# Project Alice

## Manual 2024

Version 1.0

*Timo Weide*

17.08.2024

# Licensing Notice

**Creative Commons Attribution-NonCommercial 4.0 International Public License**

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, you are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

https://creativecommons.org/licenses/by-nc/4.0/legalcode

**You are free to:**

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

**Under the following terms:**

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes .

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

**Notices:**

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

For permissions or inquiries regarding commercial use, please contact the author directly.

Weiderobotics@info.com

# Abstract

Humanity continues to transcend its traditional limits through technological innovation, which has rapidly evolved over the past century. While this progress raises ethical questions, it also empowers us to redefine what’s possible. In the field of humanoid robotics, we are witnessing a transformative era, marked by advancements from early models like Honda's ASIMO [1] to contemporary robots such as Teslabot [2] and Boston Dynamics' ATLAS [3]. This progression lays the groundwork for a future shaped by sophisticated robotics.

Project Alice represents a significant step in this evolution, with its first test version documented in this manual. Alice is a humanoid robot designed for remote operation via a Virtual-Reality (VR) Headset, showcasing versatility in applications later ranging from disaster response and hazardous work environments to research and military operations. Powered by an ESP32 microcontroller and costing under 200 Euros, this initial version demonstrates a blend of affordability and functionality.

This manual provides a detailed overview of Alice’s design, technical specifications, and capabilities in its preliminary phase. Future developments and enhancements will build upon the insights gained from this test version, aiming to refine and expand Alice’s capabilities

# Contents

## 1 Introduction

Welcome to the manual for Project Alice, an advanced humanoid robot designed for remote control via a Virtual-Reality (VR) Headset. Alice serves as a versatile avatar capable of performing a diverse range of tasks across various sectors, including disaster response, hazardous environments, scientific research, and military operations.

The first version of Alice is equipped with an ESP32 microcontroller, which serves as the core of its functionality, enabling sophisticated features while maintaining an overall cost under 200 Euros. This document explores Alice's various sensors, actuators, and structural components, highlighting their roles in its performance across different applications.

**Key Features of the first version**

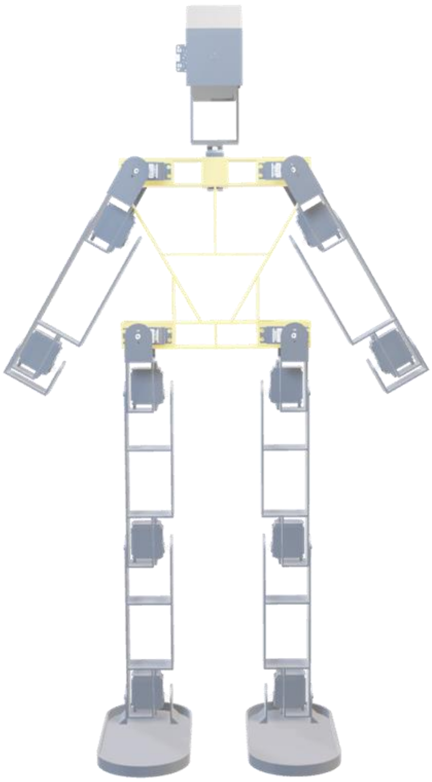
- **Remote Control via VR-Headset**: Alice can be seamlessly operated using a VR-Headset, providing the user with an immersive control experience.
- **Wide Range of movements**: Automated full head control and video streaming through VR-Headset, arm and leg movement trough joystick.
- **Cost-Effective**: Despite its advanced capabilities, Alice is an affordable solution, with a total cost of under 200 Euros for the first version.
- **Advanced Technology**: Powered by an ESP32 microcontroller, Alice combines efficiency with performance, making it a reliable tool for various high-stakes scenarios.

This document details the initial test version of the Alice humanoid robot, focusing on evaluating its core functionalities and design principles. Insights from this preliminary version will guide future enhancements and feature expansions. The aim is to provide a clear understanding of Alice's current state and potential for future development.

