


# Roboterarm-Tutorial

# Einleitung

**Unser Unternehmen**

ACEBOTT STEM Education Tech Co., Ltd

Unternehmensgeschichte: Gegründet 2013, mit Sitz in Shenzhen, Chinas Silicon Valley. Unser Team besteht aus 150 Mitarbeitern, darunter Experten in Forschung & Entwicklung, Produktion, Vertrieb und Logistik. Unser Ziel ist es, unseren Kunden erstklassige Produkte und Dienstleistungen im Bereich MINT-Bildung anzubieten.

Wir arbeiten mit STEM-Bildungsexperten und Geschäftspartnern weltweit zusammen, um unseren Kunden exzellente STEM-Bildungskits anzubieten. Gleichzeitig bieten wir unseren Kunden OEM-Dienstleistungen an, darunter die Anpassung von Produktverpackungen und PCB-Logos.

**Tutorial**

Dieser Kurs und das Lernset für Roboterarme richten sich an Kinder und Jugendliche ab 8 Jahren und vermitteln ein tieferes Verständnis des ESP32-Entwicklungsboards, von Roboterarmen und elektronischer Hardware. Wenn Sie mehr über Roboterarme erfahren möchten, bietet Ihnen dieses Set das nötige Wissen und die praktischen Schritte, um Ihren eigenen Roboterarm zu bauen.

**Mit diesem Kit können Sie:**

I. Lernen, wie Sie das ESP32-Entwicklungsboard effektiv verwenden, einschließlich des Hochladens von Code, des Kennenlernens seiner Eigenschaften sowie des Programmierens in der Arduino IDE.

II. Auf der Grundlage der C-Sprache eine solide Programmierbasis aufbauen, da der ESP32 eine vereinfachte C/C++-Programmiersprache zur Steuerung von Schaltungen und Sensoren verwendet.

III. Das Funktionsprinzip des Servomotor-Moduls erkunden und verstehen, wie mehrere Servomotoren im Roboterarm-Projekt zusammenarbeiten.

IV. Mithilfe der Anleitung das ACEBOTT-Set Schritt für Schritt zum Aufbau Ihres eigenen Roboterarms verwenden und Ihre Maker-Fähigkeiten verbessern.

V. Im Roboterarm-Projekt grundlegende Funktionen wie Joystick-Steuerung, Teaching, Websteuerung und App-Steuerung realisieren.

VI. Ihr umfassendes Verständnis des Konzepts des Roboterarms verbessern und sich auf weiterführendes Lernen in der Zukunft vorbereiten.

Insgesamt ist der ACEBOTT-Roboterarm ein speziell für Anfänger entwickeltes, auf ESP32 basierendes Lernset. Mit diesem Set können Benutzer die Funktionen des Steuerboards und der Servomotoren im Roboterarm umfassend verstehen. Durch die Arbeit mit den im Set enthaltenen Anleitungen können Schüler verschiedener Altersgruppen wertvolles Wissen über Roboterarme erwerben und ihr eigenes Roboterarm-Projekt erfolgreich aufbauen.

**Kundendienst**

ACEBOTT ist ein dynamisches und schnell wachsendes Unternehmen im Bereich STEM-Bildungstechnologie, das sich der Bereitstellung hervorragender Produkte und qualitativ hochwertiger Dienstleistungen verschrieben hat, um Ihren Erwartungen gerecht zu werden. Wir schätzen Ihr Feedback und ermutigen Sie, uns per E-Mail an **support@acebott.com** Ihre Kommentare oder Vorschläge zuzusenden.

Unser erfahrenes Team von Ingenieuren ist bestrebt, alle Probleme oder Fragen, die Ihnen bei der Nutzung unserer Produkte begegnen, schnell zu lösen. An Werktagen garantieren wir eine Antwort innerhalb von 24 Stunden.

.

# Inhaltsverzeichnis

**Lektion 1 Hardware-Erkenntnis und Software-Installation ..... 1**

I. Hardware-Kenntnisse ..... 2

II. Softwareinstallation ..... 3

III. Servomotor kennenlernen ..... 31

**Lektion 2 Zusammenbau des Roboterarms ..... 35**

I. Zubehörliste ..... 35

II. Bauteilliste ..... 36

III. Montageschritte ..... 37

**Lektion 3 Joystick-Steuerung für Roboterarm ..... 61**

I. Servomotorsteuerung des Roboterarms ..... 61

II. Joystick-Modul kennenlernen ..... 62

III. Grundlegende Bewegungen des joystickgesteuerten Roboterarms ..... 64

IV. Erweiterte Aufgaben ..... 67

**Lektion 4 Räumliche Koordinaten des Roboterarms ..... 68**

I. Kartesisches Koordinatensystem ..... 68

II. Gelenkkoordinatensystem ..... 69

III. Vorwärts- und inverse Kinematik ..... 70

IV. Koordinatendiagramm des Roboterarms ..... 71

V. Kalibrierungsanweisungen für den Roboterarm ..... 72

VI. Punktbewegungen in räumlichen Koordinaten ..... 73

**Lektion 5 Palettierung mit dem Roboterarm ..... 77**

I. Palettierung mit dem Roboterarm ..... 78

II. Erweiterungsaufgaben ..... 80

**Lektion 6 Teach-Funktion des Roboterarms ..... 81**

I. Teach-Programm ..... 81

II. Erweiterungsaufgaben ..... 83

**Lektion 7 Websteuerung des Roboterarms ..... 85**

I. Webbasierte Steuerungssoftware ..... 86

II. Anmeldung auf der Webseite ..... 86

III. Erweiterte Aufgaben ..... 88

**Lektion 8 APP-Steuerung des Roboterarms ..... 90**

I. App herunterladen ..... 90

II. App-Steuerung des Roboterarms ..... 92

# Lektion 1 Hardware-Erkenntnis und Software-Installation

Ein Roboterarm ist ein automatisiertes mechanisches Gerät, das in der Robotik weit verbreitet ist. Der Arm ist eine wichtige Komponente des Roboteraktuators und dient dazu, das gegriffene Werkstück an eine vorgegebene Position zu transportieren. In der Industrie können Roboterarme Menschen bei monotonen und sich wiederholenden Produktionsvorgängen oder bei Bearbeitungsvorgängen in gefährlichen und rauen Umgebungen ersetzen.

Der Roboterarm besteht aus drei grundlegenden Teilen: dem Hauptkörper, dem Antriebssystem und dem Steuerungssystem. Der Hauptkörper besteht hauptsächlich aus der Basis, dem Arm, dem Handgelenk und dem Endeffektor. Je nach unterschiedlichen Anwendungsszenarien wird der Hauptkörper des Roboterarms außerdem in vierachsige, fünfachsige, sechsachsige oder sogar mehrachsige Ausführungen unterteilt; das Antriebssystem umfasst die Antriebseinheit und den Übertragungsmechanismus, wobei der Kern aus dem Untersetzungsgetriebe und dem Servo besteht, die dazu dienen, den Roboterarm zu den entsprechenden Bewegungen anzutreiben; das Steuerungssystem sendet gemäß dem eingegebenen Programm Befehlssignale an das Antriebssystem und steuert die Bewegung des Roboterarms.

In diesem Tutorial handelt es sich bei dem Roboterarm um einen vierachsigen Roboterarm: Die erste Achse ist das Basisservo, die zweite Achse das Armservo, die dritte Achse das Ellbogenservo und die vierte Achse das Endservo.

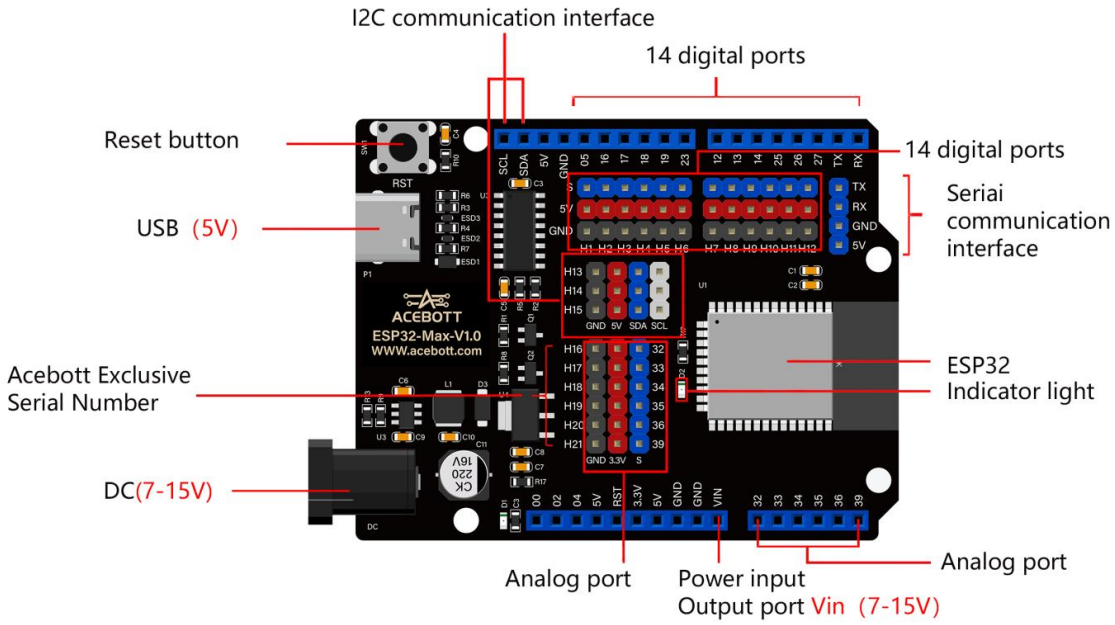
In diesem Tutorial wird als Hauptplatine des Roboterarms das gängige ESP32-Entwicklungsboard verwendet und in der Arduino IDE programmiert; zu den Steuerungsmethoden gehören: Joystick-Steuerung, Websteuerung sowie APP-Steuerung.

Wie wird der Roboterarm konkret gesteuert? Im Folgenden lernen wir dies Schritt für Schritt im Tutorial.



## I. Hardware-Kenntnisse

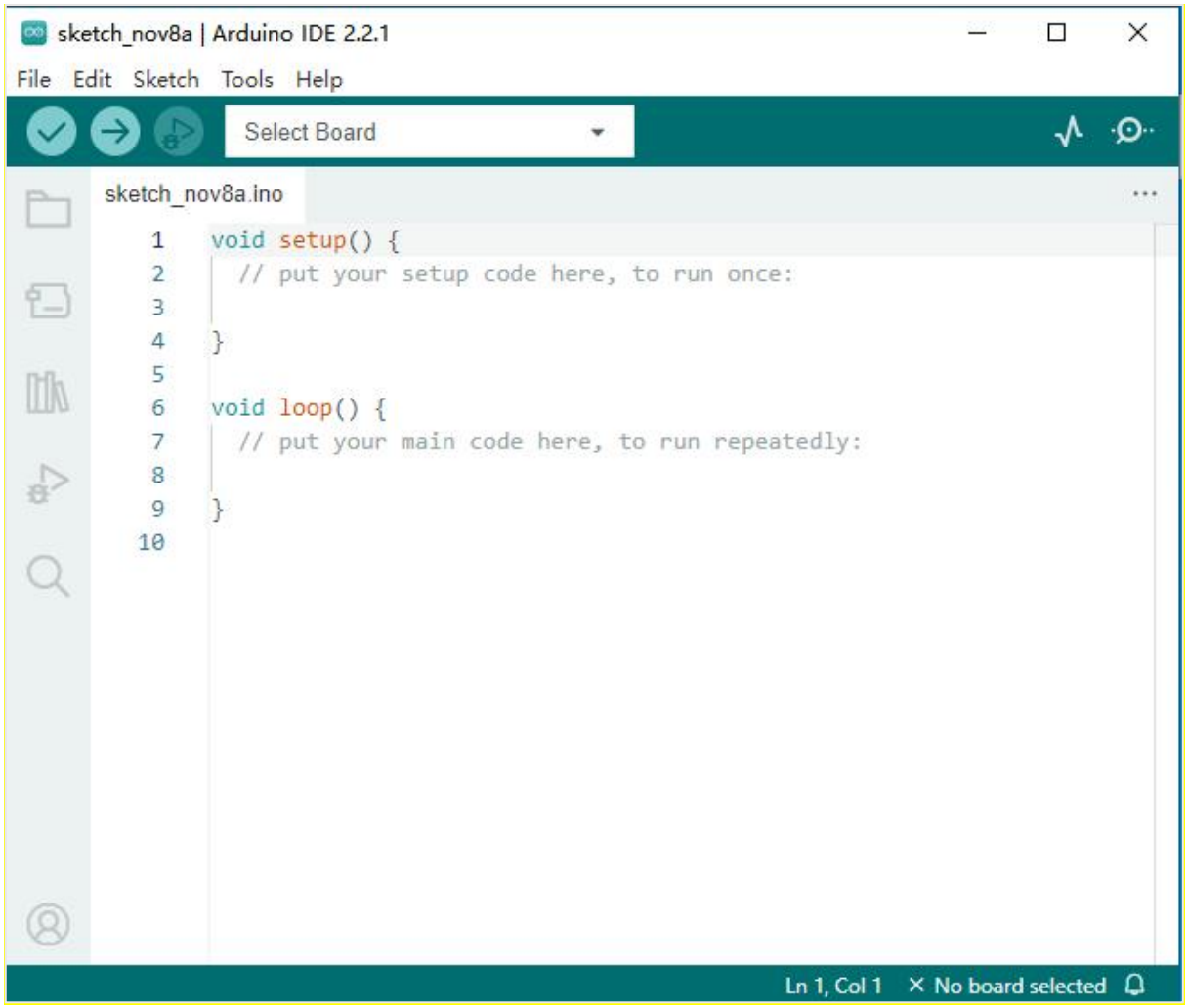
Das ESP32-Entwicklungsboard(ESP32 Max V1.0 Controller Board) ist ein stromsparender, leistungsstarker Mikrocontroller, der sich ideal für die IoT-Entwicklung eignet. Es verfügt über einen 240-MHz-Dual-Core-Prozessor, 520 KB RAM und 4 MB Flash-Speicher. Integrierte WLAN- und Bluetooth-4.2-Module ermöglichen die drahtlose Kommunikation. Es verfügt über 34 GPIO-Anschlüsse, an die verschiedene Peripheriegeräte angeschlossen und gesteuert werden können.




## II. Softwareinstallation

Im Projekt des Roboterarms verwenden wir hauptsächlich die Arduino IDE als Programmiersoftware. Sie ist eine Open-Source-Programmierumgebung und kann mit verschiedenen Entwicklungsboards kompatibel sein, wie z. B.: Arduino UNO, ESP32, ESP8266, STM32 usw.

Mit der Arduino IDE müssen Sie lediglich den Programmcode in der IDE schreiben und anschließend auf das Entwicklungsboard hochladen; das Programm teilt dem Board mit, welche Aufgaben es ausführen soll.




## 1. Installationsmethode der Arduino IDE

Öffnen Sie zunächst die offizielle Download-Seite der Arduino IDE: https://www.arduino.cc/en/software

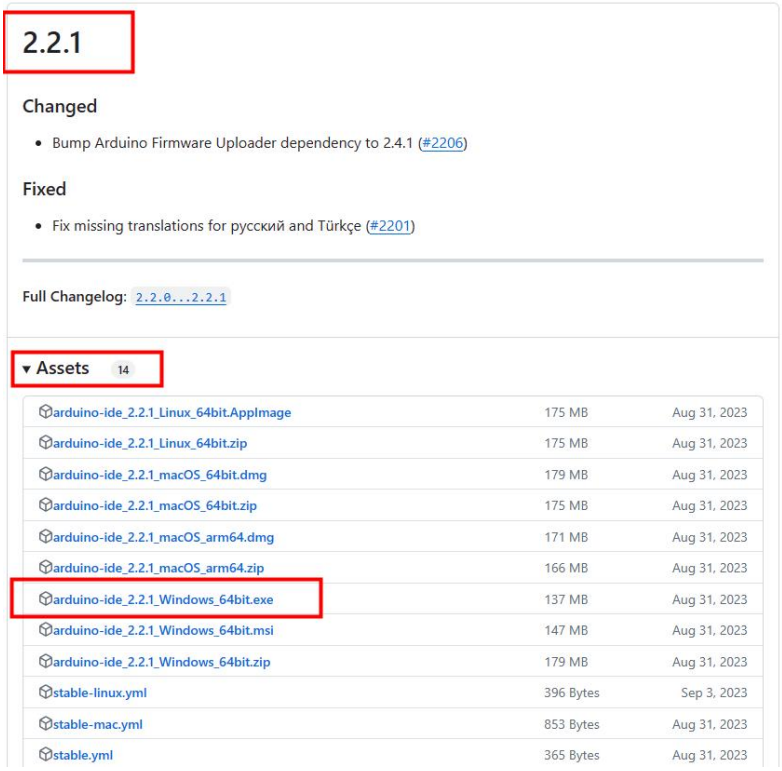
Wählen Sie je nach Betriebssystem des Computers die entsprechende Softwareversion aus.

### (1) Installationsmethode unter dem Windows-System

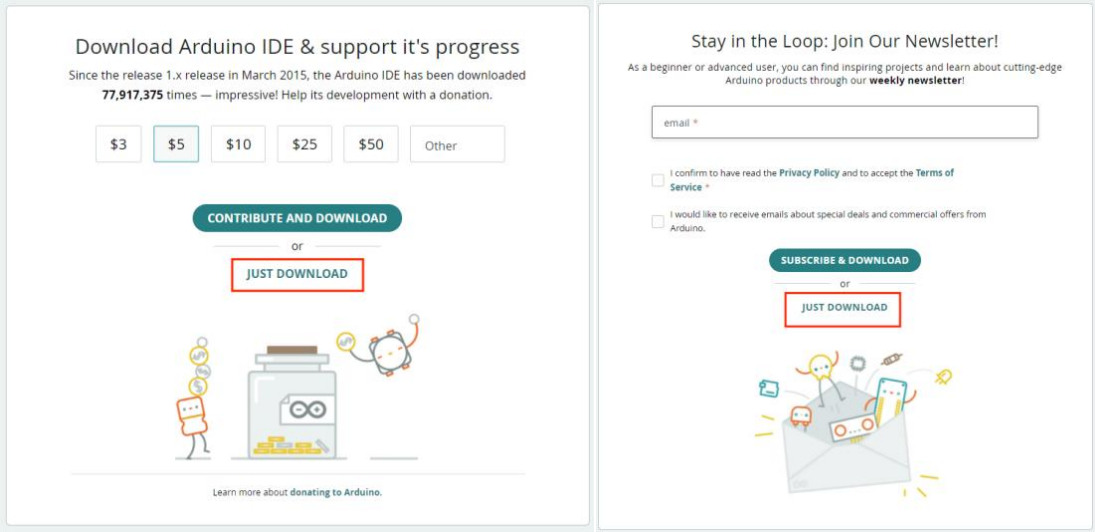
①Klicken Sie mit der Maus auf die in der Abbildung gezeigte Position. (Hinweis: Da in neueren Versionen möglicherweise Kompatibilitätsprobleme auftreten, wird empfohlen, die Arduino IDE Version 2.2.1 zu installieren.)



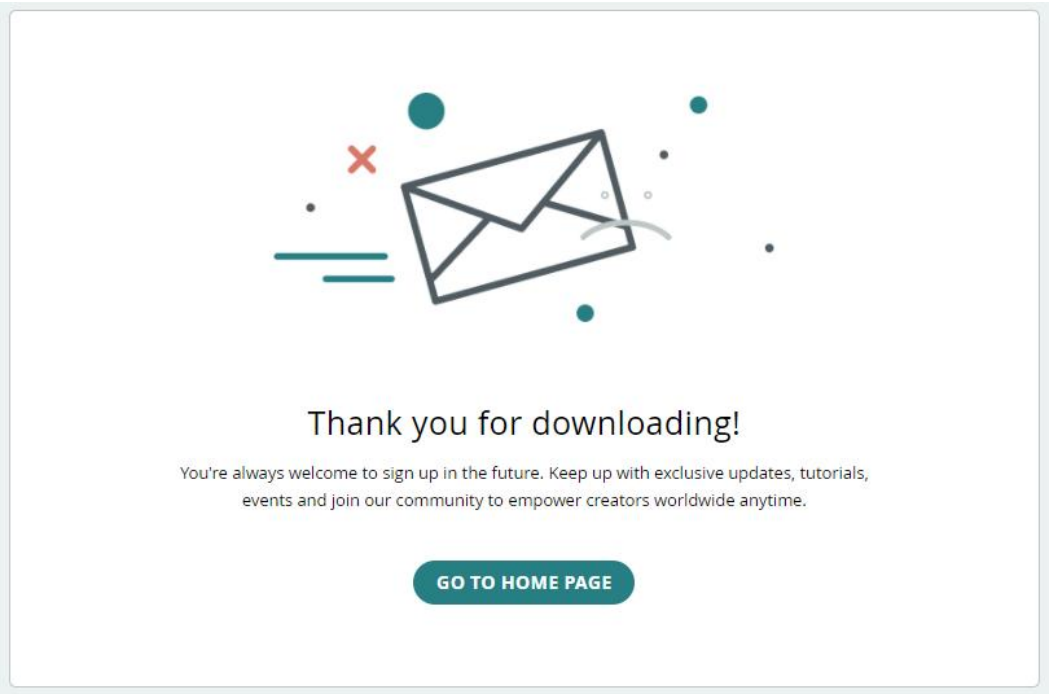

Link zur Version 2.2.1: https://github.com/arduino/arduino-ide/releases




② Wählen Sie „JUST DOWNLOAD“.



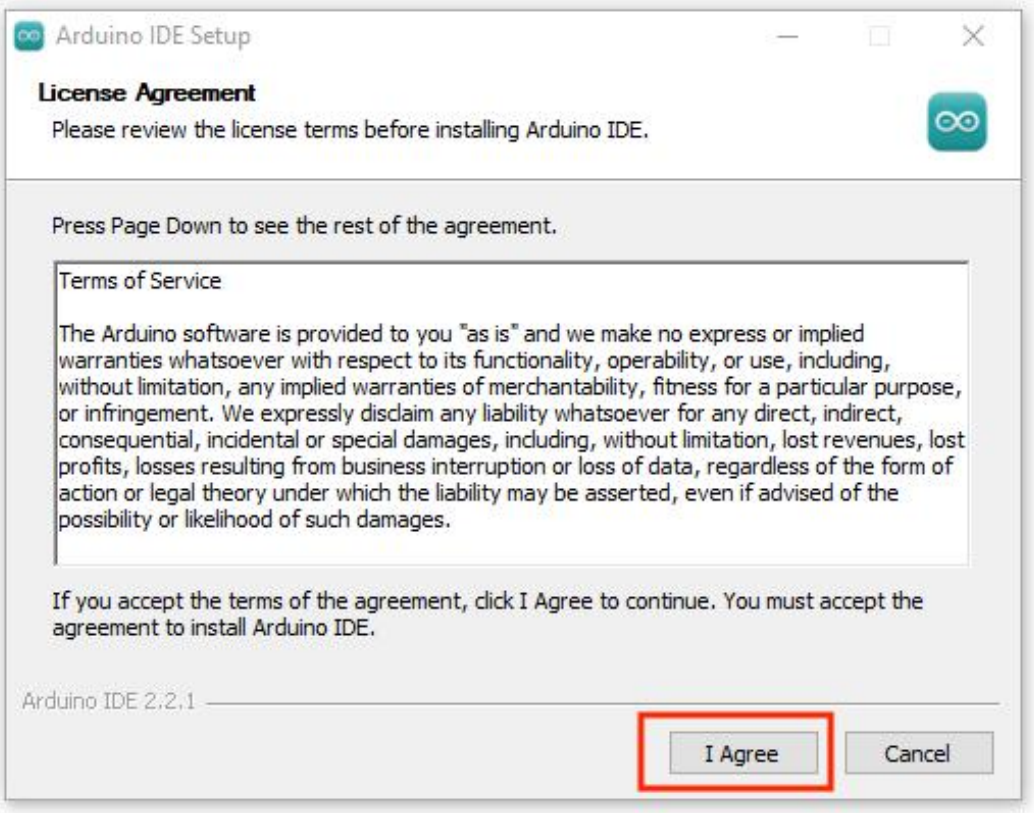

③ Wenn die folgende Oberfläche erscheint, wird angezeigt, dass die Arduino IDE heruntergeladen wird.



