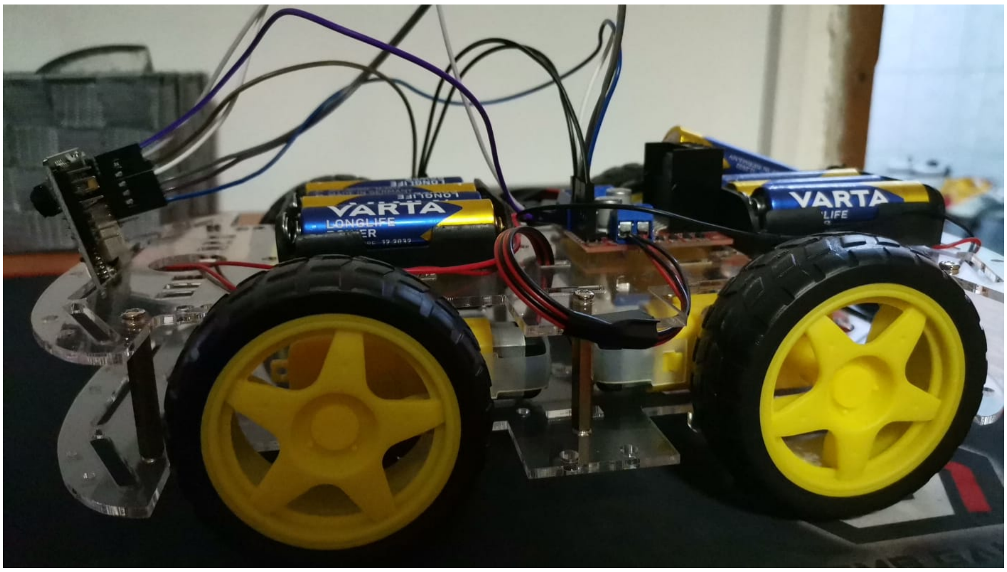
The circuit required for the code upload on the ESP32 cam module

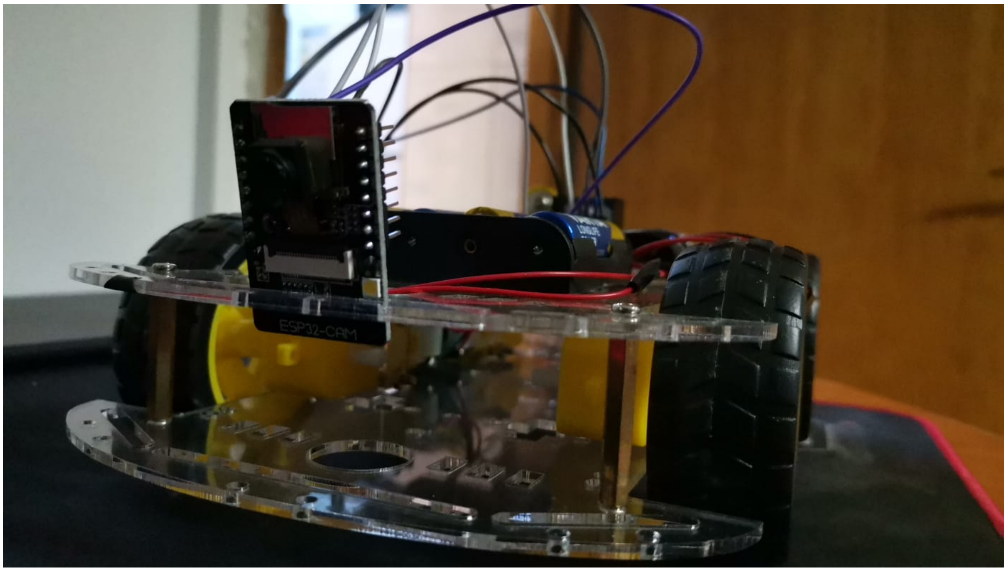
Software Design

Firstly, the ESP32 board was installed using Arduino Board Manager and the AsyncTCP library was added. In the AsyncWebSocket library folder the I set the WS_MAX_QUEUED_MESSAGES was set to 1 for a smooth stream. In the code, I included the AsyncTCP and ESPAsyncWebServer libraries and I assigned the left and right motor pins. The light pin was also defined and also the channels for speed and light. I added camera-related constants and set a name for the ESP32 Wi-Fi and a password. WebSocket is used for camera and car input control. Then, I created an HTML page for the camera control app. The rotateMotor function takes motorNumber and motorDirection as arguments as it will set the car in forward or backward direction. The moveCar function takes the cases and assigns the movement direction to the rotateMotor function and it moves the car accordingly. The setupCamera function enables the camera. The sendCameraPicture captures the image and sends it to the camera WebSocket client.

Results







Most importantly, the car works and runs smoothly. I can follow its stream on my mobile phone and I can adjust the speed and the light. Quickly tapping the light adjusting button makes it flash its headlight just like a car.

Conclusions

The car moves faster than expected and setting the cam module in a position where it has more visibility was tricky. The car-phone connection works as long as I am connected to the wifi connection the ESP32 cam module provides so that I can control the camera from quite a decent distance. Still, more testing needs to be done regarding the maximum distance I can control the car from. The car can run over small obstacles but its nature prevents it from being further tested in an off-road manner. The ESP32 module streams smoothly and the quality of the video is better than the moon landing footage I expected.

Download

https://drive.google.com/drive/folders/1IWhhHo9ssKpGtshJXIPaGPGebp8l0fNb?usp=sharing

https://github.com/codrin-preda/PM-Project-2022/tree/main/PM%20Project%202022

Jurnal and Guide

Modus operandi:

1. assembly car
2. take ESP32 module out
3. connect it to arduino uno via breadboard in order to upload the code
4. include the libraries mentioned above and set WS_MAX_QUEUED_MESSAGES to 1
5. upload the code
6. connect ESP32 back to the car
7. connect the power source
8. open up phone
9. connect to ESP32 wifi
10. open 192.168.4.1 in the browser
11. enjoy!

Bibliografie/Resurse

https://dev.to/pawel/how-to-build-your-own-rc-car-with-arduino-ardcar-12ei

https://create.arduino.cc/projecthub/shubhamsuresh/how-to-make-a-bluetooth-controlled-rc-car-at-home-521212

https://www.instructables.com/Smartphone-Controlled-RC-Car-Using-Arduino/

https://techatronic.com/how-to-make-arduino-bluetooth-rc-car/

https://create.arduino.cc/projecthub/muhammad-aqib/high-speed-arduino-rc-car-5c2a3d

Export to PDF