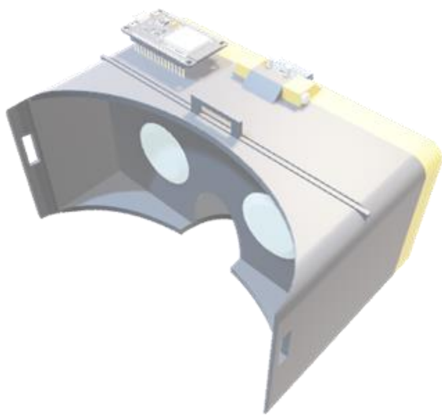
## 2 Overview

The project features a dual-component system: the humanoid robot Alice and a Virtual-Reality-Headset. These components interact seamlessly through a combination of HTTP and ESP-NOW protocols, enabling efficient data exchange between them. The VR headset enables precise control of head movements and video streaming. Additionally, a joystick allows for the control and movement of the arms and legs.

| | |
|---|---|
| Height | 82 cm |
| Weight | ~2000 g |
| Costs | 112 € |
| Print-Time | 24 h 41 min |
| Consumption | ~0,5 A |
| Voltage | 6 V |



| | |
|---|---|
| Dimension LxWxH | 17x12x10 cm |
| Weight | ~250 g |
| Costs | 45 € |
| Print-Time | 8 h |
| Consumption | ~0,2 A |
| Voltage | 5 V |

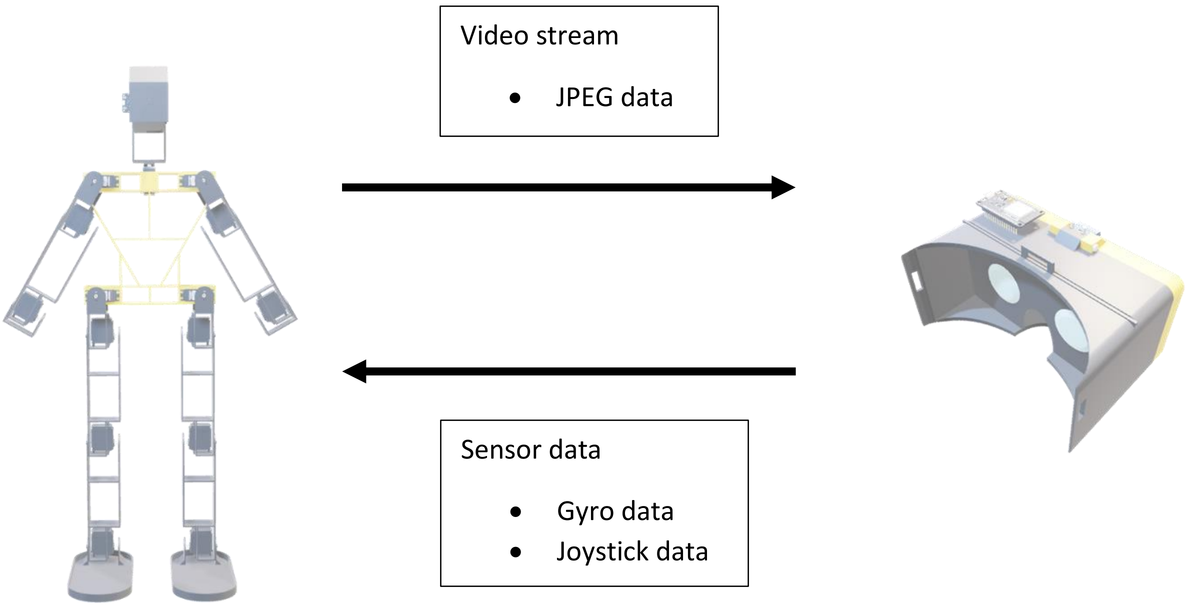


## 3 Setup

The robot Alice will start a WiFi network as soon as it is connected to a power source. This WiFi network serves as the communication hub for the entire system. Once the network is active, the Virtual-Reality-Headset can be connected to its own power source. Upon powering up, the VR-Headset will automatically initiate a connection to Alice’s WiFi network.

When both devices are successfully connected, the system enters normal operation mode. In this mode, the video feed from Alice's camera is streamed directly to the VR-Headset, providing the operator with a real-time view from the robot’s perspective. Simultaneously, sensor data collected by the VR-Headset is transmitted back to Alice, enabling interactive control and feedback.

This seamless interaction between the robot and the VR-Headset allows for an immersive and responsive experience. For a more detailed explanation of the underlying software and how this setup is configured, please refer to the Chapter on Software.