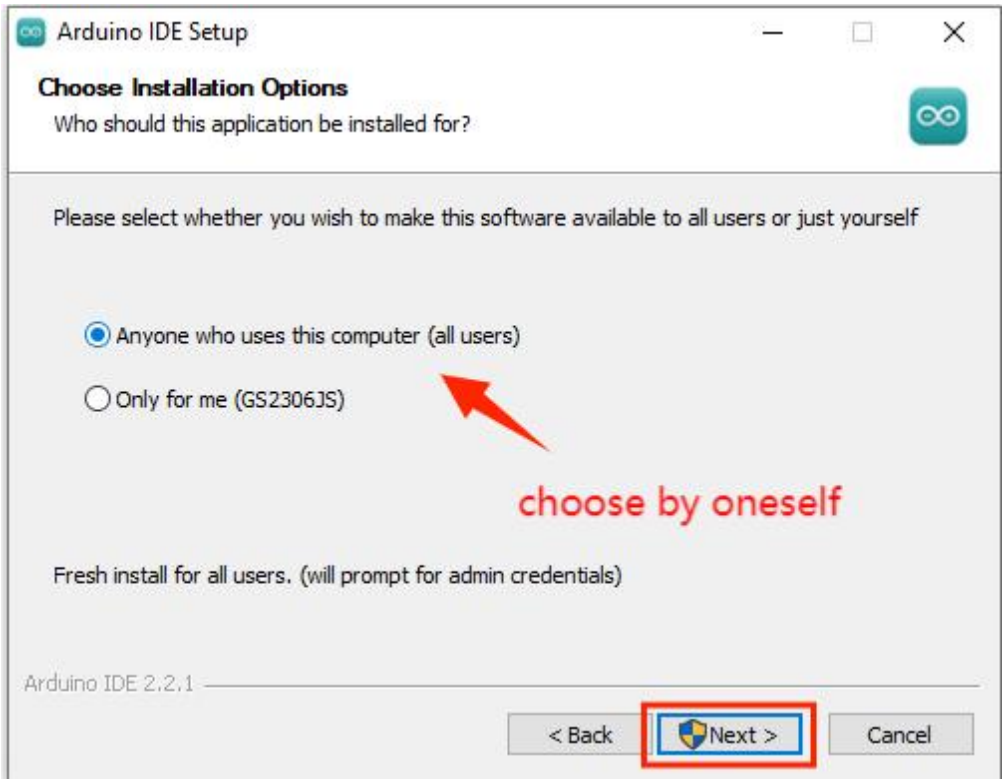

④ Nach Abschluss des Downloads erscheint die im Bild gezeigte Datei. Klicken Sie darauf, um die Software zu installieren.



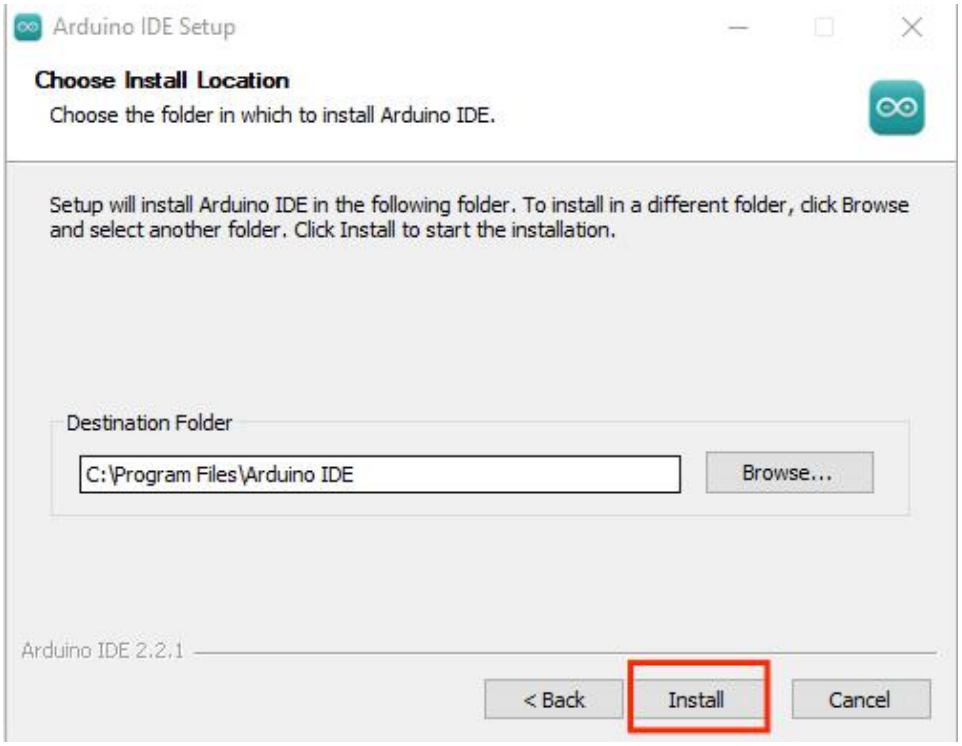

⑤ Nach der Installation erscheint die folgende Oberfläche. Wählen Sie „I Agree“.



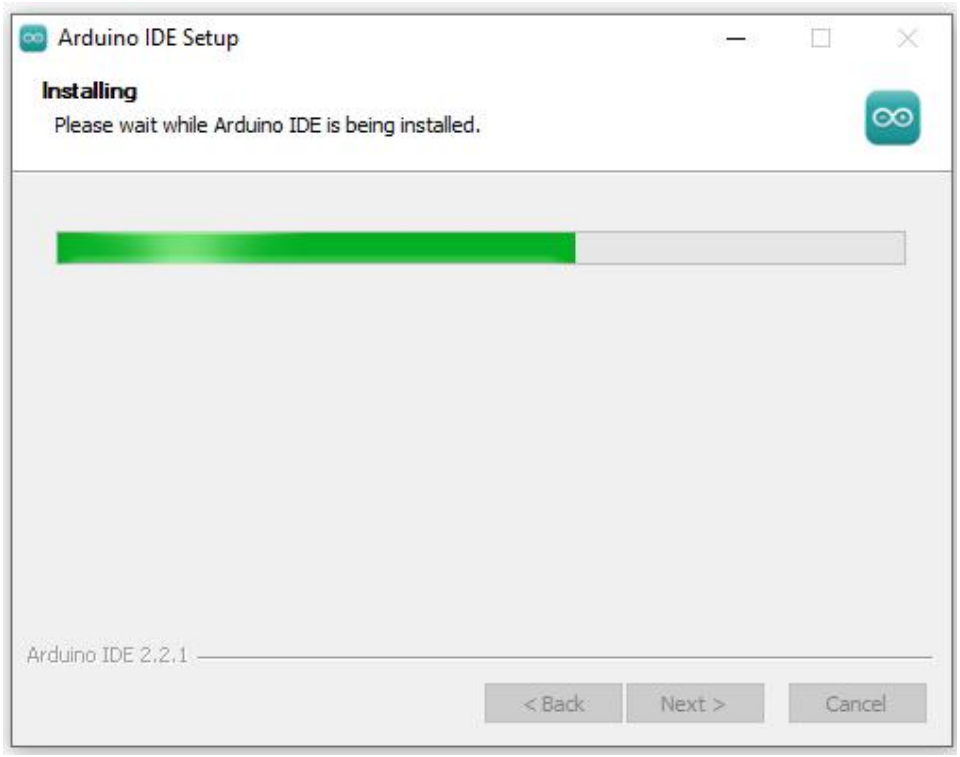

⑥ Nach Auswahl von „I Agree“ erscheint die folgende Oberfläche. Wählen Sie „Next“.



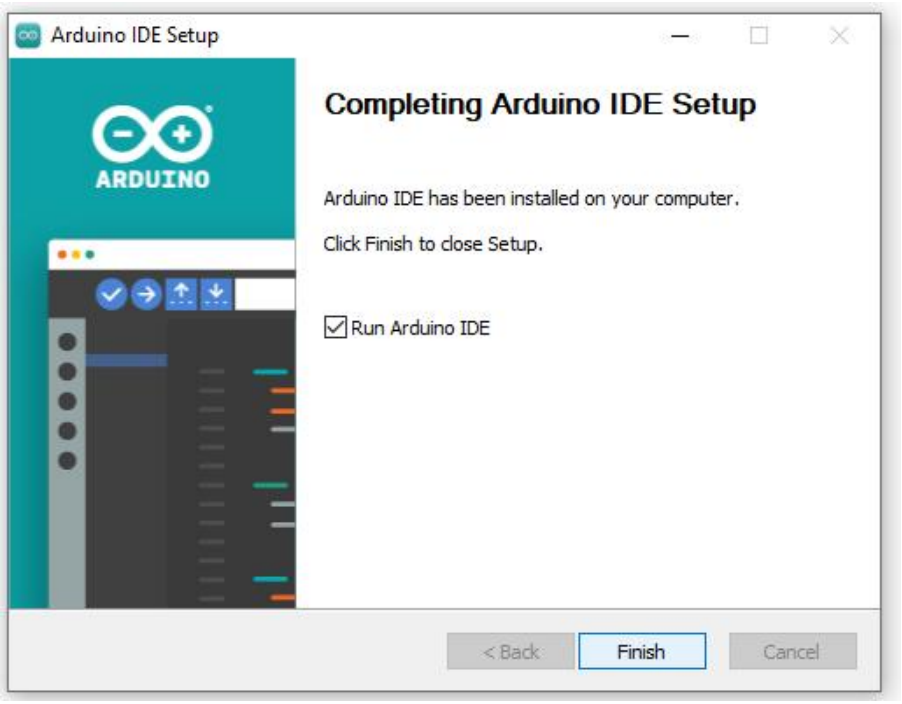
⑦ Nach Auswahl von „Next“ erscheint die folgende Oberfläche. Wählen Sie „Install“.




⑧ Die Arduino IDE wird installiert.



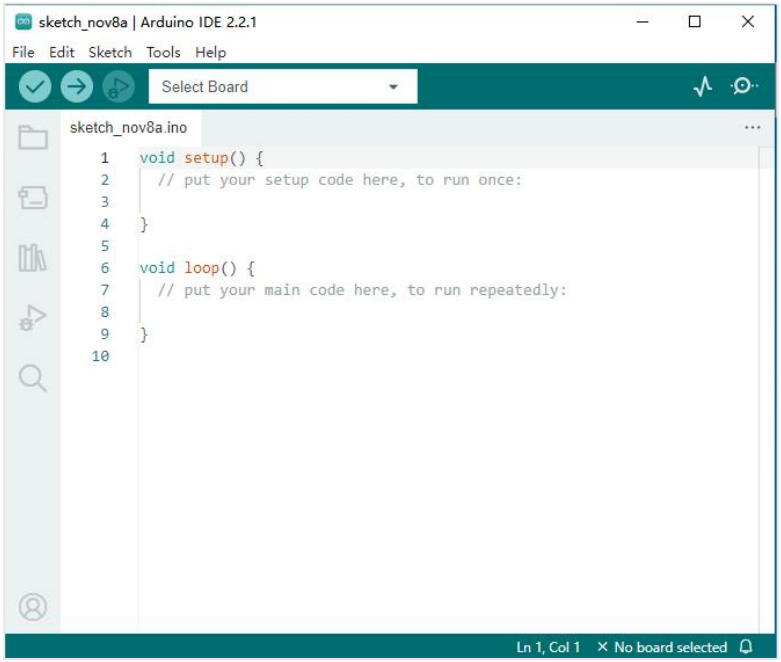

⑨ Die Installation ist abgeschlossen. Klicken Sie auf „Finish“.




⑩ Nach der Installation erscheint ein Verknüpfungssymbol der Arduino IDE auf dem Desktop.




⑪ Nach dem Öffnen erscheint die folgende Benutzeroberfläche der Software.

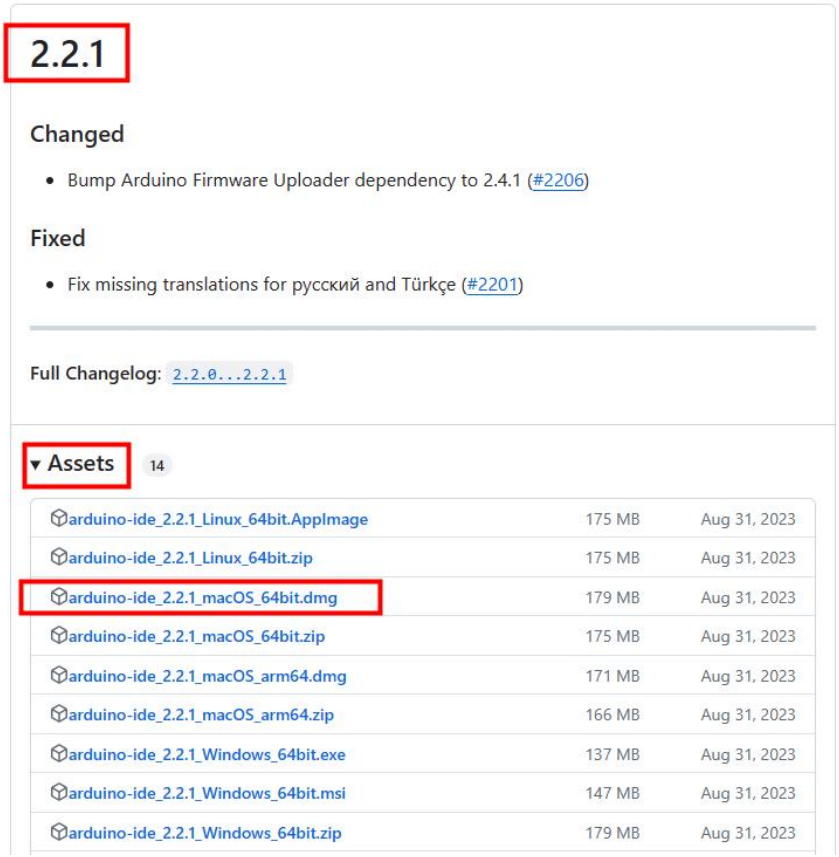


### (2) Installation unter macOS

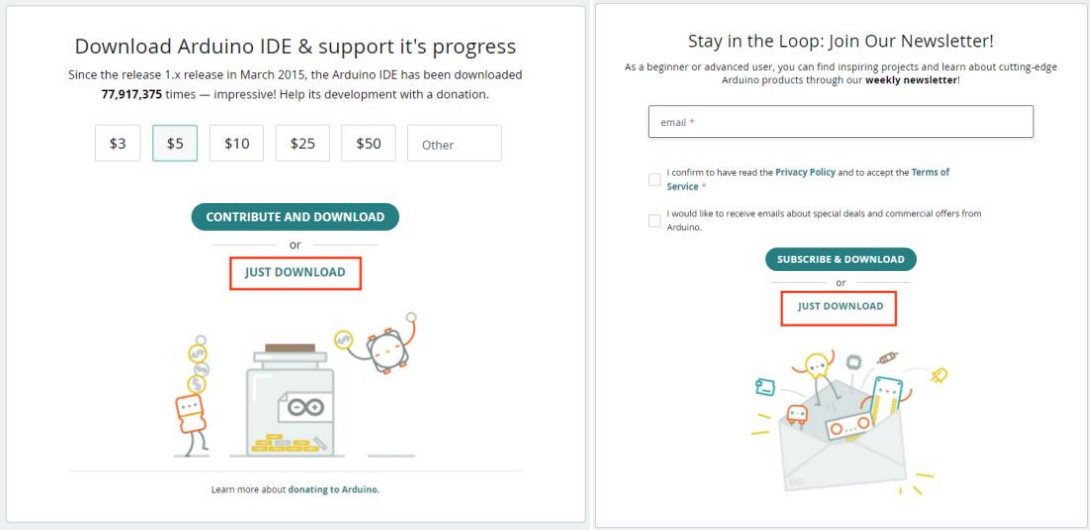
①Klicken Sie mit der Maus auf die in der Abbildung gezeigte Position. (Hinweis: Da in neueren Versionen möglicherweise Kompatibilitätsprobleme auftreten, wird empfohlen, die Arduino IDE Version 2.2.1 zu installieren.)



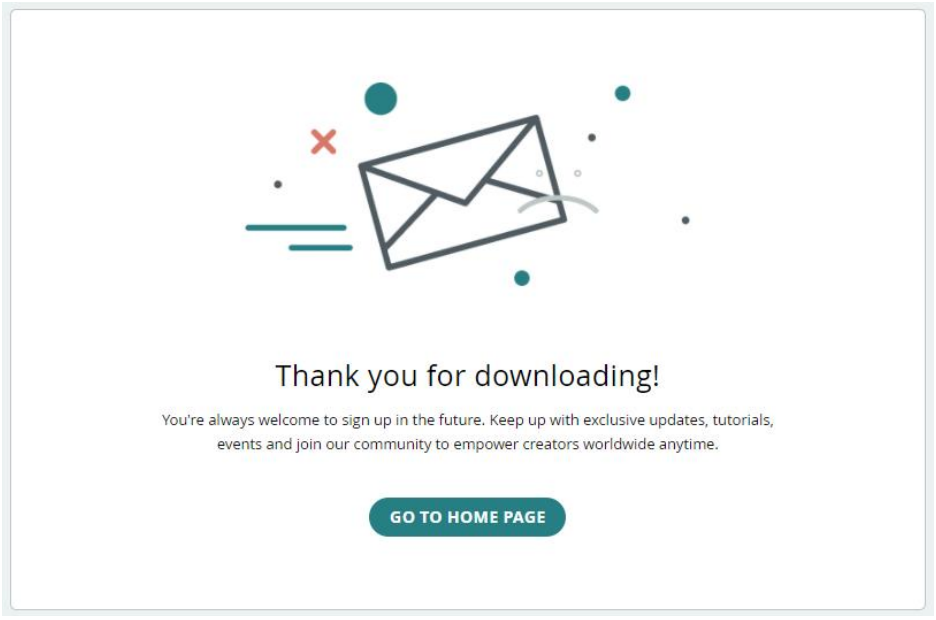

Link zur Version 2.2.1: https://github.com/arduino/arduino-ide/releases



② Wählen Sie „JUST DOWNLOAD“.

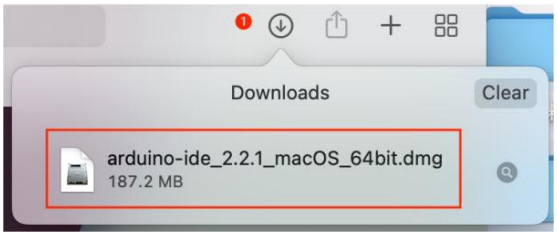



③ Wenn die folgende Oberfläche erscheint, wird angezeigt, dass die Arduino IDE heruntergeladen wird.

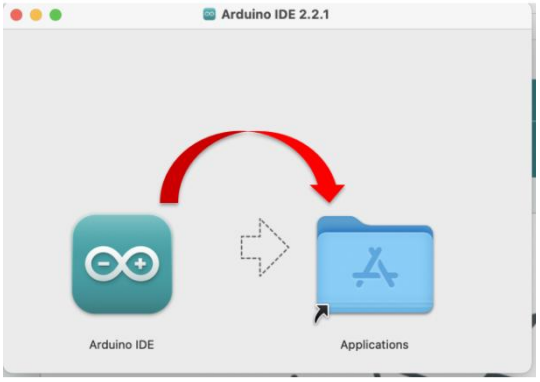



④ Nach Abschluss des Downloads klicken Sie auf das Download-Symbol im Browser und suchen das Installationspaket der Arduino IDE.





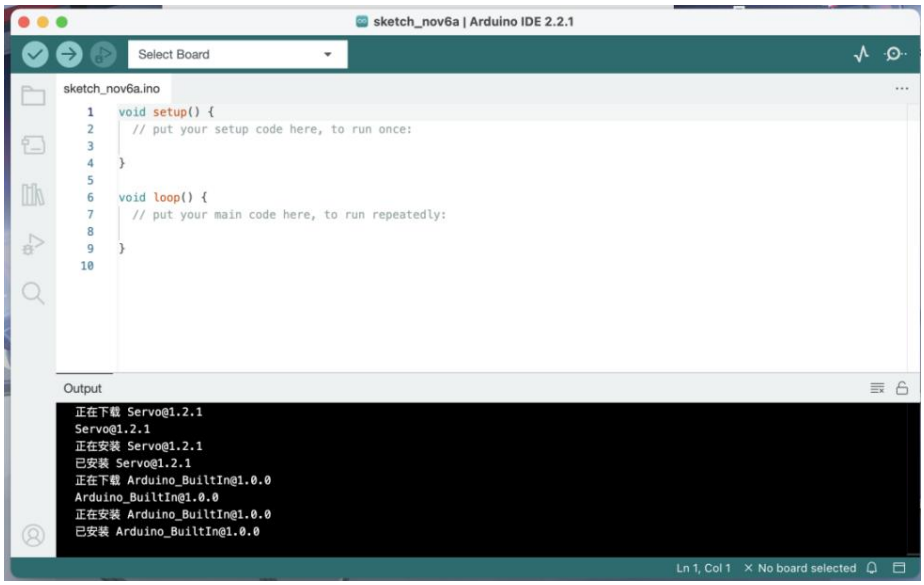

⑤ Klicken Sie auf das Installationspaket. Es erscheint die Installationsoberfläche. Wählen Sie einfach das Symbol der Arduino IDE aus und ziehen Sie es in den Ordner „Applications“, um das Programm zu installieren.



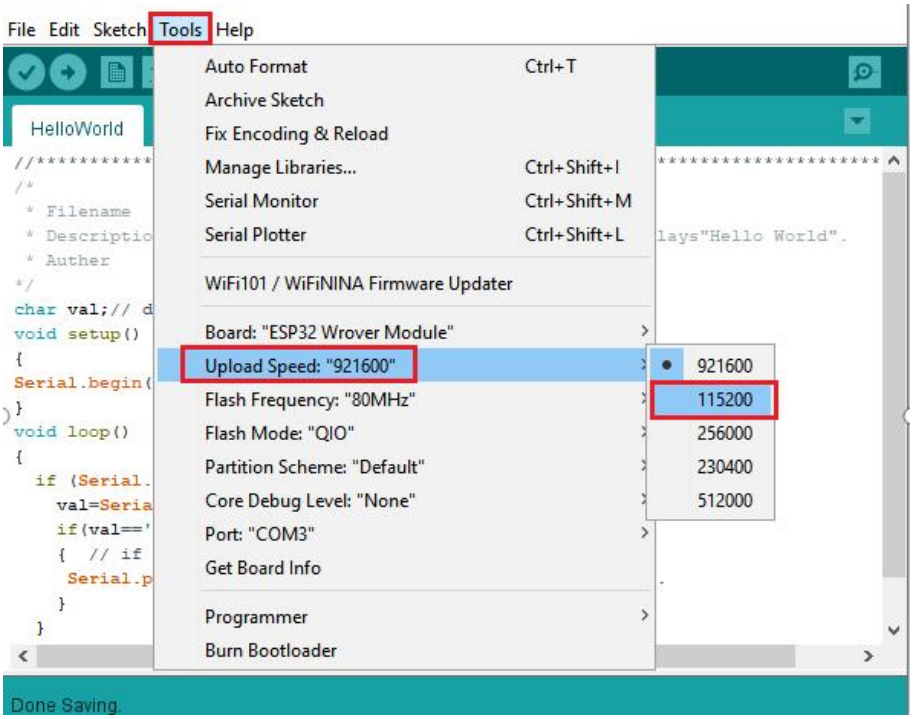

⑥ Im Arbeitsbereich finden Sie die Arduino IDE und öffnen sie.