## 4 Hardware

In the following sections, we will delve into the individual components of the Alice humanoid robot and the Virtual-Reality-Headset, examining each segment in detail. Understanding the hardware is crucial to grasping how Alice operates and achieves its versatile capabilities. This section will provide a comprehensive overview of the key hardware elements, including their functions, specifications, and the role they play in the overall system.

By thoroughly understanding the hardware components, you will gain insights into the innovative design and engineering behind Alice, paving the way for potential enhancements and applications in the field of humanoid robotics.

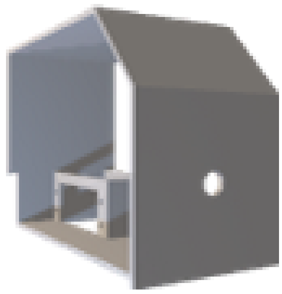
A comprehensive list of all components and materials can be found at the end of this manual.

### 4.1 3D parts

First, we take a close look to the fully printable hull of the humanoid Robot Alice. The parts were printed on an Anycubic Neo 2 with 1,75mm black PLA filament. For the first version all parts were designed openly to show the prototype status. Every part is underlined with several pictures and explanation.

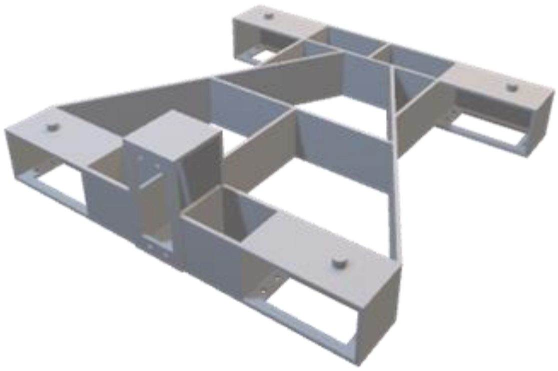
#### 4.1.1 Head and neck

The head of the humanoid robot Alice is a crucial component as it houses the “brain” of the robot, which is the ESP32 microcontroller, along with the camera. The design is intentionally kept as a single piece to simplify the assembly process and ensure robustness during testing. The head serves as the central hub for controlling the robot's various functions. The design of the head allows for easy integration of additional sensors if required in the future, making it adaptable for further upgrades. The neck part connects the head to the torso and allows the correct head movements.



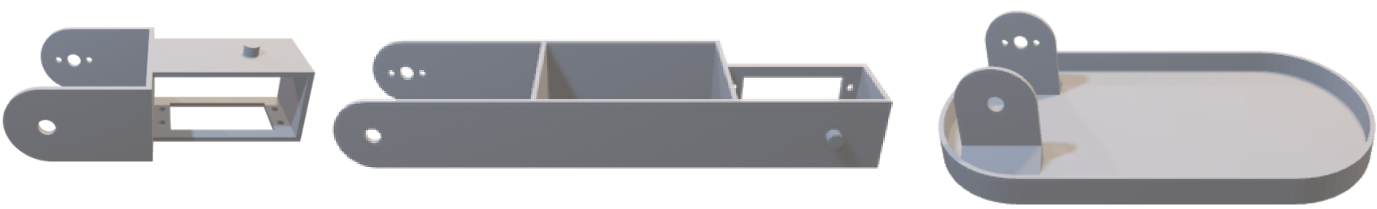
#### 4.1.2 Torso

The torso of Alice is designed as a single piece, emphasizing both simplicity in assembly and structural integrity. This one-piece design reduces the number of components needed, which minimizes potential points of failure and enhances the overall durability of the robot. The torso serves as the central framework that connects the head, arms, and legs. It also acts as a conduit for the internal wiring, allowing cables to be neatly routed from the power source and central processor (located in the head) to the other parts of the robot. This design not only keeps the internal components organized but also provides ample space for additional hardware such as batteries, sensors, or future upgrades.



#### 4.1.3 Leg

The legs of Alice are composed of two distinct sections: the leg part and the hip part. These components are specifically designed to work seamlessly with the servomotors, ensuring that the robot maintains balance and stability during movement. The modular design allows for precise adjustments and replacements if necessary. The leg sections are built to support the weight of the robot while providing flexibility and strength, which are crucial for tasks that require movement or interaction with the environment. The design also considers the placement of the servomotors in such a way that they can deliver the necessary torque without compromising the robot's overall aesthetic and functionality.

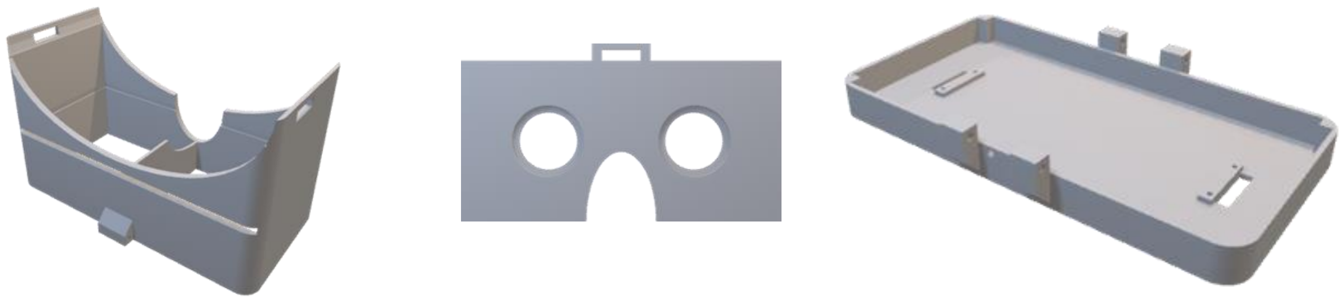


### 4.1.4 Arm

The arm design of Alice mirrors the leg design, utilizing similar components to maintain consistency and reduce the complexity of manufacturing. This approach not only simplifies the production process but also ensures that parts are interchangeable, reducing the need for unique components. In addition to the basic structure, a smaller manipulator is attached to the arm, which serves as a hand. This manipulator is designed to perform basic tasks such as grasping objects, allowing Alice to interact with its environment. The modular design of the arm and hand ensures that additional functionality can be added, such as more sophisticated grippers or tools, depending on the requirements of the task at hand.

### 4.1.5 Virtual-Reality-Headset

The VR-Headset for Alice consists of three main parts: the main body, the lens holder, and the display holder. This component is crucial for enabling Alice to interact with virtual environments or to provide an immersive interface for human operators. The main body houses the electronic components, including the display, which projects the virtual environment. The lens holder ensures that the visual output is correctly focused, providing a clear and immersive experience. The display holder securely positions the display within the main body, ensuring stability during operation. The design is both ergonomic and functional, ensuring that the headset can be used for extended periods without discomfort.



### 4.2 Electronics Design

The electronic design of Alice was intentionally kept as simple as possible to ensure ease of repair and enhance overall reliability. Future iterations will feature custom-designed PCBs to replace the current off-the-shelf modules, thereby reducing the amount of cabling and further simplifying the assembly.

The electrical Schematics for both Alice and VR-Headset can be found at the end of this section.