⑦ Nach dem Öffnen sehen Sie die folgende Softwareoberfläche.



⑧ Für Mac-OS-Benutzer stellen Sie bitte vor dem Klicken auf die Upload-Schaltfläche die Baudrate auf 115200 ein.




**2. Seriellen Porttreiber Installieren (überspringen, Falls Bereits Installiert)**

Die serielle Schnittstelle ist eine Art von Computerkommunikationsschnittstelle und wird üblicherweise verwendet, um Daten zwischen dem Computer und anderen Geräten (wie Modems, Sensoren, Druckern, Mikrocontrollern usw.) zu übertragen. Die USB-serielle Schnittstelle ist die von uns am häufigsten verwendete Kommunikationsschnittstelle.

In der Regel muss nach der ersten Installation der Arduino IDE ein Treiber für die serielle Schnittstelle installiert werden, damit die erstellten Programme auf das Mainboard übertragen werden können.

Der USB-zu-Seriell-Chip der ESP32-Steuerplatine ist CH340, daher muss auch der Treiber für diesen Chip installiert werden. Nachdem die Steuerplatine über ein USB-Kabel mit dem Computer verbunden wurde,

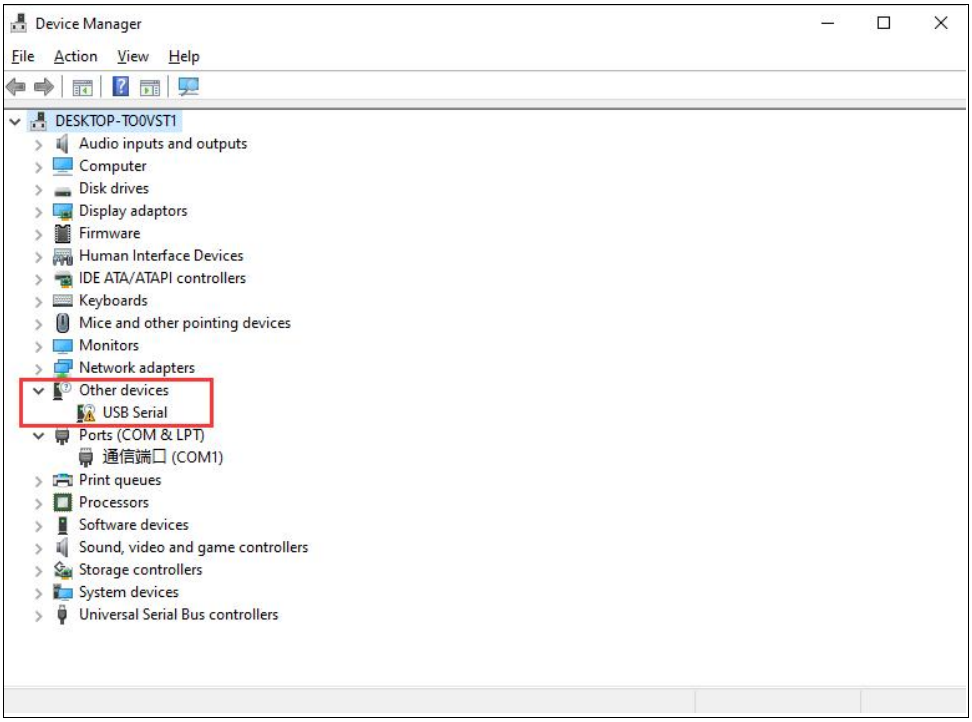
wird der Treiber normalerweise unter MacOS und Windows automatisch installiert. Falls die automatische Installation fehlschlägt, muss der Treiber manuell installiert werden.

Die Treiber können im Internet heruntergeladen oder im von uns bereitgestellten Ressourcenpaket im Ordner „CH340 Driver“ gefunden werden. Wählen Sie entsprechend dem Betriebssystem Ihres Computers das passende Installationspaket aus und folgen Sie den untenstehenden Schritten zur Installation.

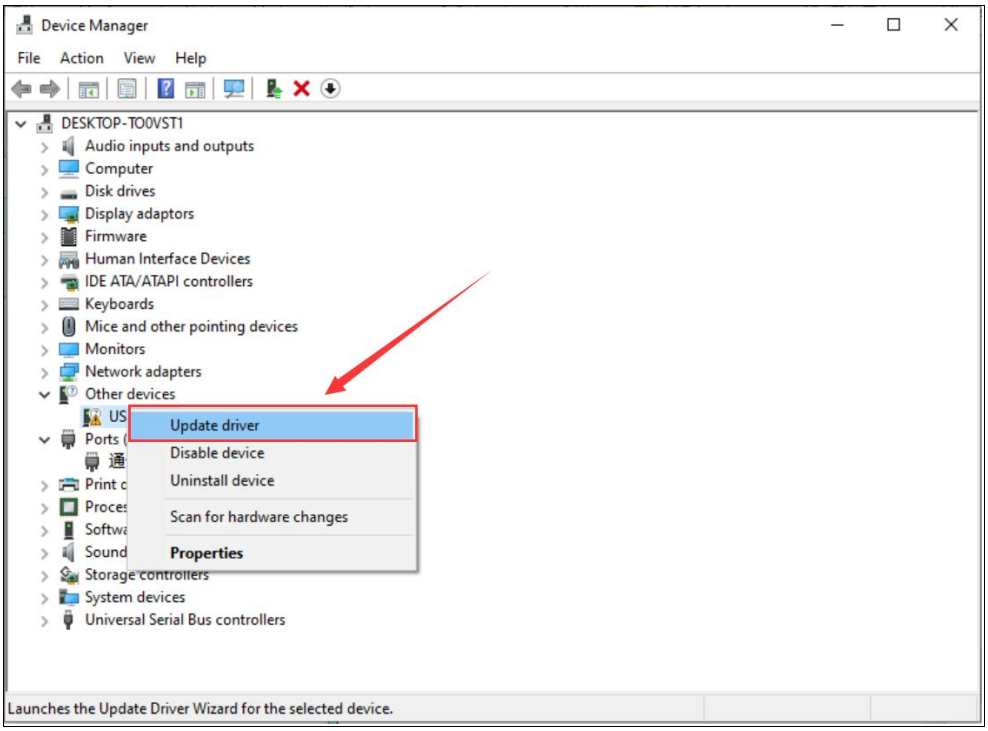
**(1) Installationsmethode für Treiber unter Windows**

① Stecken Sie ein Ende des USB-Kabels in das ESP32-Entwicklungsboard und das andere Ende in den USB-Anschluss des Computers.

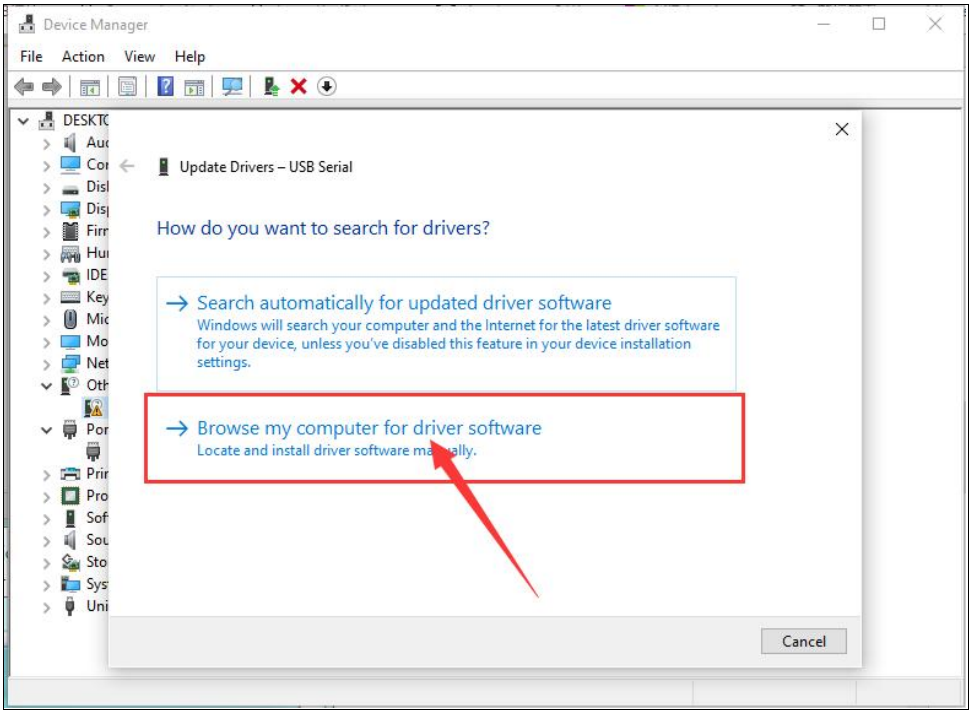
② Wenn Sie das ESP32-Entwicklungsboard zum ersten Mal mit dem Computer verbinden, klicken Sie mit der rechten Maustaste auf „My Computer“ → „Properties“ → „Device Manager“. Unter „Other devices“ sehen Sie dann „USB Serial“ oder ein unbekanntes Gerät.



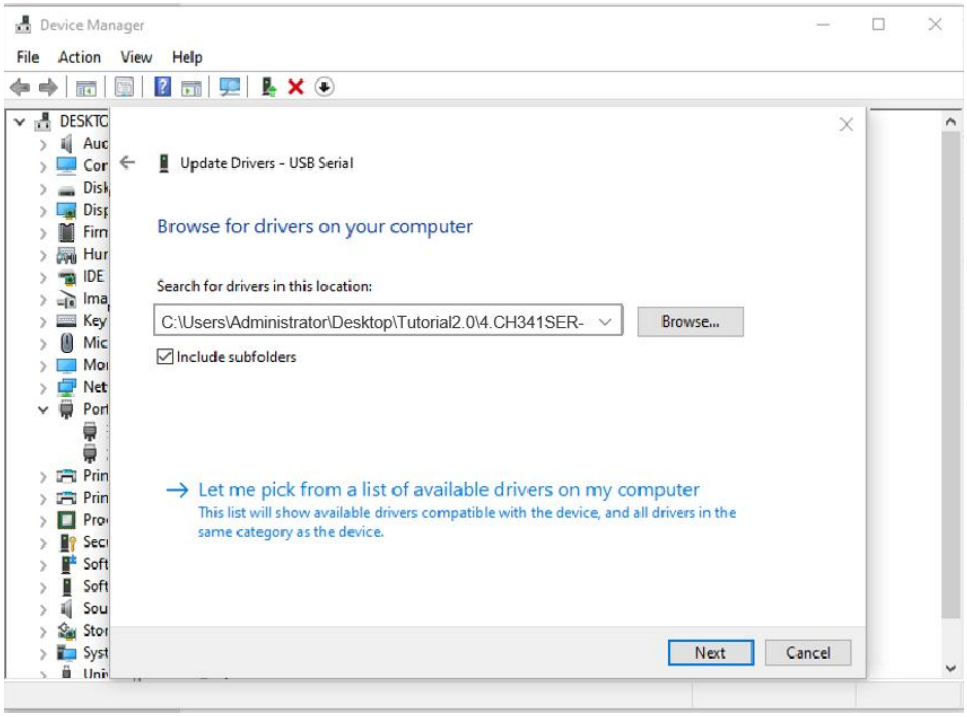
③ Klicken Sie zunächst mit der rechten Maustaste auf das Gerät und wählen Sie dann im oberen Menü „Update driver“, wie in der Abbildung gezeigt.



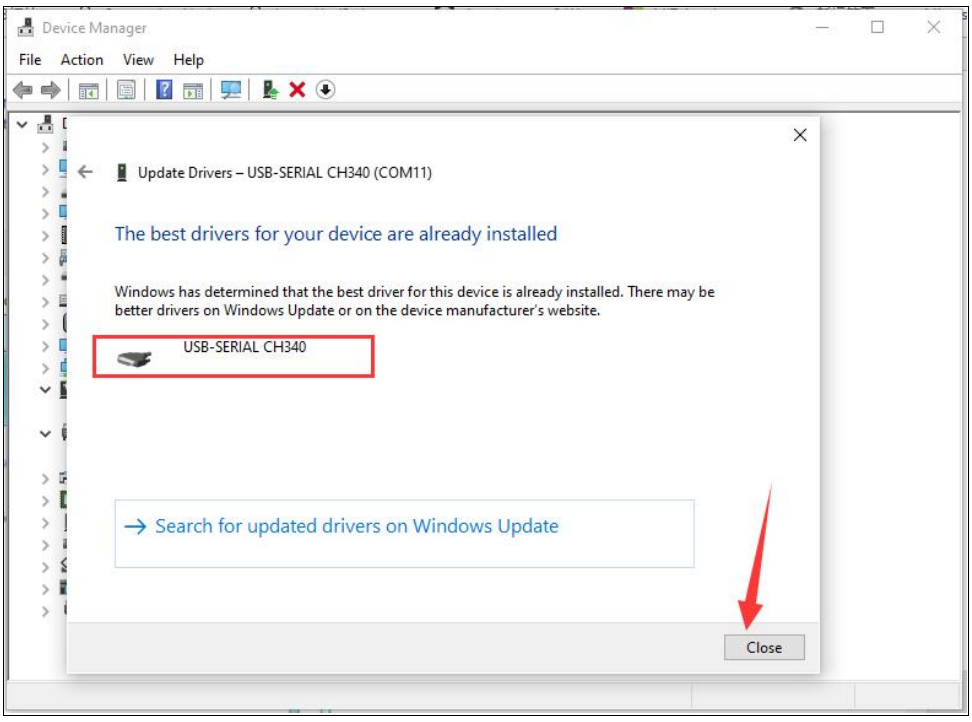

④ Es erscheint die Auswahl „Search automatically for updated driver software“ oder „Browse my computer for driver software“. Wählen Sie „Browse my computer for driver software“.



⑤ Fügen Sie anschließend den Pfad zur Treiberdatei hinzu.

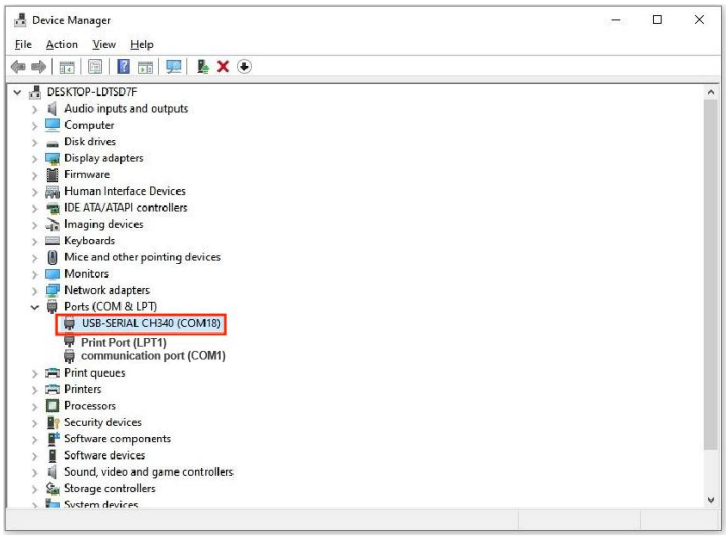



⑥ Nach Abschluss der Installation erhalten Sie eine Bestätigung. Klicken Sie auf „Close“.



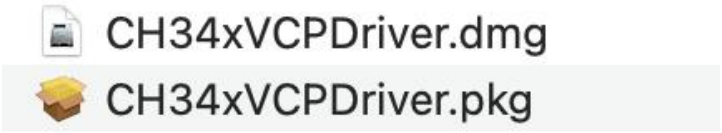

⑦ Überprüfen Sie, ob die Installation erfolgreich war: Stecken Sie ein Ende des USB-Kabels in die ESP32-Steuerplatine und das andere Ende in den USB-Anschluss des Computers. Klicken Sie mit der rechten

Maustaste auf „My Computer“ → „Properties“ → „Device Manager“. Nach dem Anschließen der Steuerplatine zeigt die folgende Abbildung eine erfolgreiche Installation an.

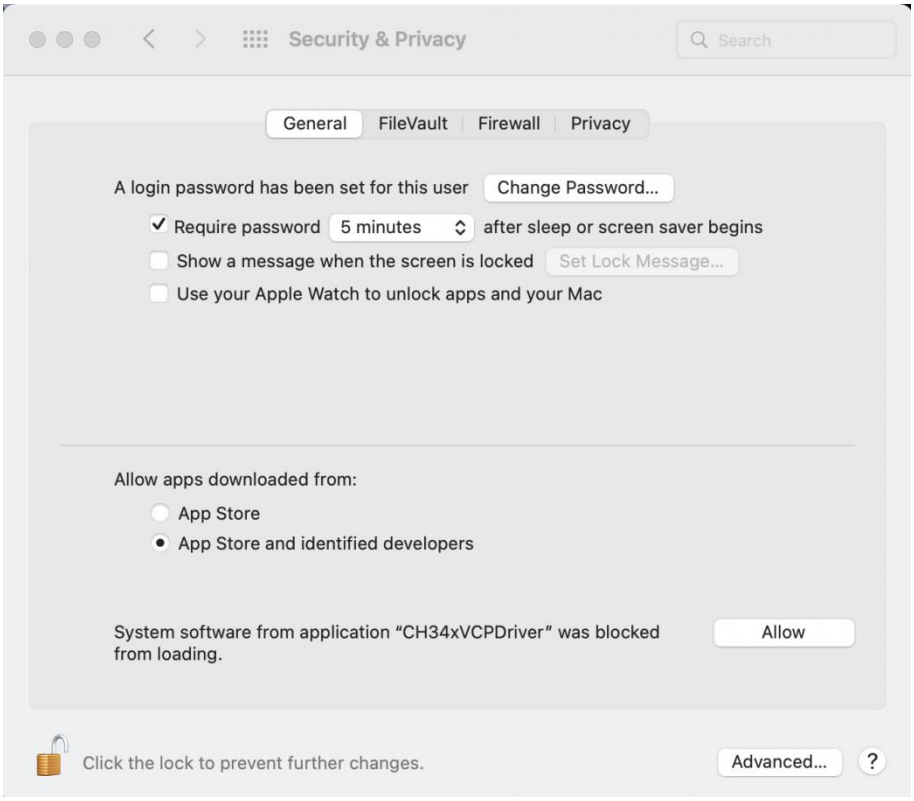



### (2) Installationsmethode unter Mac

Öffnen Sie den Ordner „CH340 Driver File-MAC“. Dort finden Sie die von uns bereitgestellten Treiberinstallationsdateien. In der Regel wird standardmäßig der Treiber im pkg-Format installiert. Falls OS X 11.0 oder höher Rosetta nicht unterstützt, installieren Sie bitte den Treiber im dmg-Format.

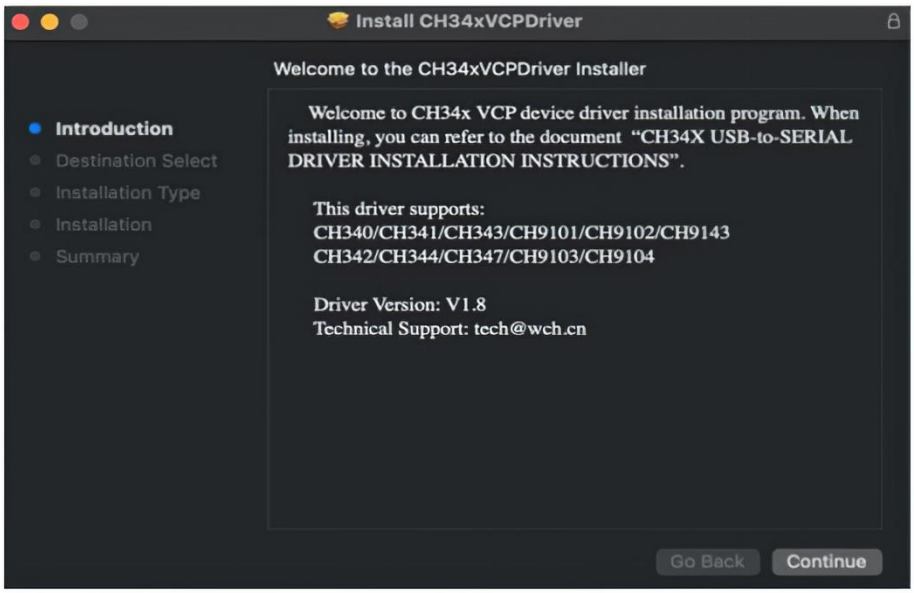


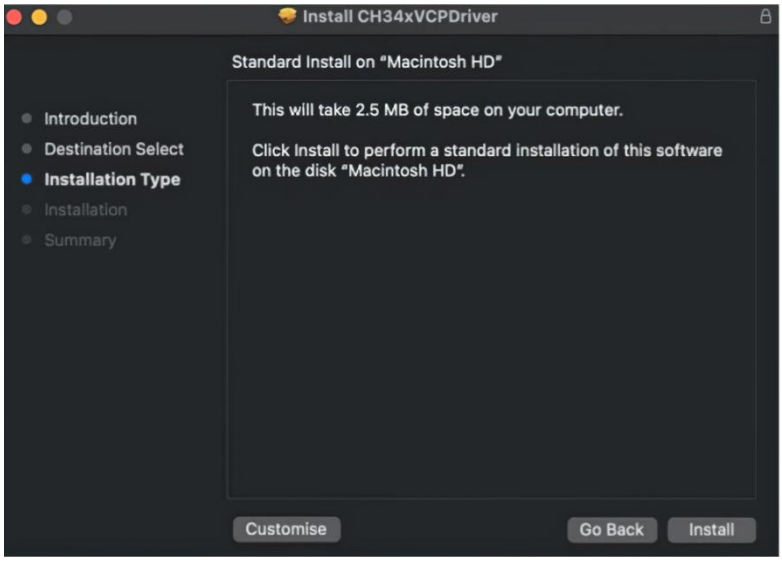

Vor der Installation gehen Sie bitte zu „System Preferences“ → „Security & Privacy“ → „General“ und wählen unter „Allow apps download from: “ die Option „App Store and identified developers“, damit der Treiber ordnungsgemäß ausgeführt werden kann.