#### 4.2.1 Alice electronical Design

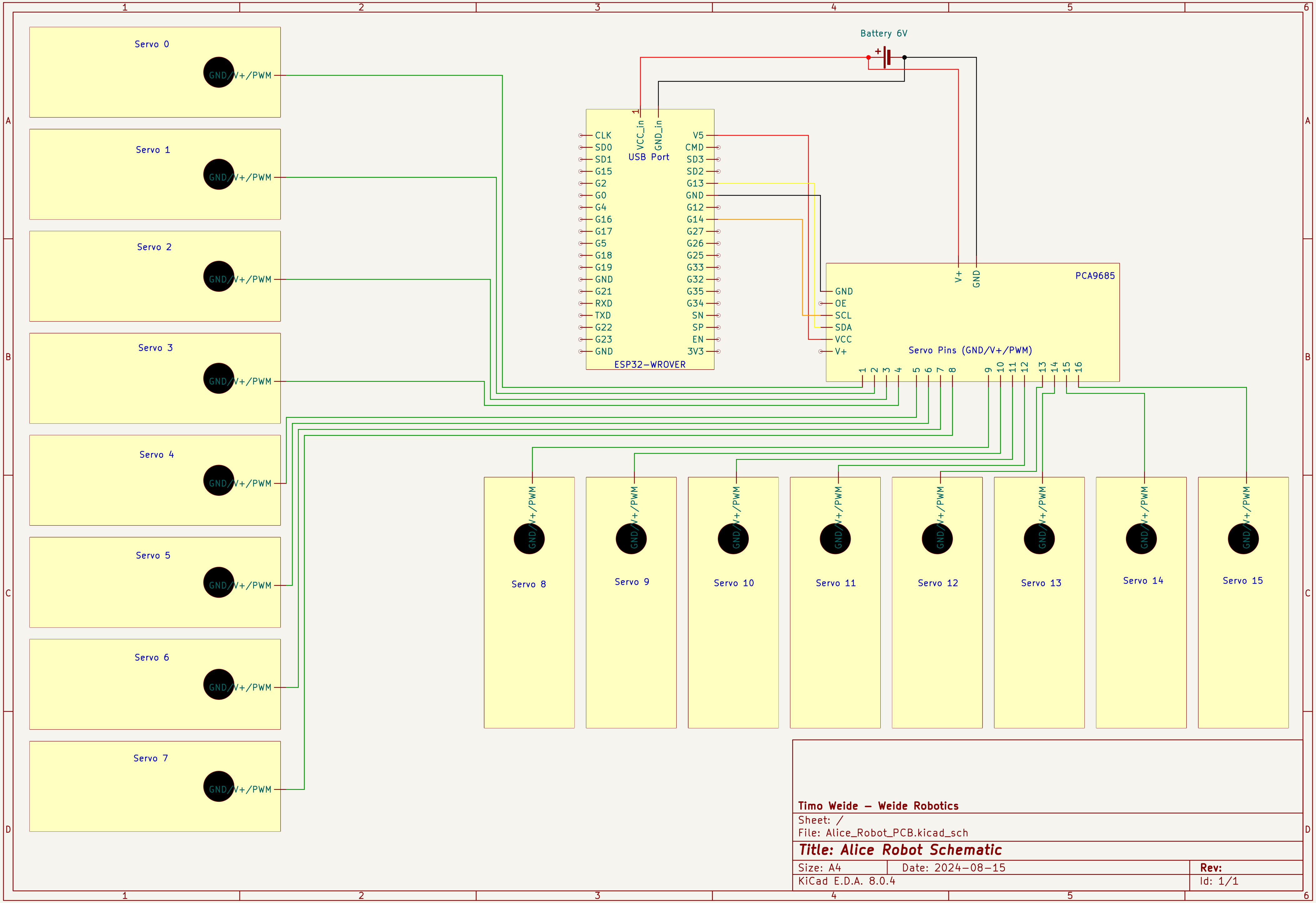
At the core of Alice's electronic system is the ESP32 WROVER Module with a built-in camera, which serves as the "brain" of the humanoid robot. This powerful microcontroller not only manages the robot's operations but also comes equipped with a built-in camera, enabling real-time video processing and communication. The ESP32 communicates with a PCA9685 module, which is responsible for controlling the DM996 servomotors via PWM signals. The various components are interconnected using male-to-female (MxF) and female-to-female (FxF) jumper cables, ensuring flexible and secure connections. Power for the entire system is supplied by a 6-volt lead-acid battery, which provides the necessary voltage and current to sustain the robot's operations.

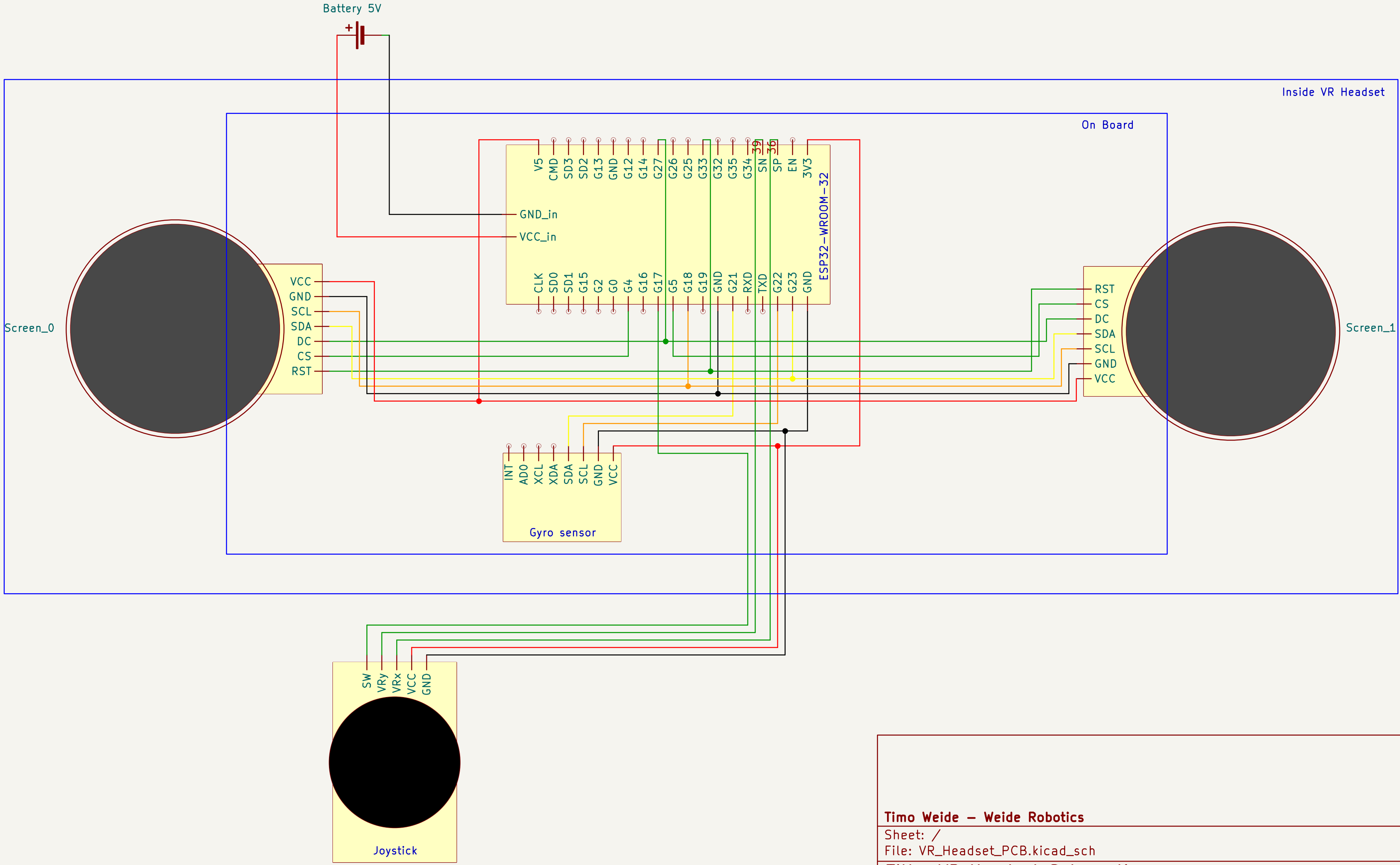
#### 4.2.2 VR-Headset electronical Design

The VR-Headset is powered by an ESP32 WROOM Module, which orchestrates the functionality of the headset. The headset includes two 240x240 TFT displays, one for each eye, providing a clear and immersive visual experience. Additionally, it incorporates a GY-521 MPU-6050 3-axis gyroscope and accelerometer, which tracks the head's position and movements. This data is crucial for the interactive control of Alice, allowing the robot to respond accurately to the operator's head movements. Equipped with a joystick, the robot is capable of performing a range of tasks, including walking and arm movements. Future iterations will enhance the VR-Headset with a Computer Brain Interface System, allowing for joystick-free control through brainwave interactions.

### 4.3 Other parts

In addition to the electrical and 3D printed parts, there are several other essential components that ensure the proper functioning of Project Alice. These include M3x12mm screws and M3 hex nuts, both of which are electro-galvanized to prevent corrosion and provide strong, reliable connections. These small but critical elements are vital for assembling and maintaining the integrity of the robot.