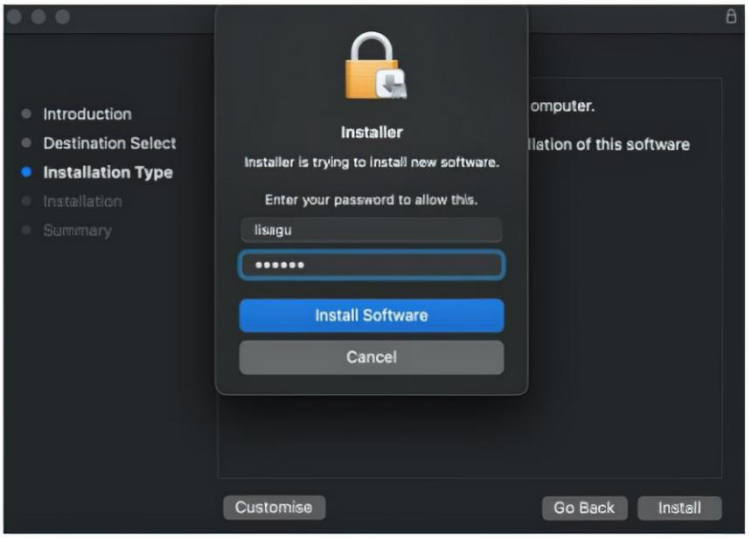

## A.Installation des pkg-Treibers

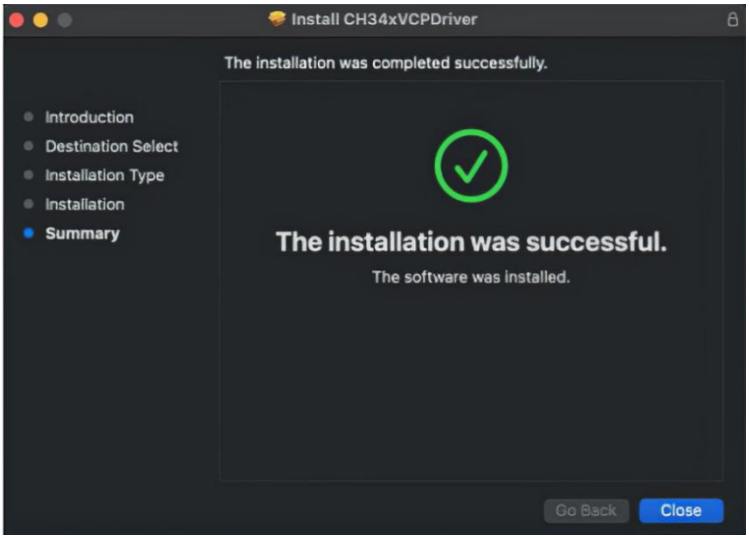
① Klicken Sie auf die Treiberdatei → „Continue“ → „Install“.



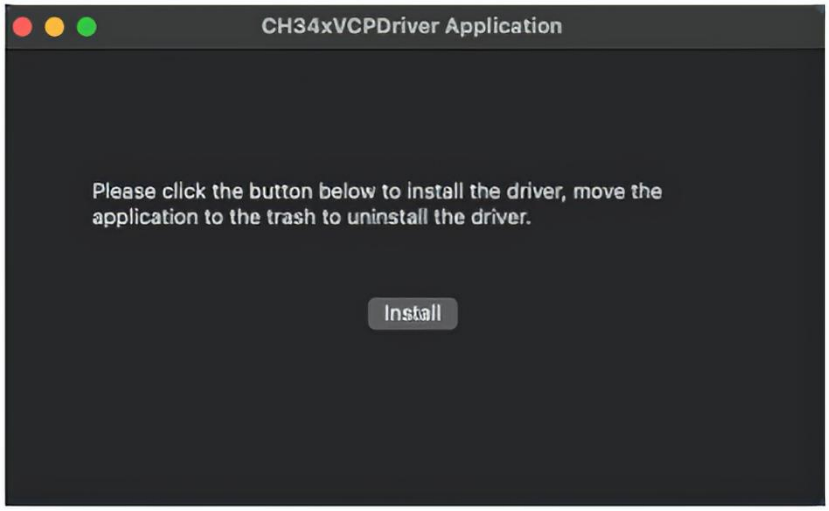



② Installation erfolgreich.

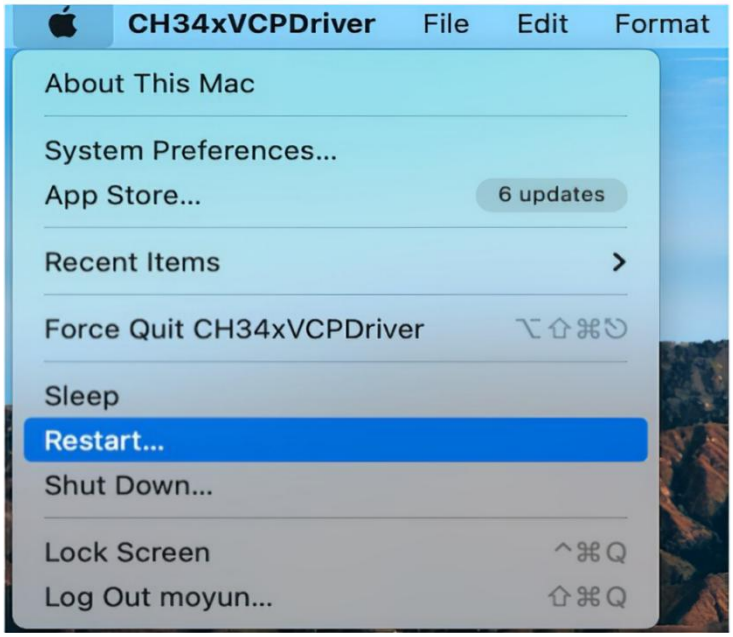


③ Unter OS X 11.0 und höher: Öffnen Sie „LaunchPad“ → „CH34xVCPDriver“ → „Install“.




④ Unter OS X 10.9 bis OS X 10.15 klicken Sie auf „Restart“, um den Computer neu zu starten, und führen Sie nach dem Neustart die folgenden Schritte aus.



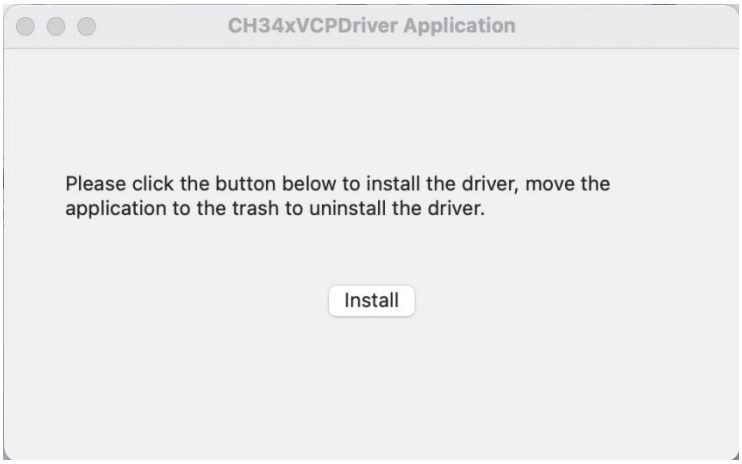

**B.Installation des dmg-Treibers (Hinweis: Wenn Schritt A erfolgreich war, überspringen Sie diesen Schritt.)**

① Installieren Sie den dmg-Treiber, indem Sie die dmg-Datei öffnen und „CH34xVCPDriver“ in den Programme-Ordner ziehen.

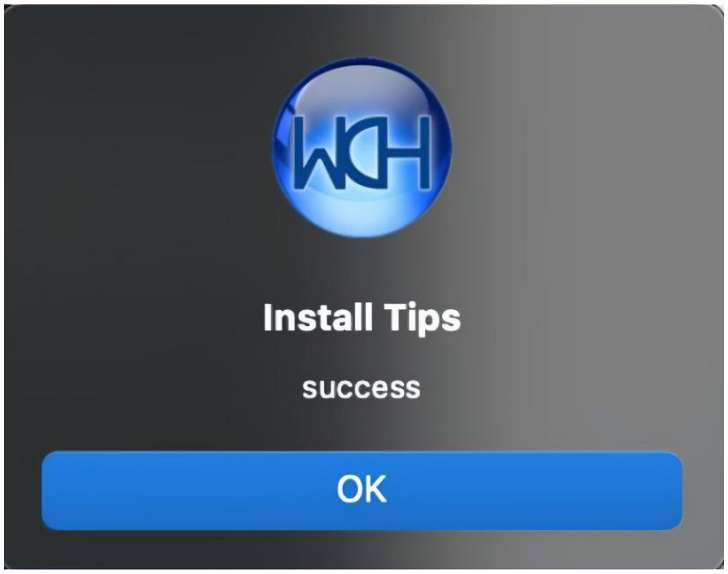



② Öffnen Sie anschließend „LaunchPad“ → „CH34xVCPDriver“ → „Install“.



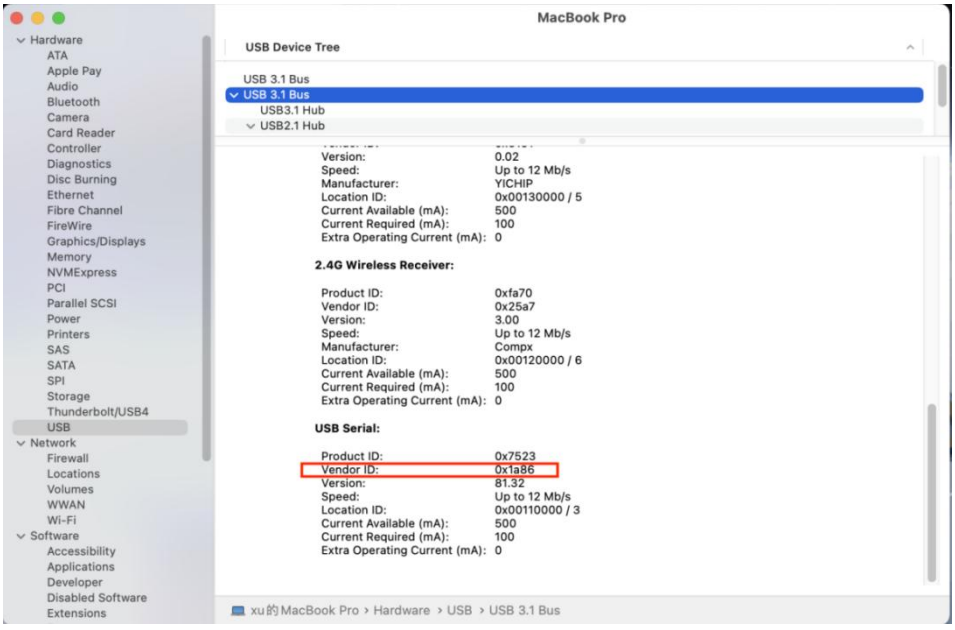




③ Installation erfolgreich.




## C.Überprüfung, ob der CH340-Treiber installiert ist

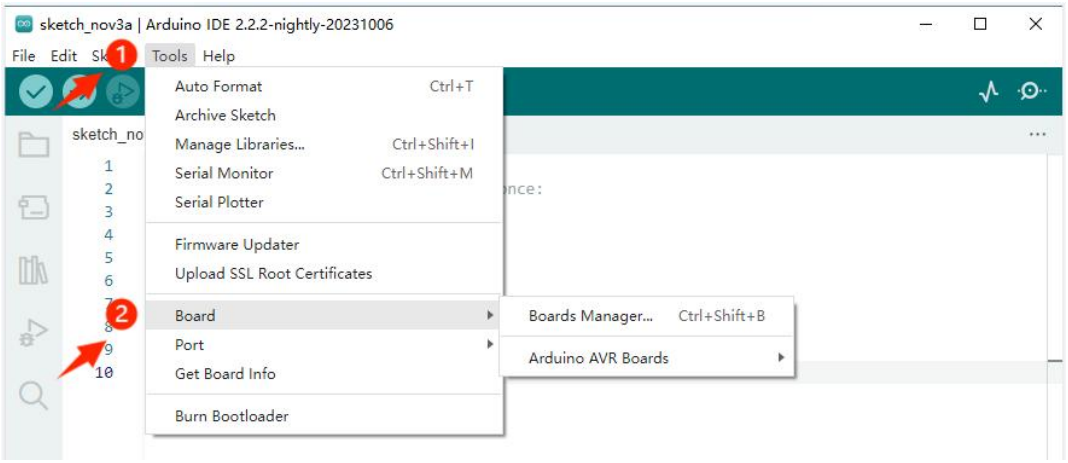
Schließen Sie die Steuerplatine an den USB-Port an und öffnen Sie „System Report“ → „Hardware“ → „USB“. Rechts werden die USB-Geräte angezeigt. Wenn das USB-Gerät ordnungsgemäß funktioniert, finden Sie ein Gerät mit der „Vendor ID“ [0x1a86].



### 3.Installation der ESP32-Bibliothek

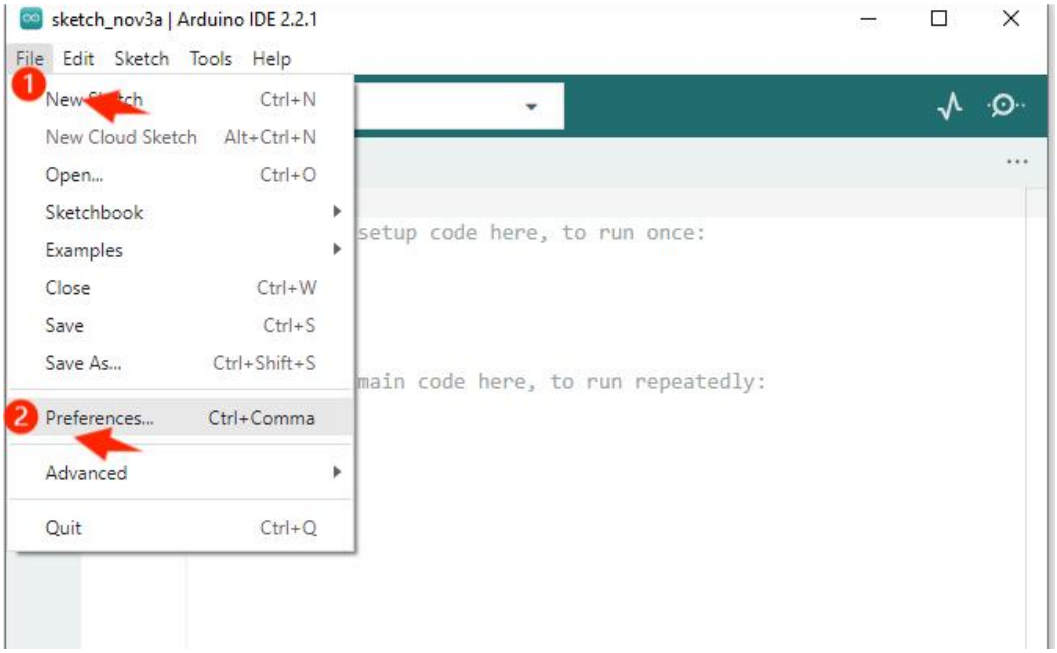
Da das von uns verwendete Entwicklungsboard ESP32 ist, müssen Sie zusätzlich die ESP32-Bibliothek hinzufügen, damit Sie ESP32 in der Arduino IDE entwickeln können.

Wenn Sie die Arduino IDE öffnen und Tools > Board auswählen, werden Sie feststellen, dass in der Arduino IDE nur „Arduino AVR Boards“ vorhanden sind und kein ESP32.

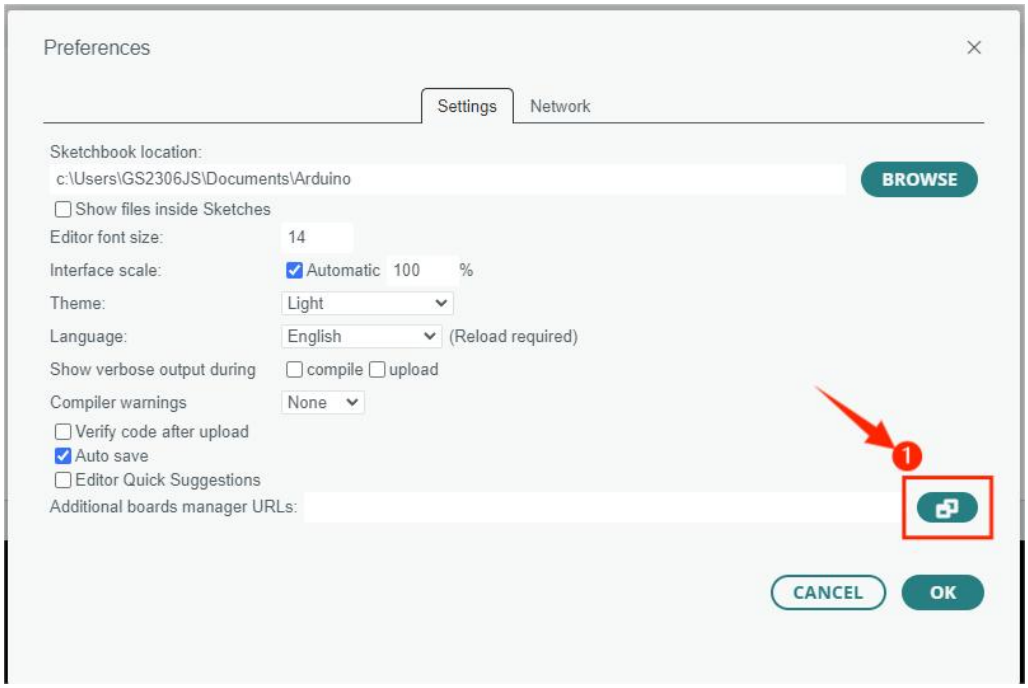



Führen Sie die folgenden Schritte aus, um die ESP32-Bibliothek zu installieren:

① Öffnen Sie „File“ > „Preferences“.

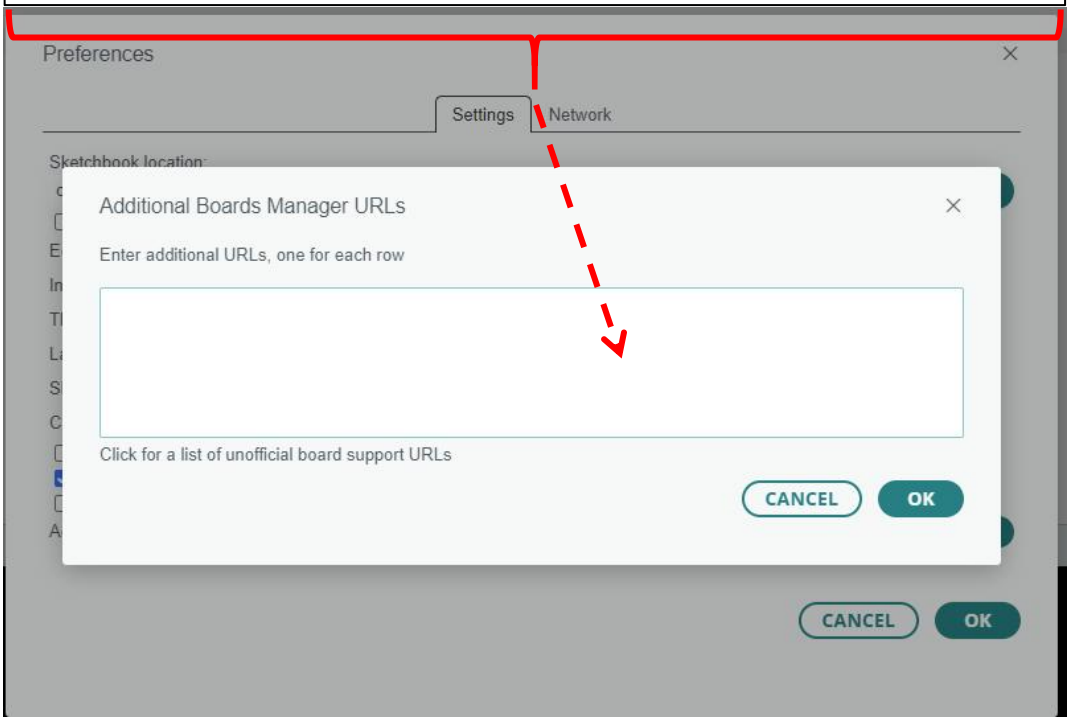



② Fügen Sie die URL für den Boardverwalter hinzu.

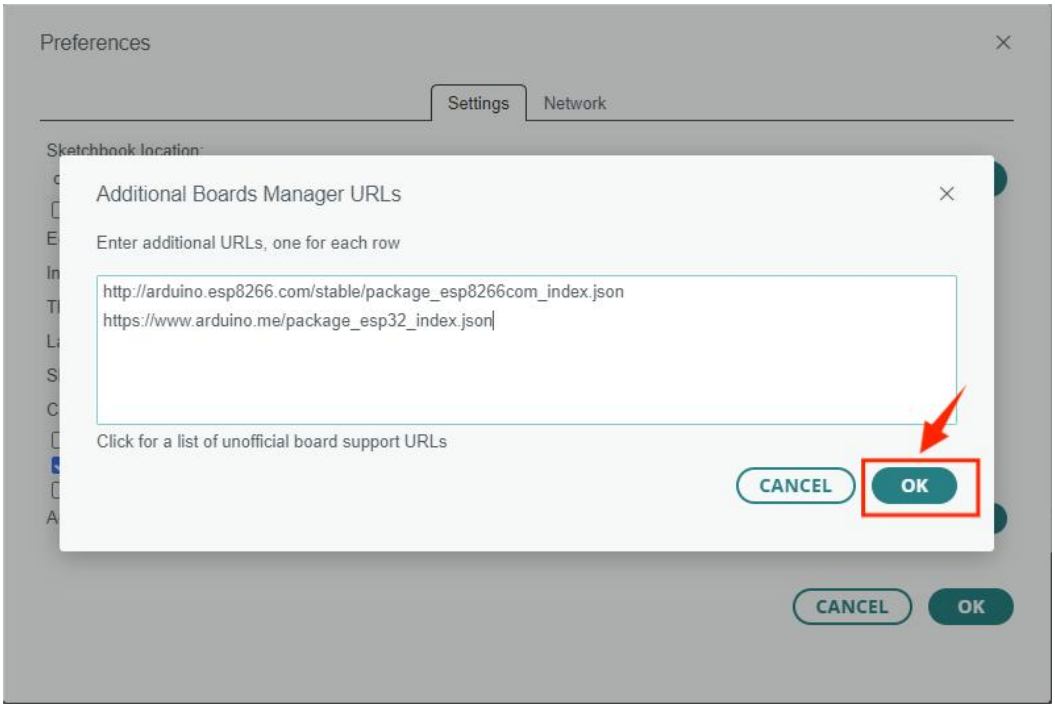


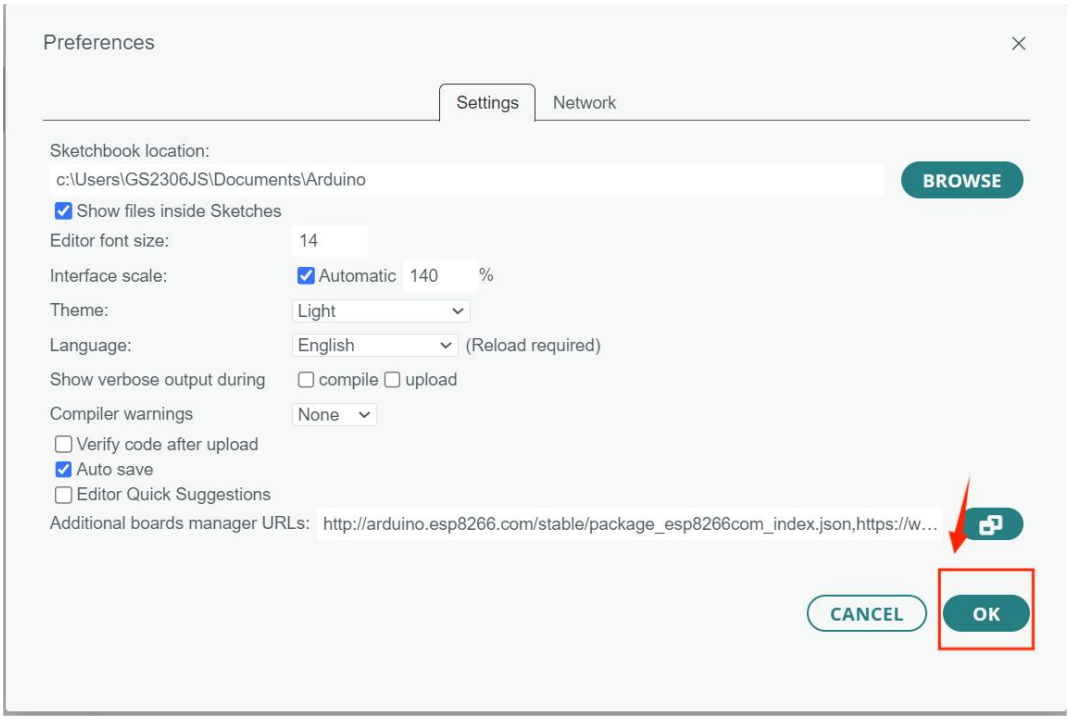

③ Kopieren Sie die URL aus dem untenstehenden Textfeld und fügen Sie sie in „Additional Boards Manager URLs“ ein.