Battery 6V
Servo 0
Servo 1
Servo 2
Servo 3
Servo 4
Servo 5
Servo 6
Servo 7
Servo 8
Servo 9
Servo 10
Servo 11
Servo 12
Servo 13
Servo 14
Servo 15
GND/V+/PWM
VCC_in
GND_in
USB Port
CLK SD0 SD1 G15 G2 G0 G4 G16 G17 G5 G18 G19 GND G21 RXD TXD G22 G23 GND
V5 CMD SD3 SD2 G13 GND G12 G14 G27 G26 G25 G33 G32 G35 G34 SN SP EN 3V3
ESP32-WROVER
PCA9685
V+
GND
GND
OE
SCL
SDA
VCC
V+
Servo Pins (GND/V+/PWM)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Timo Weide - Weide Robotics
Sheet: /
File: Alice_Robot_PCB.kicad_sch
Title: Alice Robot Schematic
Size: A4
Date: 2024-08-15
Rev:
KiCad E.D.A. 8.0.4
Id: 1/1




Timo Weide – Weide Robotics

Sheet: /

File: VR_Headset_PCB.kicad_sch

**Title: VR Headset Schematic**

| | | |
|---|---|---|
| Size: A4 | Date: 2024-08-15 | Rev: |
| KiCad E.D.A. 8.0.4 | | Id: 1/1 |

# 5 Software

The Software of both the humanoid robot Alice and the VR-Headset is written in Arduino INO with the programming language C++. Several Libraries are necessary to make the Code works with its different hardware components. In the following section I will display the Communication Setup on both devices as an example.

The full codes can be found in GitHub at https://github.com/WeideRobotics/Project-Alice

## 5.1 Code Alice

Example Code – The Communication Setup on the humanoid robot Alice

```
// Communication setup
  WiFi.mode(WIFI_AP_STA);
  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return; }
  esp_now_register_recv_cb(esp_now_recv_cb_t(OnDataRecv));
  server.on("/picture", HTTP_GET, [](AsyncWebServerRequest *request){
    camera_fb_t* fb = capturePicture();
    if (fb) {
      request->send_P(200, "application/octet-stream", (const uint8_t*)fb->buf, fb-
>len);
      esp_camera_fb_return(fb); }
      }); server.begin();
```

This setup allows the robot to communicate with other devices via ESP-NOW and serve images over WiFi through a simple web interface.

### 5.2 Code VR-Headset

Example Code – The Communication Setup on the VR-Headset

```
// Communication setup
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print("."); }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return; }
  memcpy(peerInfo.peer_addr, broadcastAddress, 6);
  peerInfo.encrypt = false;
  if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return; }
```

This setup allows the VR-Headset to connect to a WiFi network and establish communication with a specific peer using the ESP-NOW protocol.

### 5.3 Libraries

Various code libraries are necessary to ensure that the code runs smoothly. Below, I will discuss the libraries used and their functions.

<TFT_eSPI.h> This library is used to drive TFT (Thin-Film Transistor) displays with the ESP32 microcontroller. It provides functions for drawing graphics, text, and handling touch input if your display supports it. [4]

<WiFi.h> This library provides the necessary functions to connect the ESP32 to a Wi-Fi network. It handles Wi-Fi connection management, including connecting, disconnecting, and scanning for networks. [5]

<HTTPClient.h> This library simplifies making HTTP requests (GET, POST, etc.) over the internet. It's used to interact with web servers, APIs, and other network services. [6]

<esp_now.h> This library facilitates ESP-NOW, a protocol developed by Espressif for low-power, peer-to-peer communication between ESP32 devices. It allows for fast and efficient communication without the need for Wi-Fi infrastructure. [7]

<esp_camera.h> This library is used for interfacing with the camera module of the ESP32. It provides functions for capturing images and handling camera settings. [8]

<Wire.h> This is the standard library for I2C communication in Arduino and compatible platforms. It allows the ESP32 to communicate with I2C devices like sensors and EEPROMs. [9]

<Adafruit_PWMServoDriver.h> This library controls PWM (Pulse Width Modulation) servo motors using the Adafruit 16-channel PWM driver board. It simplifies driving multiple servos with precise control. [10]

<ESPAsyncWebServer.h> This library provides an asynchronous web server for the ESP32, allowing it to handle HTTP requests and serve web pages without blocking the main program execution. [11]

## 6 Bibliography

**License**

https://creativecommons.org/licenses/by-nc/4.0/legalcode

**References**

[1] https://en.wikipedia.org/wiki/ASIMO

[2] https://en.wikipedia.org/wiki/Optimus_(robot)

[3] https://bostondynamics.com/atlas/

**Code Libraries**

[4] https://github.com/Bodmer/TFT_eSPI

[5] https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi

[6] https://github.com/espressif/arduino-esp32/tree/master/libraries/HTTPClient

[7] https://github.com/espressif/esp-now

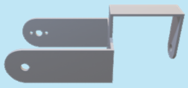
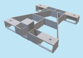
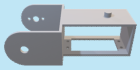
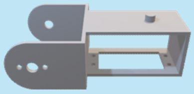
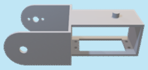
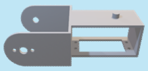
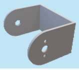
[8] https://github.com/espressif/esp32-camera

[9] https://github.com/arduino/ArduinoCore-avr/tree/master/libraries/Wire

[10] https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library

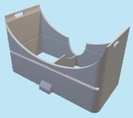
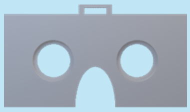
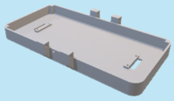
[11] https://github.com/me-no-dev/ESPAsyncWebServer/tree/master

# Robot Alice

| Name | Image | Quantity | Price (total) | Print-Time (total) | Supplier |
|---|---|---|---|---|---|
| | | Body Parts | | | |
| head_part |  | 1 | 0,84 € | 02:00:00 | |
| neck_part |  | 1 | 0,22 € | 00:41:00 | |
| torso_part |  | 1 | 2,10 € | 05:42:00 | |
| left_hip_part |  | 1 | 0,27 € | 00:45:00 | |
| right_hip_part |  | 1 | 0,27 € | 00:45:00 | |
| leg_part |  | 6 | 3,42 € | 08:00:00 | Self printed with Anycubic NEO 2 3D Printer and 1.75 PLA black filament |
| left_foot_part |  | 1 | 0,51 € | 01:58:00 | |
| right_foot_part |  | 1 | 0,51 € | 01:58:00 | |
| left_arm_part |  | 1 | 0,27 € | 00:45:00 | |
| right_arm_part |  | 1 | 0,27 € | 00:45:00 | |
| hand_manipulator_part |  | 2 | 0,44 € | 01:22:00 | |
| | | Electronics | | | |
| Servomotor_DM996 | | 16 | 77,31 € | | diymore-Store |
| ESP32_Cam_Dev | | 1 | 9,49 € | | Heemol-Store |
| PCA9685_Modul | | 1 | 4,66 € | | diymore-Store |
| ELEGOO Jumper Wire 20cm FF | | 4 | 0,20 € | | ELEGOO-Store |
| ELEGOO Jumper Wire 20cm MF | | 28 | 1,68 € | | ELEGOO-Store |
| Lead-Acid-Akku | | 1 | 7,50 € | | Pollin.de |
| | | Miscellaneous | | | |
| Screws_M3_20mm | | 72 | 1,64 € | | Schraubenkasten.de |
| Nuts_M3 | | 28 | 0,43 € | | Schraubenkasten.de |
| Total | | 168 | 112,04 € | 24:41:00 | |

26.07.2024

# VR Headset

| Name | Image | Quantity | Price (total) | Print-Time (total) | Supplier |
|---|---|---|---|---|---|
| | | Body Parts | | | |
| main_part |  | 1 | 1,93 € | 05:32:00 | |
| lenses_part |  | 1 | 0,32 € | 00:39:00 | Self printed with Anycubic NEO 2 3D Printer and 1.75 PLA black filament |
| display_part |  | 1 | 0,68 € | 01:46:00 | |
| | | Electronics | | | |
| ESP32_NodeMCU_Dev | | 1 | 8,33 € | | AZDelivery-Store |
| GY-61_Module | | 1 | 6,80 € | | AZDelivery-Store |
| GC9A01_TFT_Display_1.28 | | 2 | 11,33 € | | ARCELI |
| ELEGOO Jumper Wire 20cm MF | | 32 | 1,59 € | | ELEGOO-Store |
| ELEGOO Jumper Wire 20cm FF | | 10 | 0,50 € | | ELEGOO-Store |
| KY-023 Joystick Modul | | 1 | 1,90 € | | AZDelivery-Store |
| | | Miscellaneous | | | |
| Screws_M3_20mm | | 6 | 0,14 € | | Schraubenkasten.de |
| Headband | | 1 | 7,79 € | | Tbest-Store |
| Bikonvex_lenses_37mm | | 2 | 3,99 € | | Durovis |
| Total | | 59 | 45,29 € | 07:57:00 | |

26.07.2024