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
https://www.arduino.me/package_esp32_index.json
```

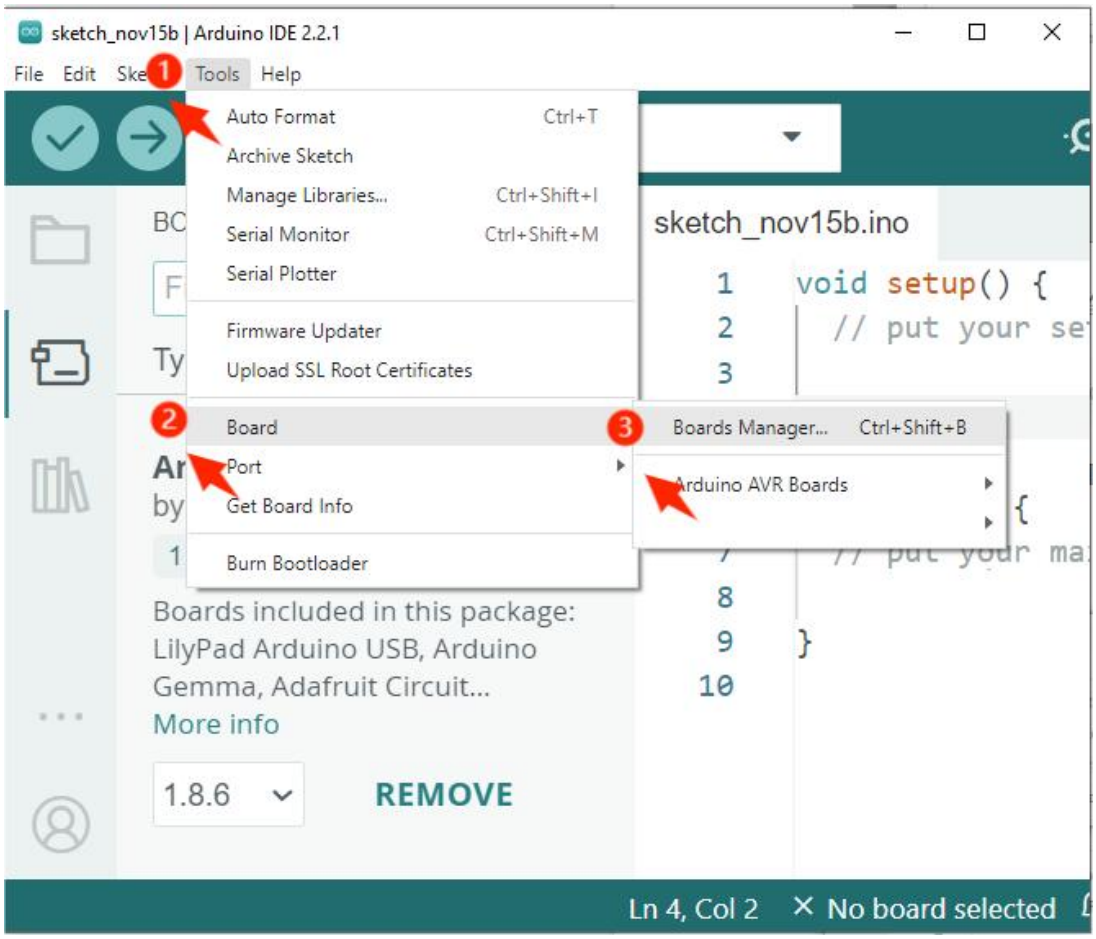



④ Klicken Sie nach dem Hinzufügen der URL auf „OK“.



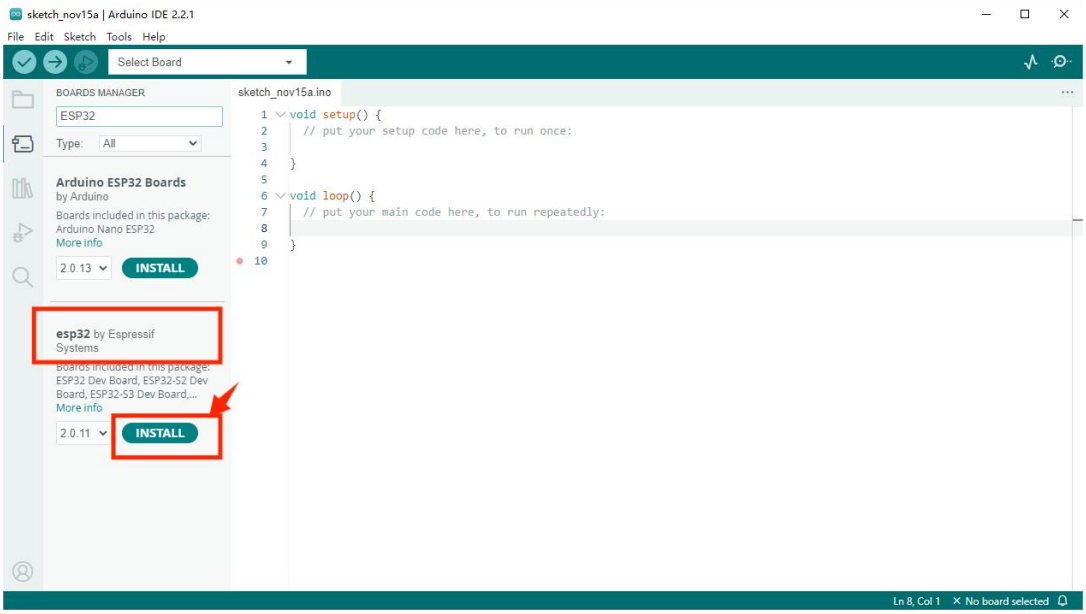




⑤ Klicken Sie auf „Tools“ > „Board“ > „Boards Manager“.

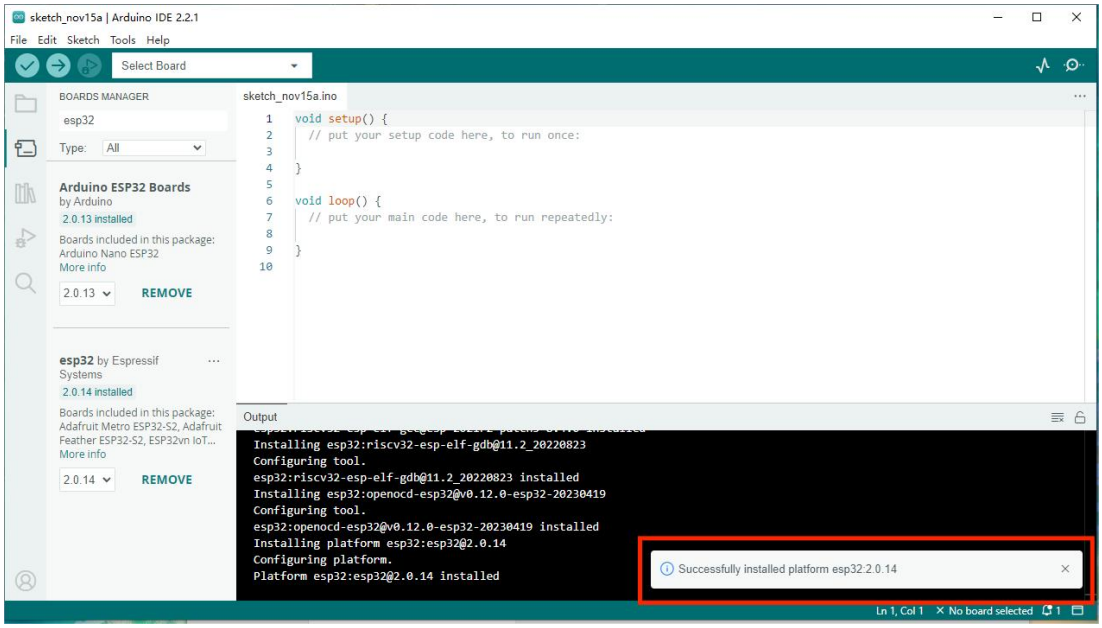



⑥ Suchen Sie im Boards Manager nach „ESP32“ und installieren Sie es.

Hinweis: Installieren Sie bitte Version 2.0.12 von esp32, da neuere Versionen nicht mit den Bibliotheken des Tutorials kompatibel sind und zu Fehlern führen können. Falls bereits Version 3.0 installiert ist, deinstallieren Sie diese und installieren Sie erneut Version 2.0.12.

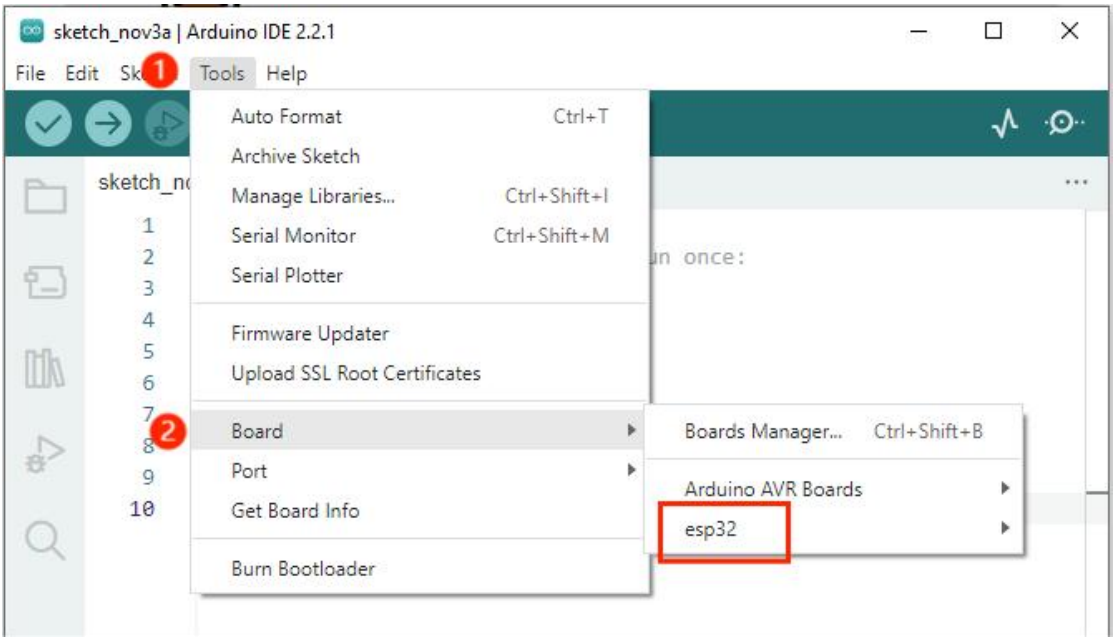



⑦ Warten Sie, bis die Installation abgeschlossen ist, und schließen Sie die Arduino IDE.




Hinweis: Da das Installationspaket auf GitHub veröffentlicht ist, kann die Installation von der Netzwerkgeschwindigkeit beeinflusst werden. Falls die Installation fehlschlägt, versuchen Sie es bitte mehrmals.

⑧ Öffnen Sie die Arduino IDE erneut und wählen Sie „Tools“ > „Board“. Sie werden sehen, dass das esp32-Board hinzugefügt wurde.



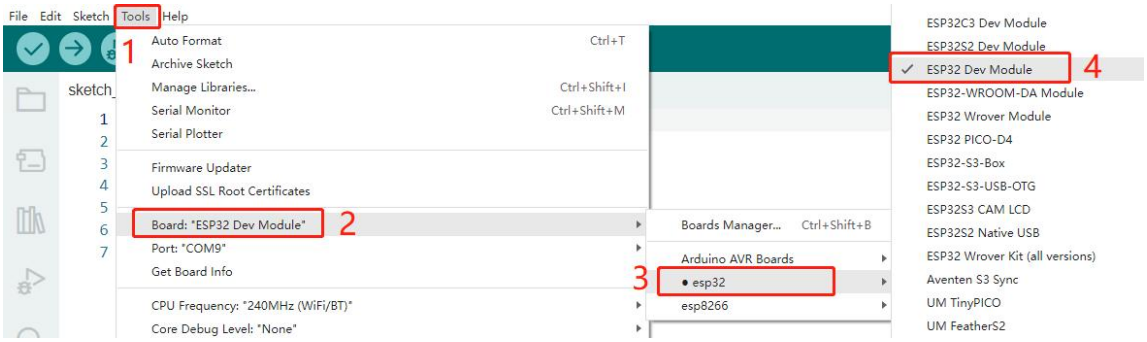

## 4.Hinzufügen von Bibliotheksdateien

Öffnen Sie die Datei „Bibliotheken hinzufügen“ im Ordner „Deutsch\Arduino(Erfahrener Lernender)\4.Bibliotheken hinzufügen“ und fügen Sie die Bibliotheken gemäß den Anweisungen hinzu.

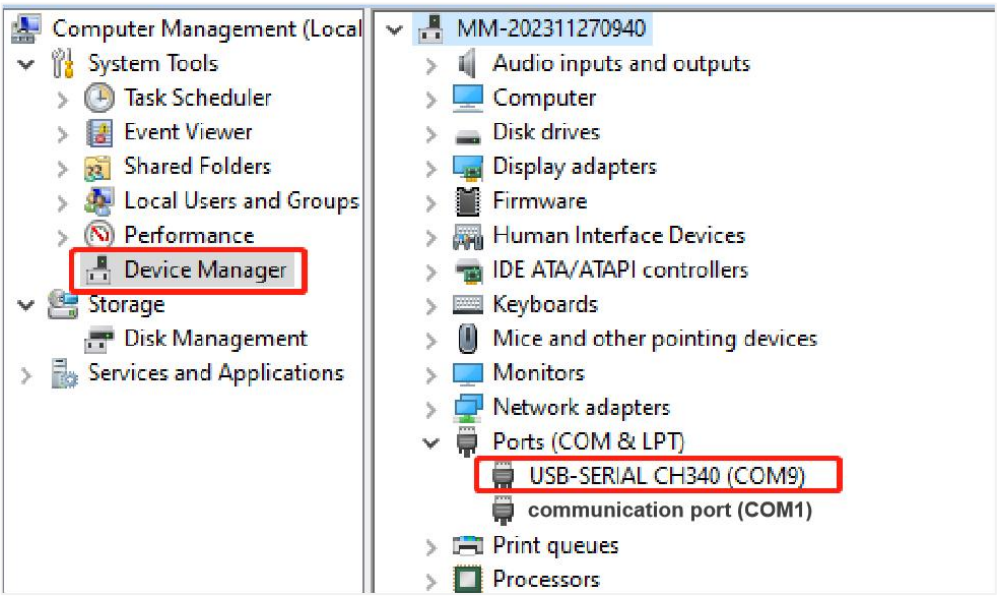
## 5.Testen der Entwicklungsumgebung

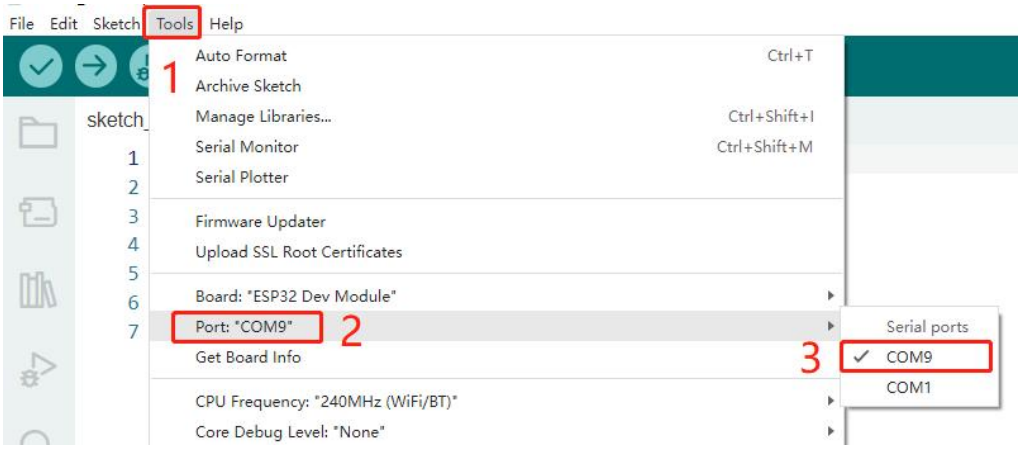
Nach der Installation der Arduino IDE und der Erweiterungsbibliothek für das ESP32-Entwicklungsboard können Sie mit einem einfachen Programm testen, ob die Entwicklungsumgebung erfolgreich eingerichtet wurde. Gehen Sie dazu wie folgt vor:

① Verbinden Sie die Steuerplatine mit dem Computer → Öffnen Sie die Arduino IDE → Klicken Sie auf „Tools“ → Wählen Sie „esp32“ → Wählen Sie „ESP32 Dev Module“.

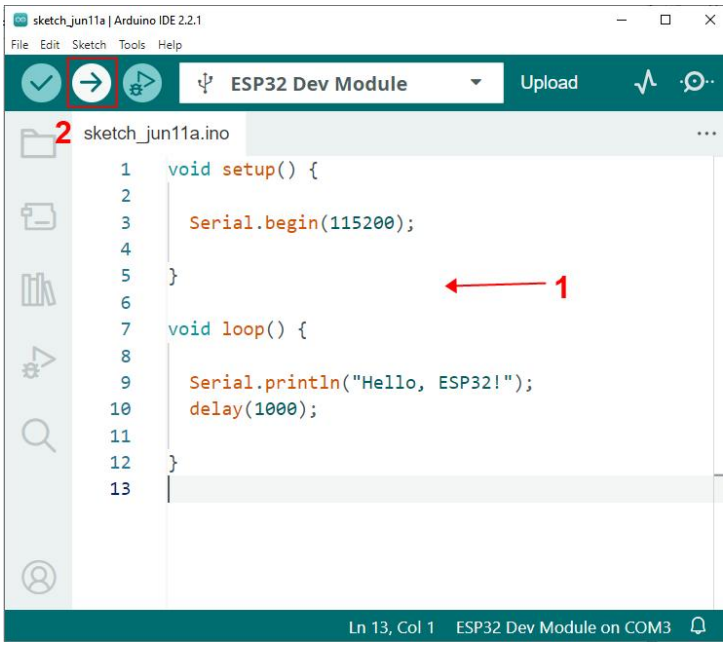


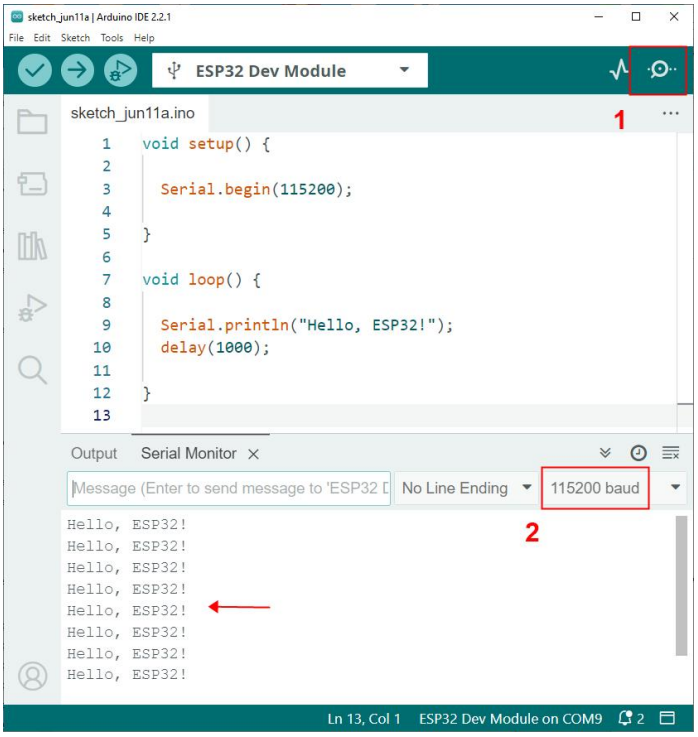

② Wählen Sie den seriellen Port (Sie können die Portnummer im Geräte-Manager überprüfen und dann den entsprechenden Port auswählen, da jede Platine unterschiedliche COM-Nummern hat).



③ Öffnen Sie „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 1\Hello_esp32“ die Datei „Hello_esp32.ino“, verbinden Sie das ESP32-Entwicklungsboard mit einem USB-Kabel mit dem Computer, wählen Sie das richtige Entwicklungsboard und den richtigen Port aus und laden Sie den Code auf das ESP32-Entwicklungsboard hoch. Wählen Sie eine Baudrate von 115200, dann können Sie im seriellen Monitor fortlaufend „Hello, ESP32!“ sehen.



## III. Servomotor kennenlernen

### 1. Einführung in Servomotor

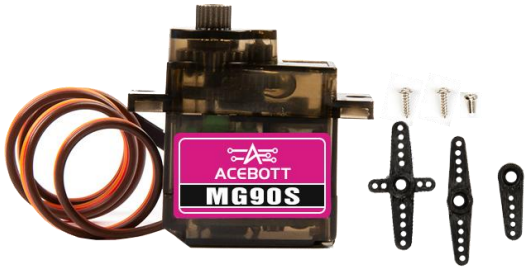
Die Hauptstruktur des Servomotors ist in der folgenden Abbildung dargestellt und besteht hauptsächlich aus mehreren Teilen: Gehäuse, Getriebe, Motor, einstellbares Potentiometer, Steuerplatine und Servohorn.

Sein Funktionsprinzip ist wie folgt: Die Steuerplatine empfängt das Steuersignal von der Signalquelle und treibt den Motor zur Drehung an; das Getriebe reduziert die Drehzahl des Motors um ein Vielfaches und verstärkt gleichzeitig das vom Motor ausgegebene Drehmoment entsprechend, bevor es ausgegeben wird; das Potentiometer dreht sich zusammen mit der letzten Stufe des Getriebes und misst den tatsächlichen Drehwinkel der Servowelle; die Steuerplatine empfängt den vom Potentiometer zurückgemeldeten tatsächlichen Winkel des Motors und vergleicht ihn mit dem Zielwinkel. Wenn eine Abweichung besteht, steuert sie den Servomotor so, dass er sich auf die Zielwinkelposition dreht.

Sein Arbeitsablauf ist: Steuersignal → elektronische Steuerplatine → Motordrehung → Getriebereduktion → Drehung des Servohorns → Rückmeldung des tatsächlichen Motorwinkels → die Steuerplatine passt anhand der Rückmeldung die Motorposition auf den Zielwinkel an.

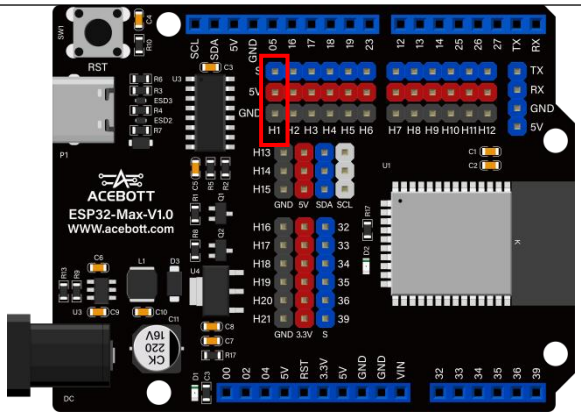
### 2. Servomotor-Pinbelegung

① In der Regel hat ein Servomotor 3 Steuerleitungen: Stromleitung, Erdungsleitung und Signalleitung.



② Pinbelegung des Servomotors: braunes Kabel — GND, rotes Kabel — 5V, oranges Kabel — Signal.

③ Verbinden Sie den Servomotor mit dem ESP32-Entwicklungsboard. Die Verbindungsweise ist wie unten dargestellt.

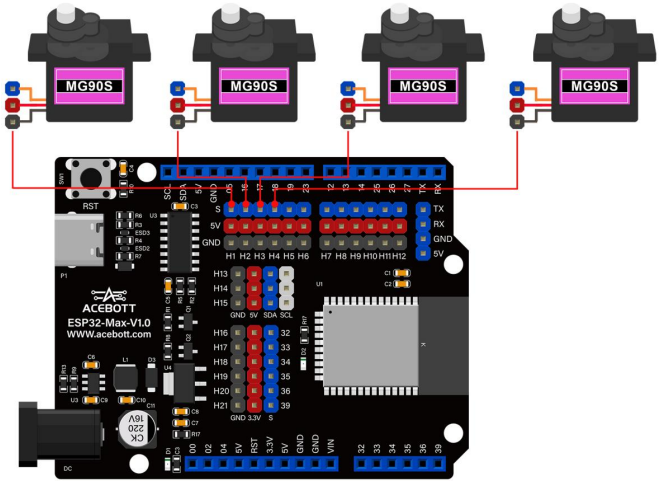
| Servomotor | ESP32 Max V1.0 Controller Board | Diagramm |
|---|---|---|
| Braune Leitung | GND |  |
| Rote Leitung | 5V | |
| Orange Leitung | GPIO5 | |



Hinweis: Bitte befolgen Sie die Verdrahtungsanweisungen genau, um das Modul mit der ESP32-Steuerplatine zu verbinden. Eine falsche Verdrahtung kann einen Kurzschluss und Schäden an der ESP32-Steuerplatine verursachen.

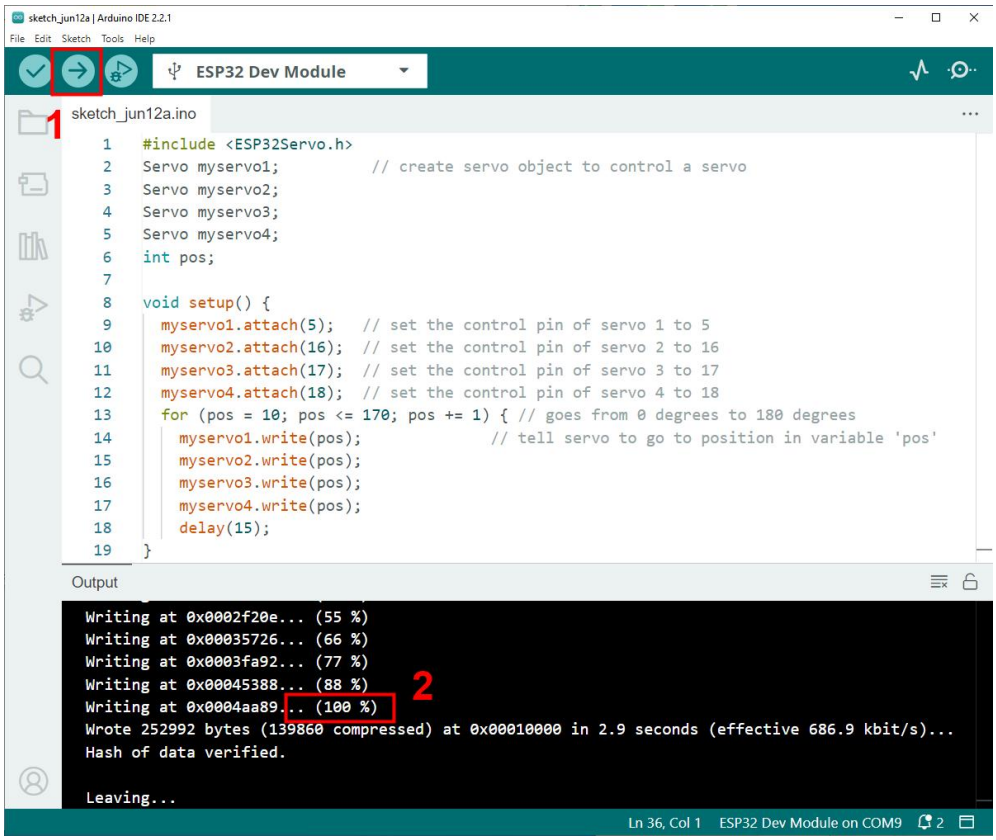
### 3. Servomotor-Test

① Verbinden Sie wie in der Abbildung gezeigt vier Servomotoren.



② Öffnen Sie „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 1\Servo_test“ und dort „Servo_test.ino“. Verbinden Sie das ESP32-Entwicklungsboard mit einem USB-Kabel mit dem Computer, wählen Sie den „Upload-Modus“, wählen Sie das richtige Entwicklungsboard, den Prozessor und den Port aus und laden Sie den Code auf das ESP32-Entwicklungsboard hoch.

Hinweis: Um eine stabile Stromversorgung zu gewährleisten, muss ein 18650-Akku in das Batteriefach eingesetzt werden. Schließen Sie dieses anschließend an den Stromanschluss der Hauptplatine an und stellen Sie den Schalter auf „ON“.




③ Wenn der Servomotor normal ist, dreht er von 0 Grad auf 180 Grad, dann von 180 Grad auf 0 Grad und zum Schluss auf die 90°-Position.

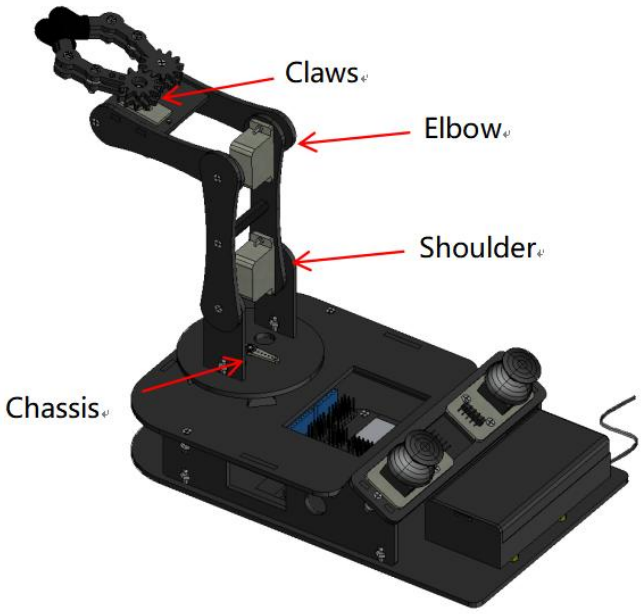
**4. Servomotor-Rückstellprogramm**

Vor dem Zusammenbau des Roboterarms müssen wir zur reibungslosen Montage der Struktur das Servomotor-Rückstellprogramm im Voraus hochladen.

Öffnen Sie „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 1\servo_90“ und darin „servo_90.ino“. Verbinden Sie das ESP32-Entwicklungsboard über ein USB-Kabel mit dem Computer, wählen Sie das richtige Entwicklungsboard, den Prozessor und den Port aus und laden Sie den Code auf das ESP32-Entwicklungsboard hoch.

# Lektion 2 Zusammenbau des Roboterarms

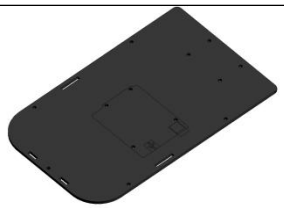
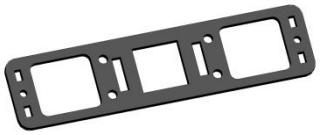
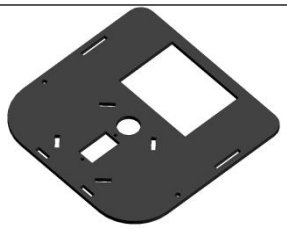
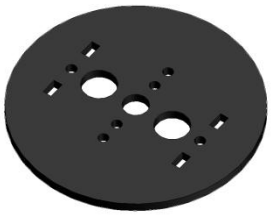
Vor dem Zusammenbau haben wir die Servomotoren – um das Montieren zu erleichtern und ihre Positionen eindeutig zu identifizieren – von oben nach unten wie folgt benannt: „Claws“, „Elbow“, „Shoulder“, „Chassis“.

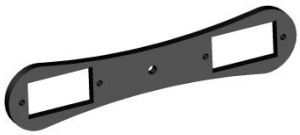
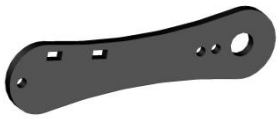
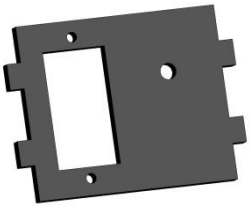



## I. Zubehörliste



# II. Bauteilliste

| Acrylstruktur | Menge | Bilder |
|---|---|---|
| Bodenplatte | 1 |  |
| Bodenblattfestiger | 1 |  |
| | 1 |  |
| | 1 |  |
| Joystick-Befestigungsplatte | 1 |  |
| Chassis-Servomotor-Montageplatte | 1 |  |
| Steckbolzen | 4 |  |
| Chassis-Scheibe | 1 |  |
| Scheibenhalter1 | 1 |  |

| | | |
|---|---|---|
| Scheibenhalter2 | 1 |  |
| Schulter 1 | 1 |  |
| Schulter 2 | 1 |  |
| Ellbogen 1 | 1 |  |
| Ellbogen 2 | 1 |  |
| Montageplatte für Klauenservo | 1 |  |
| Greiferkomponenten1 | 2 |  |
| Greiferkomponenten2 | 2 |  |

## III. Montageschritte

Hinweis: Wenn Sie ein Assembly-Video ansehen möchten, klicken Sie bitte auf den untenstehenden Link und wählen Sie das entsprechende Build-Video aus.

https://www.youtube.com/playlist?list=PLkW5fEtHNu6JIQVJpGalnK

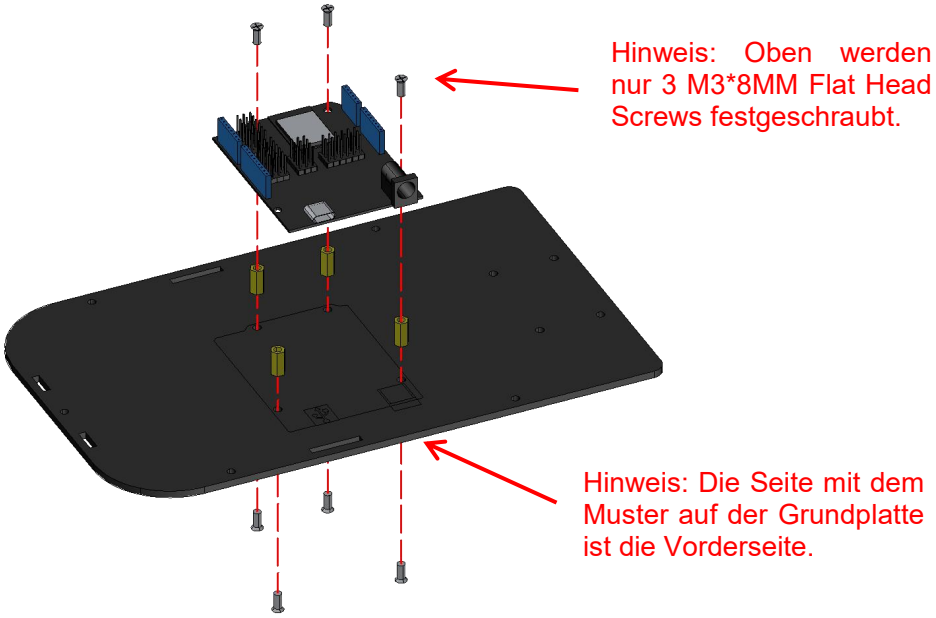
dkuk6Er4dIN

Oder scannen Sie den unten stehenden QR-Code.

**1. Ziehen Sie das auf den Acryl-Strukturteilen haftende Schutzpapier ab.**

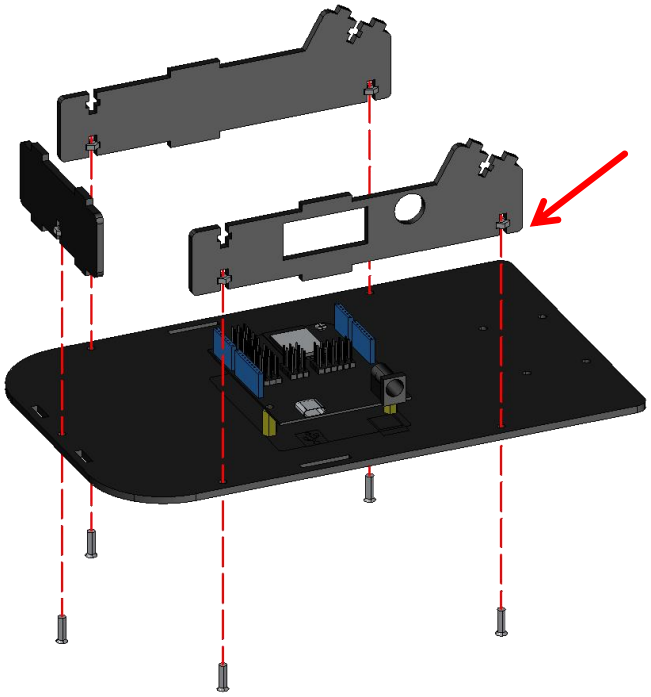
**2. Installieren Sie das ESP32-Entwicklungsboard auf der Grundplatte.**

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| ESP32 Max V1.0 Controller Board | 1 |
| Bodenplatte | 1 |
| MM3*8MM Flat Head Screws | 7 |
| M3*12MM Dual-pass Copper Pillar | 4 |



## 3. Installieren der Grundplattenhalterung.

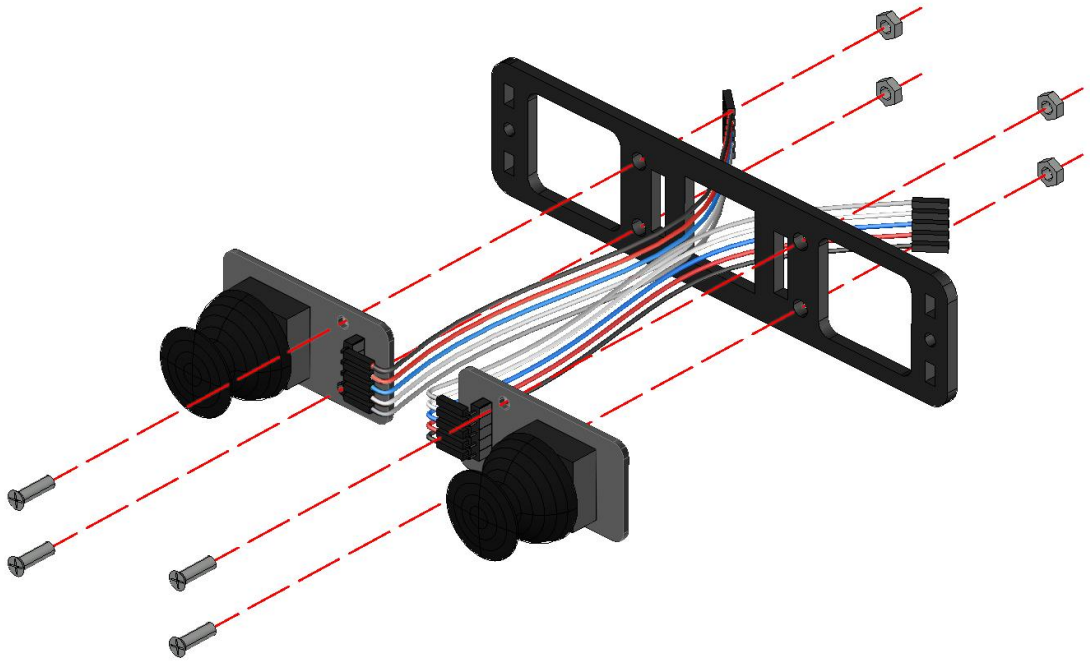
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Bodenblattfestiger | 3 |
| M3*10MM Flat Head Screws | 5 |
| M3 Nickel-Plated Nuts | 5 |



Hinweis: Setzen Sie die M3 Nickel-Plated Nuts hier ein. Halten Sie die Mutter an beiden Seiten mit den Fingern fest und ziehen Sie dann die Schraube fest.

## 4. Installieren Sie das Joystick-Modul.

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Joystick-Module | 2 |
| Joystick-Befestigungsplatte | 1 |
| M3*10MM Flat Head Screws | 4 |
| M3 Nickel-Plated Nuts | 4 |



In diesem Schritt müssen zuerst die Kabel der beiden Joystick-Module mit dem Motherboard verbunden werden. Verbinden Sie für den linken Joystick SW mit Pin 34, X mit Pin 32, Y mit Pin 33 sowie VCC und GND mit den entsprechenden VCC- und GND-Pins in der Pinreihe 32. Verbinden Sie für den rechten Joystick SW mit Pin 39, X mit Pin 35, Y mit Pin 36 sowie VCC und GND mit den entsprechenden VCC- und GND-Pins in der Pinreihe 35.

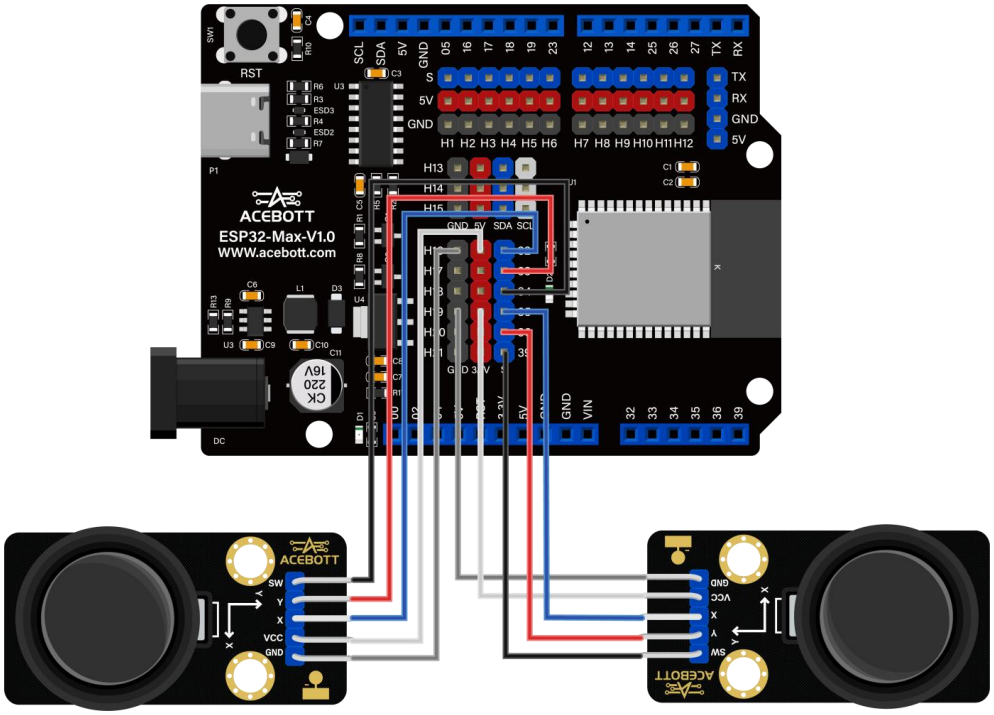
| **Pins des Joystick-Moduls** | **Pins des ESP32 Max V1.0 Controller Board** |
|---|---|
| Linker Joystick GND | GND |
| Linker Joystick VCC | 5V |
| Linker Joystick X | 32 |
| Linker Joystick Y | 33 |
| Linker Joystick SW | 34 |
| Rechter Joystick GND | GND |
| Rechter Joystick VCC | 5V |
| Rechter Joystick X | 35 |
| Rechter Joystick Y | 36 |
| Rechter Joystick SW | 39 |

Hinweis:

1. Wenn Sie das Joystick-Module und das ESP32-Entwicklungsboard mit DuPont-Kabeln verbinden, wird das Ende des DuPont-Kabels mit zusammengeführten Leitungen (5 zusammengeführte Kabel) an das Joystick-Module angeschlossen.

2. Das aufgeteilte Ende des DuPont-Kabels (aufgeteilt in 2 zusammengeführte Kabel und 3 zusammengeführte Kabel) wird an das ESP32-Entwicklungsboard angeschlossen. Der Teil mit 2 zusammengeführten Kabeln wird an 5V und GND angeschlossen; der Teil mit 3 zusammengeführten Kabeln wird an die GPIO-Ports angeschlossen.

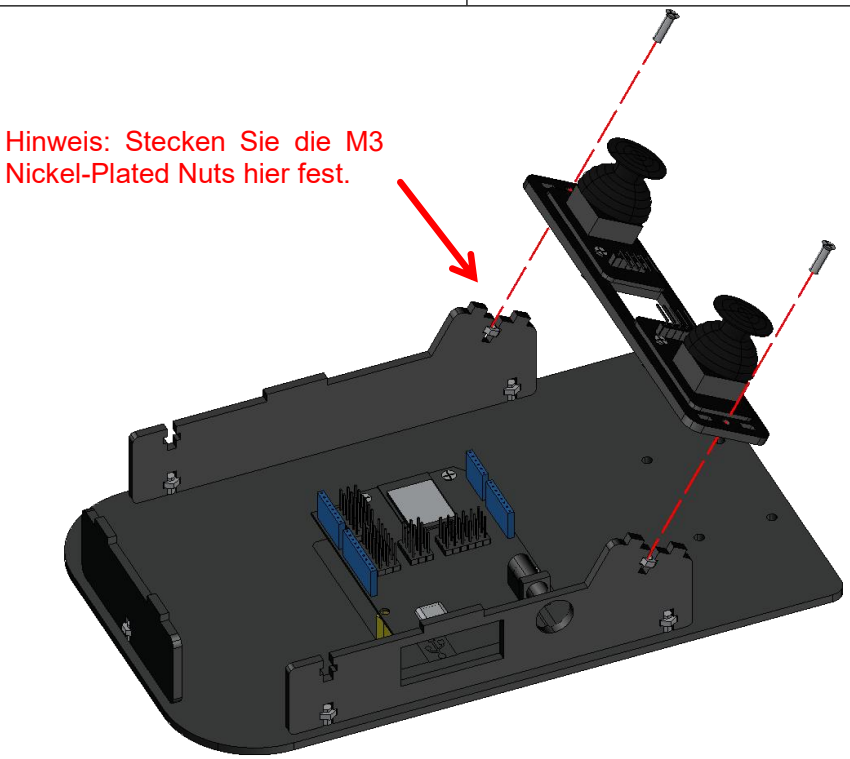
3. Bitte beachten Sie beim Verdrahten unbedingt die Farbreihenfolge, um eine sichere und korrekte Verbindung zu gewährleisten.




## 5. Installieren der Joystick-Befestigungsplatte.

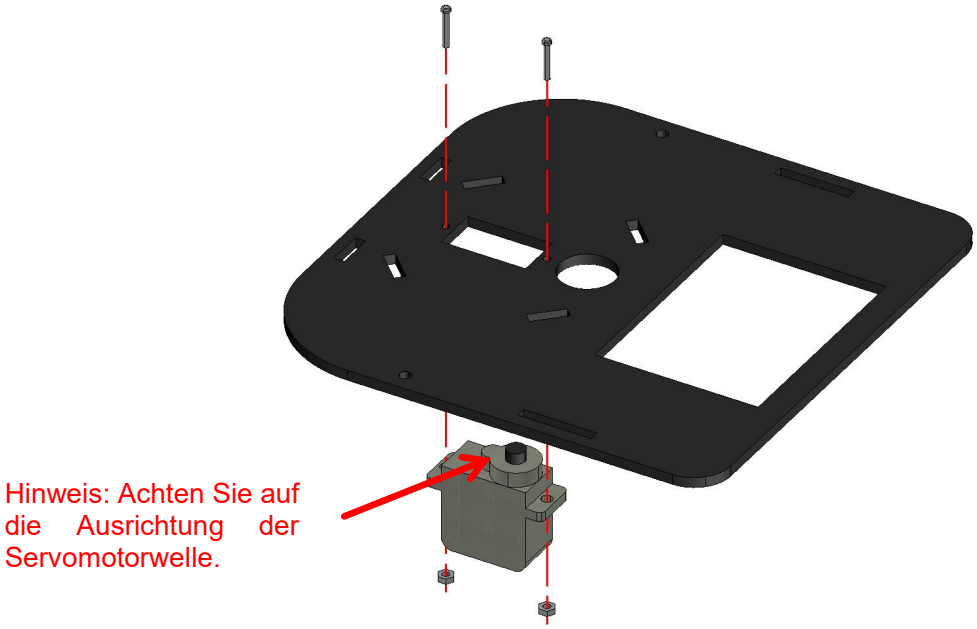
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| M3*10MM Flat Head Screws | 2 |

| | |
|---|---|
| M3 Nickel-Plated Nuts | 2 |




## 6. Installieren des Chassis-Servomotor.

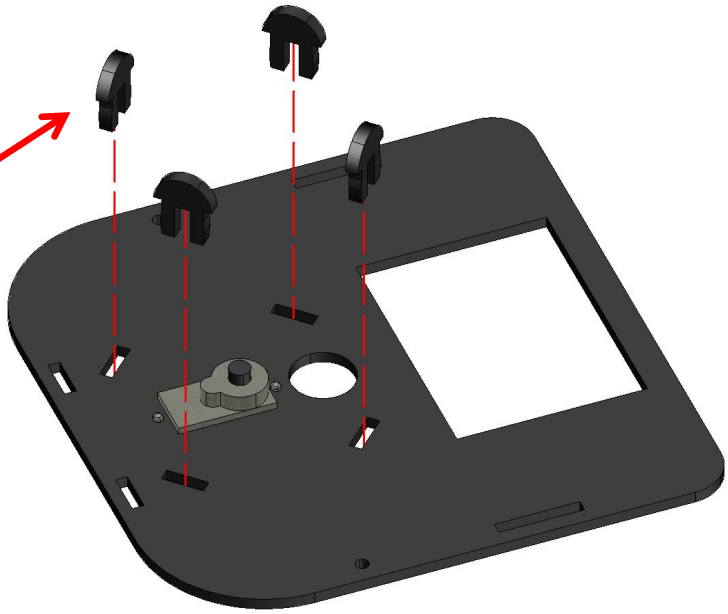
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Servomotor | 1 |
| Chassis-Servomotor-Montageplatte | 1 |
| M2*10MM Round Head Screws | 2 |
| M2 Nickel-Plated Nuts | 2 |



## 7. Installieren des Steckbolzens.

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Steckbolzen | 4 |

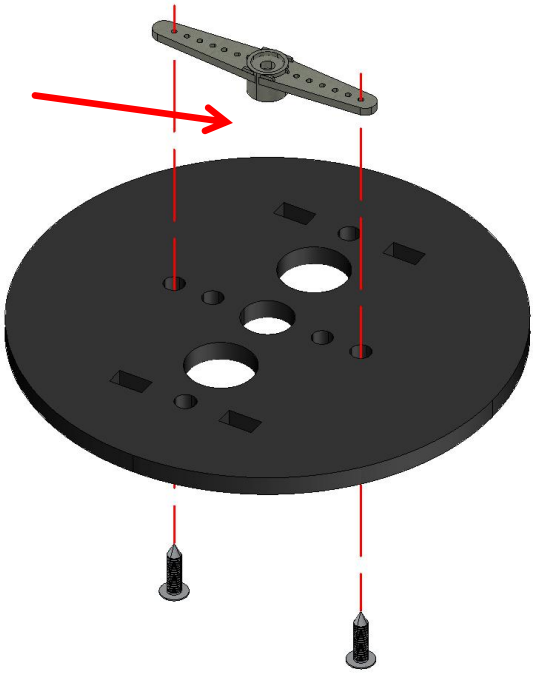
Hinweis: Beim Eindrücken des Steckbolzens nicht zu viel Kraft anwenden, um ein Brechen zu vermeiden.



## 8. Installieren des Servohorns des Chassis-Servomotor.

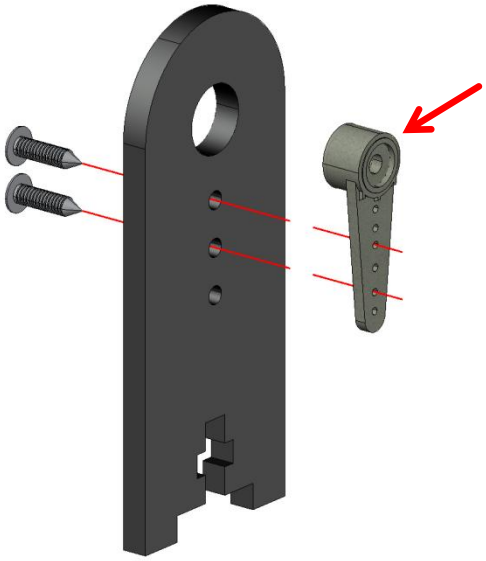
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Einarmiges Servohorn | 1 |
| Chassis-Scheibe | 1 |
| M1.7*6 Large Round Flat Head Tapping Screws | 2 |

Hinweis: Montieren Sie das Servohorn so, dass die hervorstehende Seite zum Rundloch zeigt.

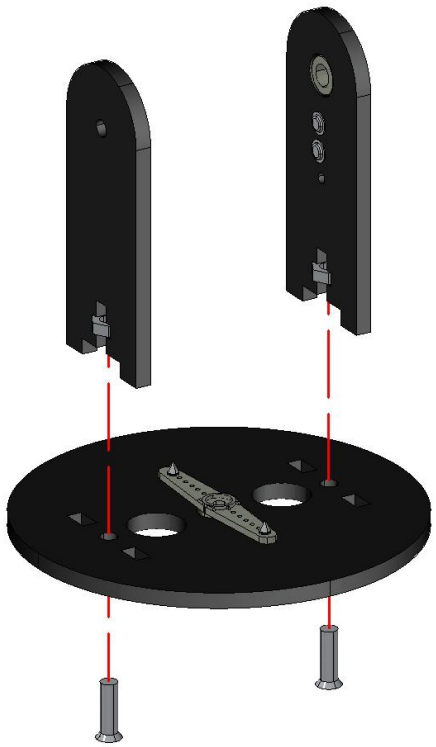


## 9. Installieren der Scheibenhalterung.

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Halbeinarmiges Servohorn | 1 |
| Scheibenhalterung1 | 1 |
| Scheibenhalterung2 | 1 |
| M1.7*6 Large Round Flat Head Tapping Screws | 2 |
| M3*10MM Flat Head Screws | 2 |
| M3 Nickel-Plated Nuts | 2 |



Hinweis:
Installieren Sie das Servohorn so, dass die hervorstehende Seite zum



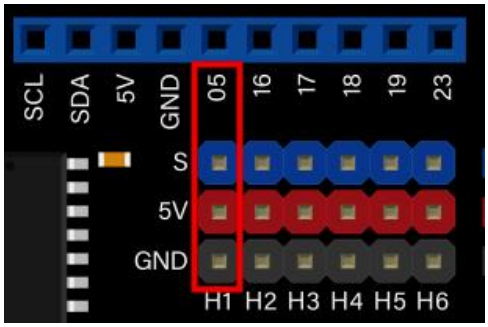
## 10. Befestigung der Roboterarm-Basisstruktur.

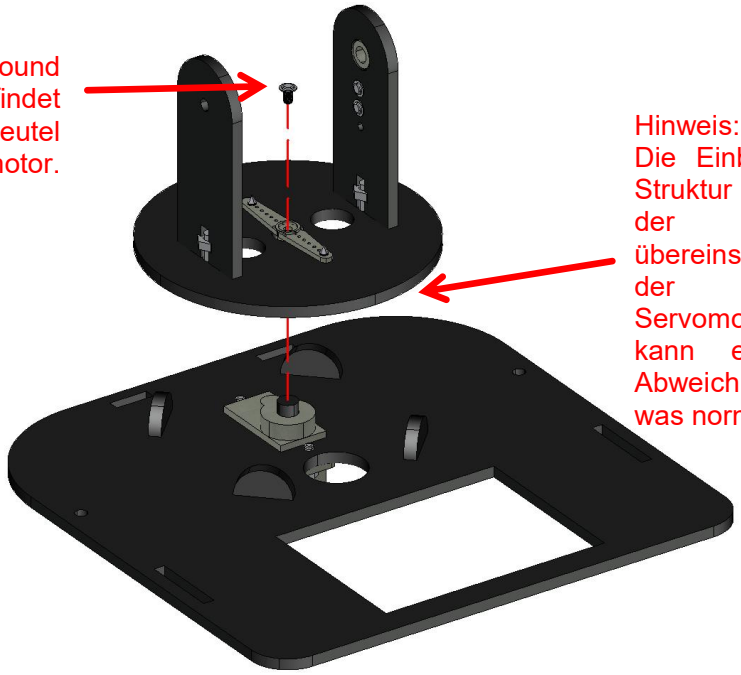
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| M2.5*4 Round Head Screw | 1 |

**Hinweis:**

**① Bevor Sie diesen Schritt ausführen, schließen Sie den Servomotor zuerst an den GPIO5-Pin des Boards an und schalten Sie dann die Stromversorgung des Boards ein, damit der Servomotor in der 90°-Position bleibt.**

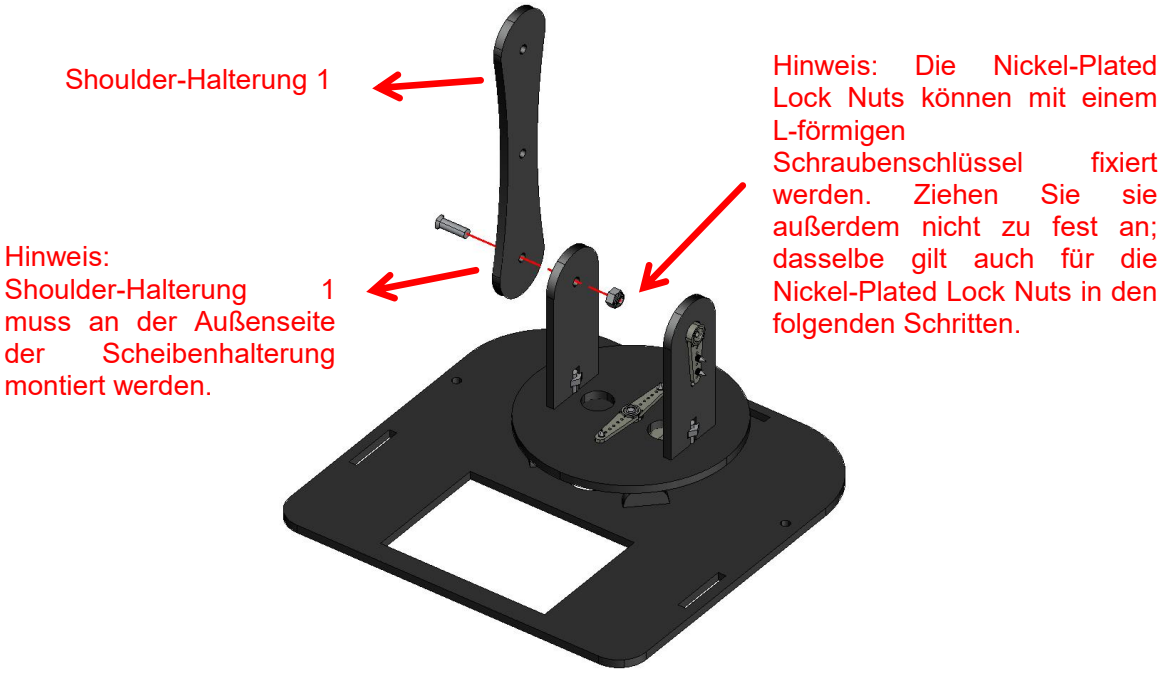
**② Um eine Beschädigung des Servomotors während der Installation zu vermeiden, drehen Sie bitte nicht an der Welle des Servomotors.**

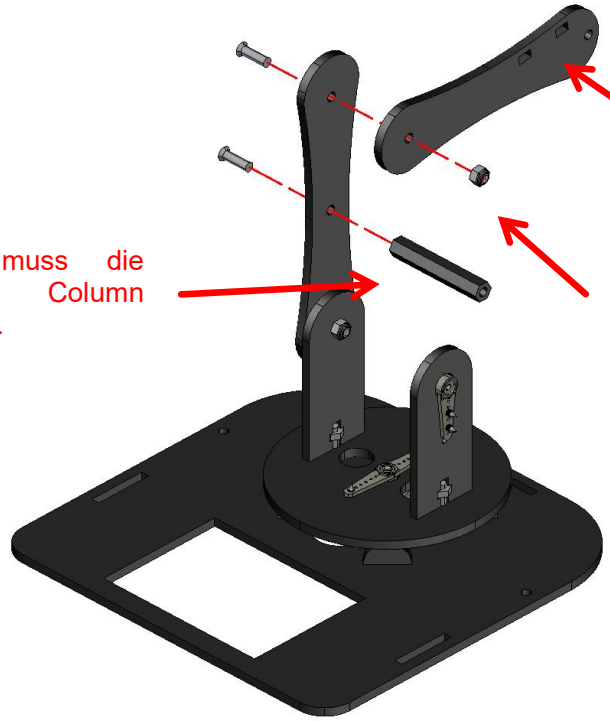




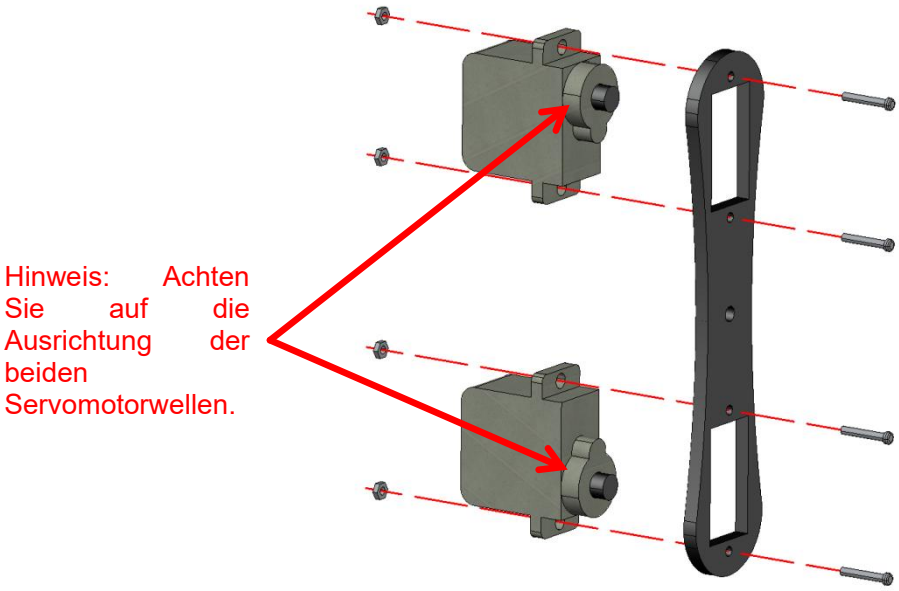

## 11. Installieren der Shoulder-Halterung.

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Shoulder-Halterung 1 | 1 |
| Elbow-Halterung 1 | 1 |
| M3*35 Nylon Column | 1 |
| M3*10MM Flat Head Screws | 3 |
| M3 Nickel-Plated Lock Nuts | 2 |






| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Servomotor | 2 |
| Shoulder-Halterung 2 | 1 |
| M2*10MM Round Head Screws | 4 |
| M2 Nickel-Plated Nuts | 4 |

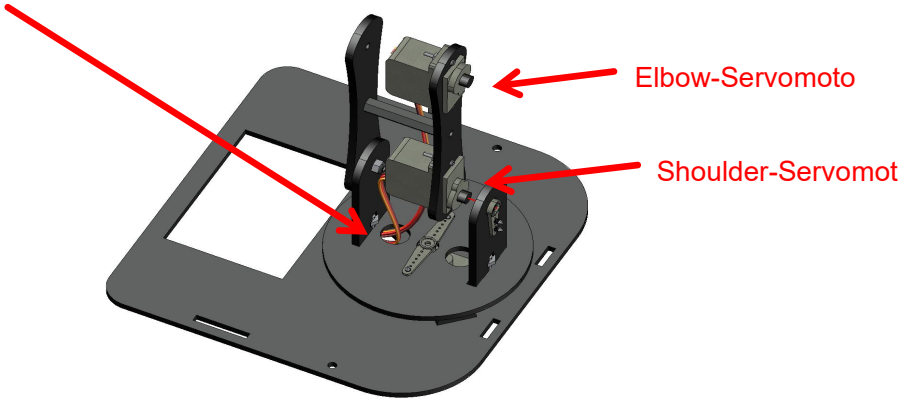



Verbinden Sie den Shoulder-Servomotor und den Elbow-Servomotor mit dem ESP32.

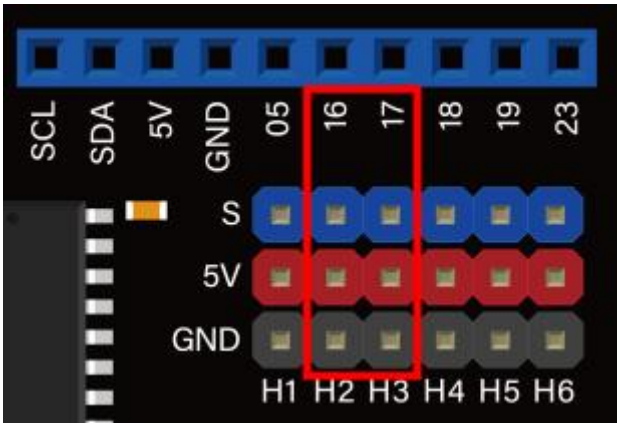
Hinweis:

**① Ziehen Sie zuerst das USB-Kabel ab, unterbrechen Sie die Stromversorgung des Boards und versetzen Sie den Servomotor des Roboterarms in einen stromlosen Zustand;**

② Drehen Sie dann das Kabelloch auf der Chassis-Scheibe vorsichtig so, dass es mit dem Kabelloch der Grundplatte ausgerichtet ist. Führen Sie anschließend die Dupont-Kabel des Shoulder-Servomotors und des Elbow-Servomotors des Roboterarms durch die beiden Kabellöcher.

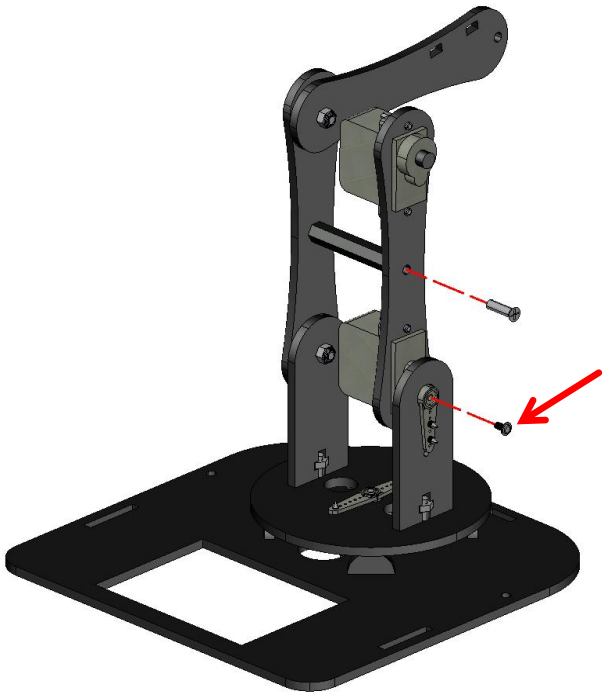



③ Schließen Sie das Kabel des Shoulder-Servomotors an den GPIO16-Pin an und das Kabel des Elbow-Servomotors an den GPIO17-Pin an.



**④ Schalten Sie dann die Stromversorgung des Boards erneut ein, damit der Chassis-Servomotor, der Shoulder-Servomotor und der Elbow-Servomotor in der 90°-Position bleiben.**

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| M2,5*4 Round Head Screw | 1 |
| M3*10MM Flat Head Screw | 1 |



Hinweis: Die Geste des Shoulder-Servo sollte senkrecht zum Boden montiert werden, bevor die Schraube des Ruders und der Nylonsäule verschlossen wird.

## 12. Befestigen der M3*40 Nylon Column

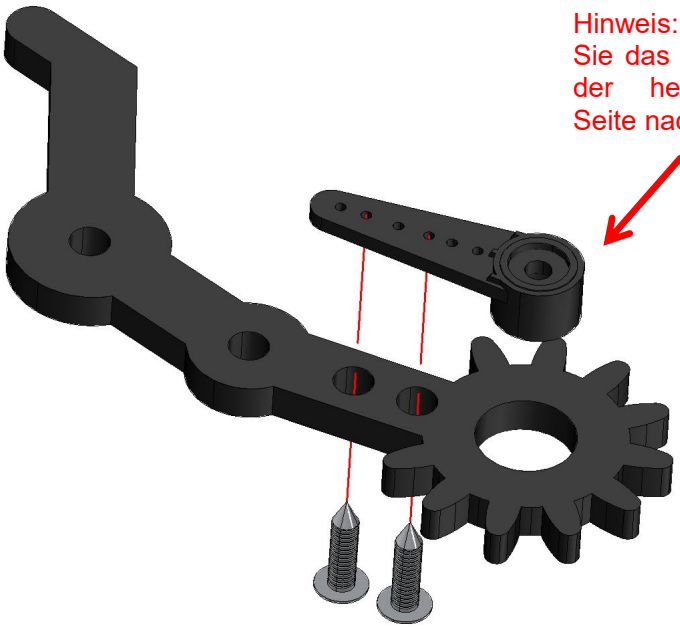
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| M3*40 Nylon Column | 1 |
| M3*10MM Flat Head Screw | 1 |



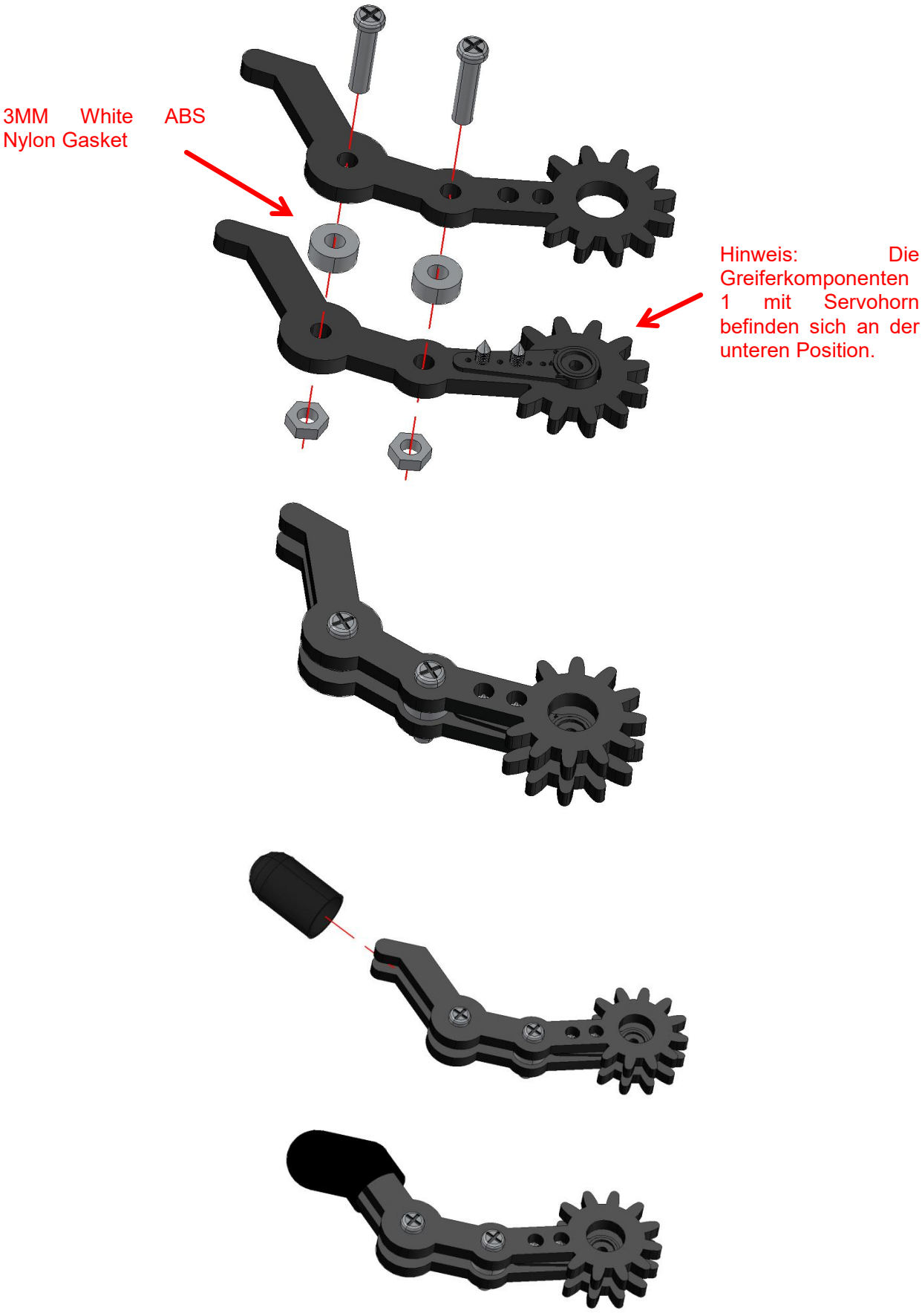
Hinweis: Verwenden Sie hier die längere Nylon Column.

## 13. Installieren Sie die linke Greiferstruktur des Roboterarms.

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Halbeinarmiges Servohorn | 1 |
| Greiferkomponenten 1 | 2 |
| M1,7*6 Large Round Flat Head Tapping Screws | 2 |
| M3*14 Round Head Screws | 2 |
| 3MM White ABS Nylon Gasket | 2 |
| M3 Nickel-Plated Lock Nuts | 2 |
| Non-Slip Sleeve | 1 |

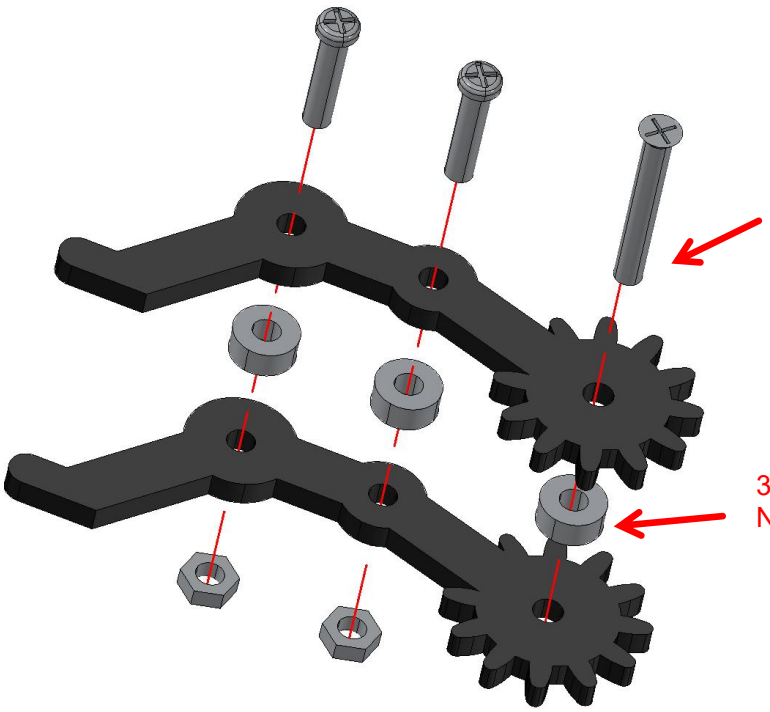


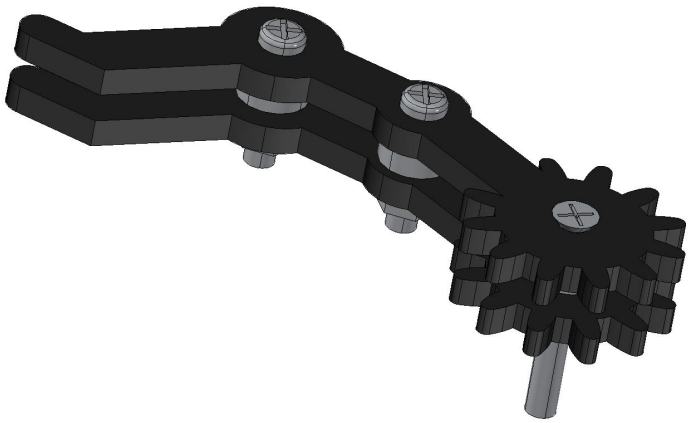
Hinweis: Installieren Sie das Servohorn mit der hervorstehenden Seite nach unten.

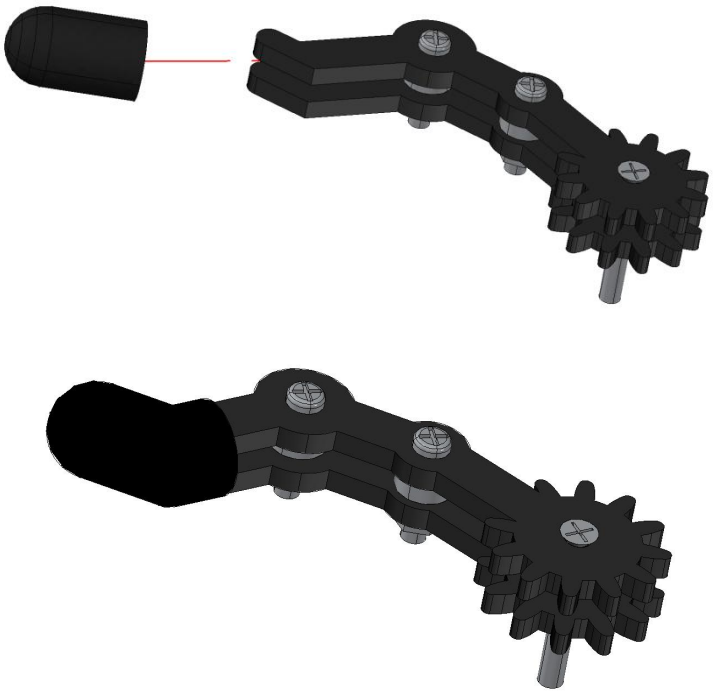

3MM White ABS Nylon Gasket
Hinweis: Die Greiferkomponenten 1 mit Servohorn befinden sich an der unteren Position.

## 14. Installieren Sie die rechte Greiferstruktur des Roboterarms.

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Greiferkomponenten 2 | 2 |
| M3*22 Flat Head Screws | 1 |
| M3*14 Round Head Screws | 2 |
| 3mm White ABS Nylon Gasket | 3 |
| M3 Nickel-Plated Lock Nuts | 2 |
| Non-Slip Sleeve | 1 |

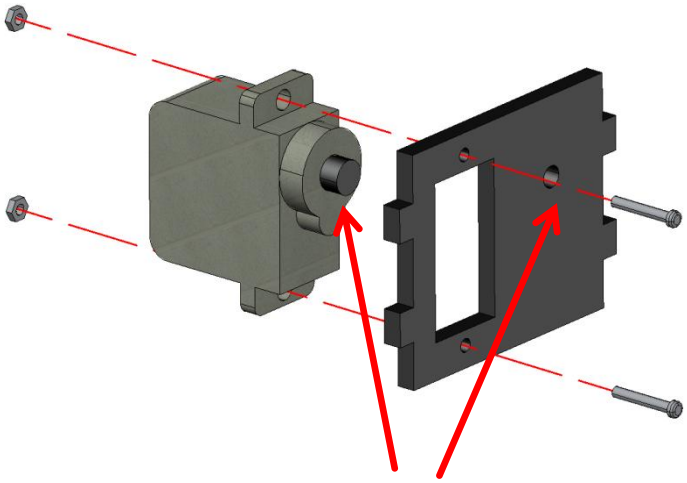




## 15. Greiferstruktur Des Roboterarms Befestigen

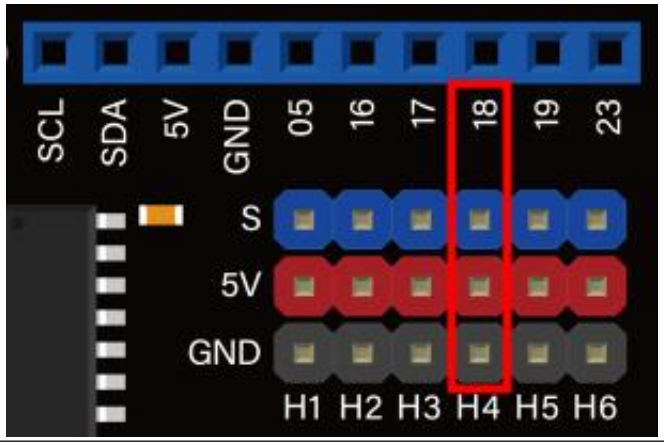
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Servomotor | 1 |
| Claws-Servomotor-Befestigungsplatte | 1 |
| M2*10 Flat Head Screws | 2 |
| M2 Nickel-Plated Lock Nuts | 2 |



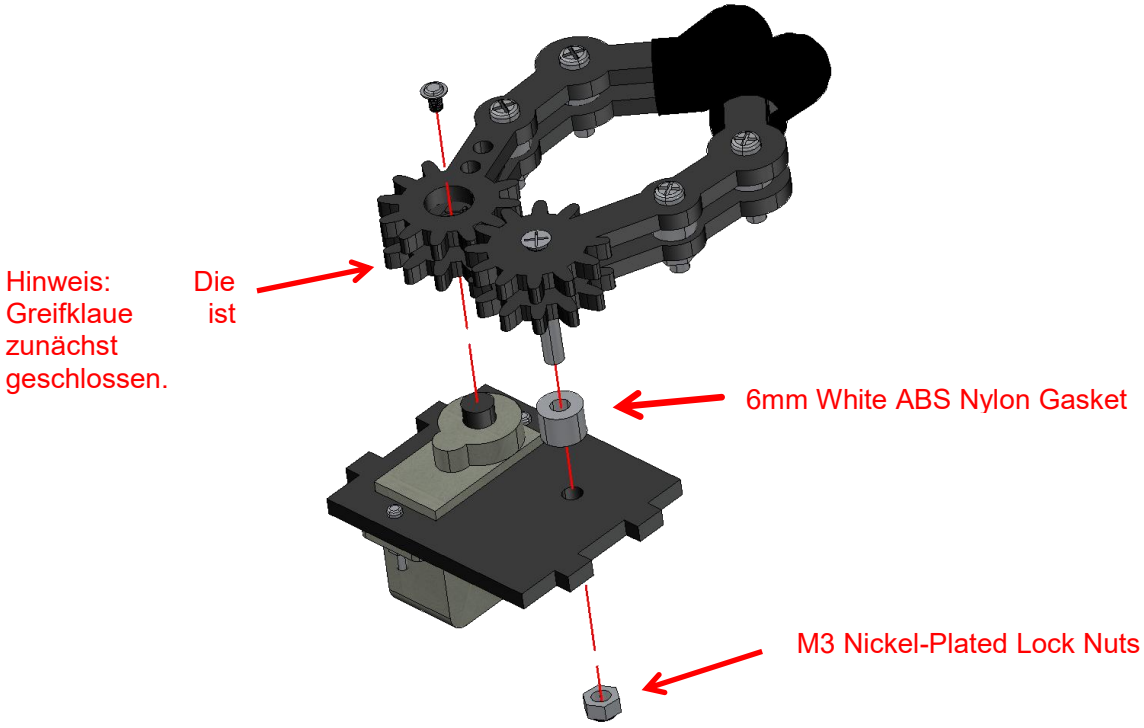
Hinweis: Sowohl die Servowelle als auch das Rundloch befinden sich oben.

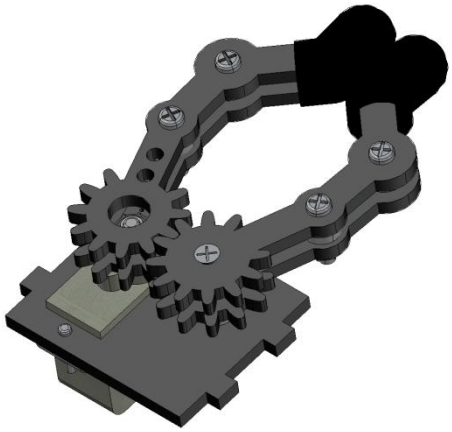
Schließen Sie das Kabel des Claws-Servomotors an den GPIO18-Pin an.

Hinweis: Schalten Sie in diesem Schritt bitte die Stromversorgung des Hauptboards ein, damit der Claws-Servomotor in der 90°-Position bleibt.



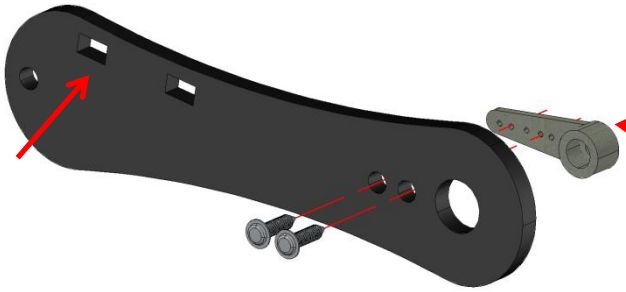

| Teileliste | |
|---|---|
| **Name** | **Menge** |
| M2,5*4 Round Head Screws | 1 |
| 6mm White ABS Nylon Gasket | 1 |
| M3 Nickel-Plated Lock Nuts | 1 |



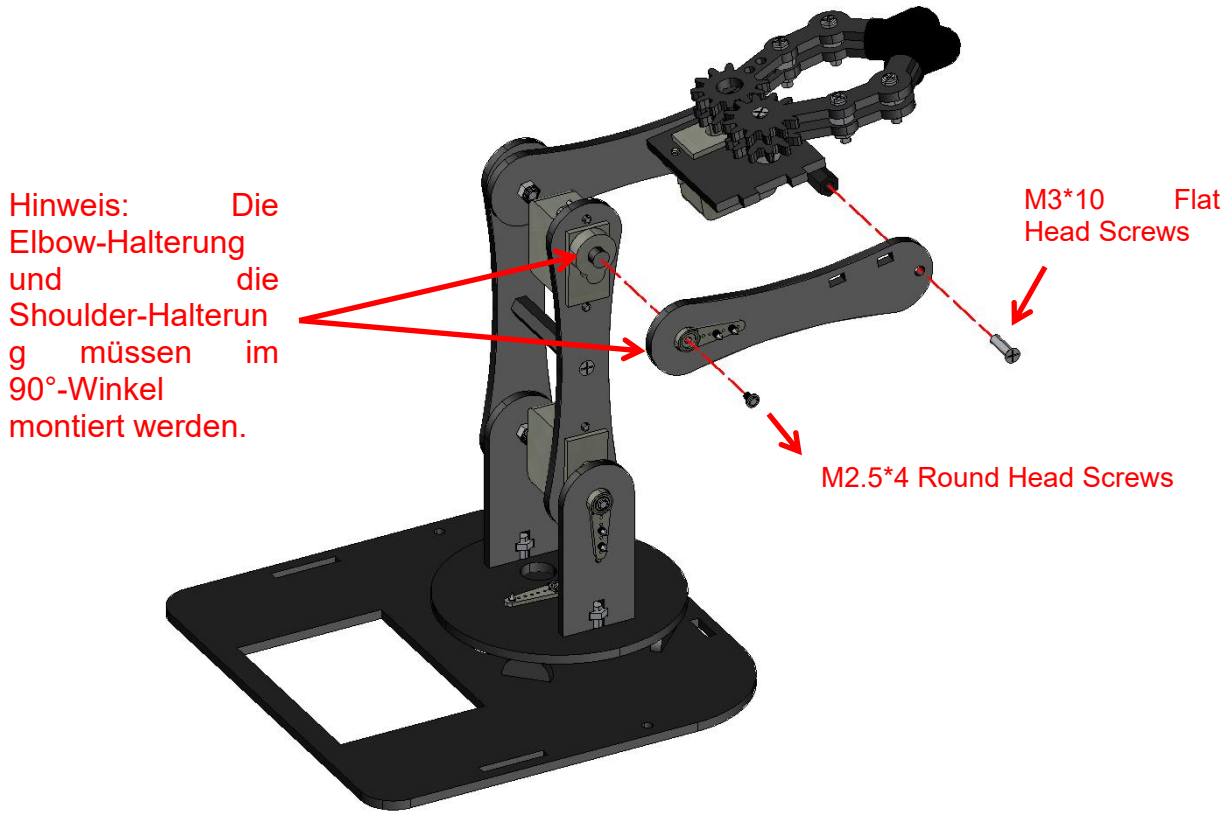


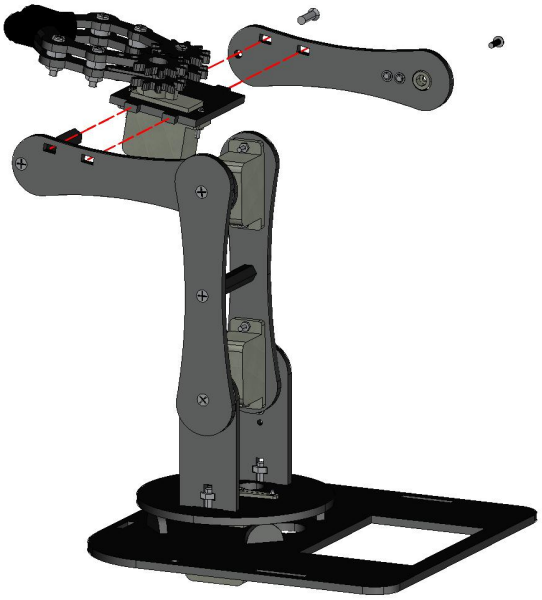
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| Halbeinarmiges Servohorn | 1 |
| M1.7*6 Large Round Flat Head Tapping Screws | 2 |
| Elbow-Halterung 2 | 1 |
| M3*10 Flat Head Screws | 1 |
| M2.5*4 Round Head Screws | 1 |

Hinweis: Die rechteckige Öffnung zeigt nach oben.



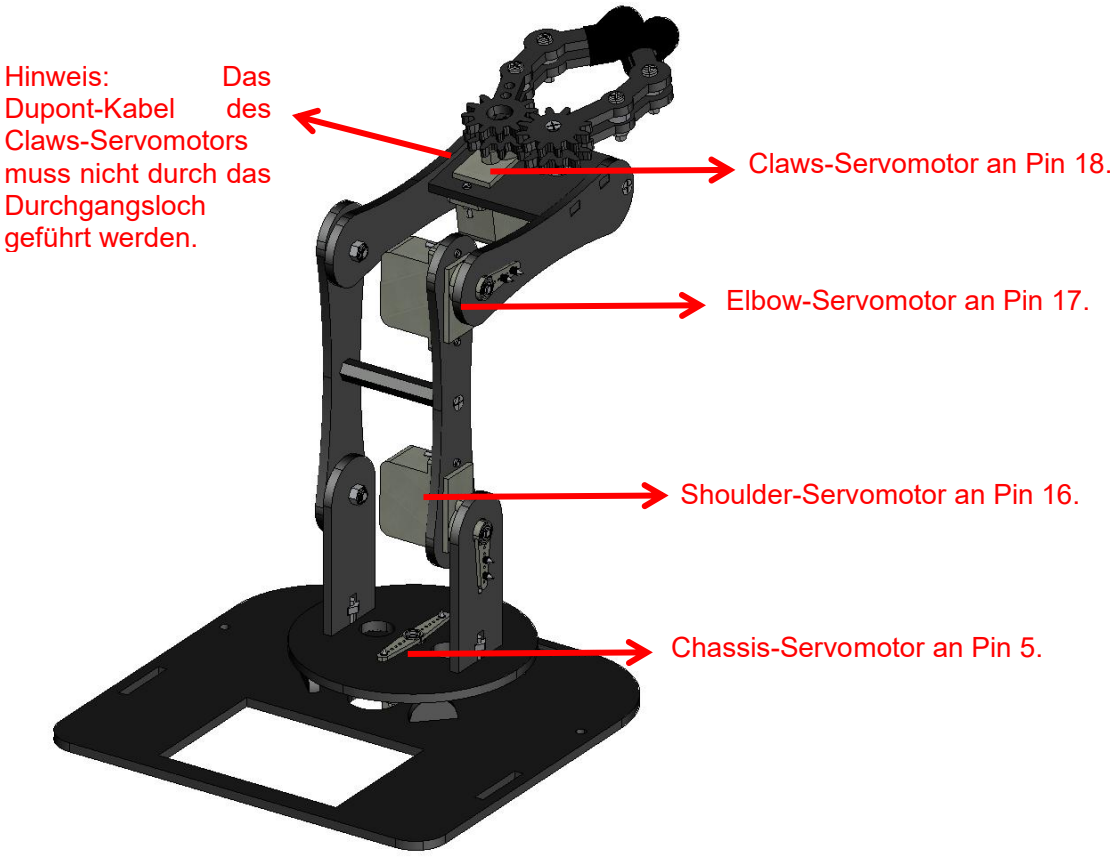
Hinweis: Die erhöhte Kante der Abstandssäule zeigt zur Rundlochseite.



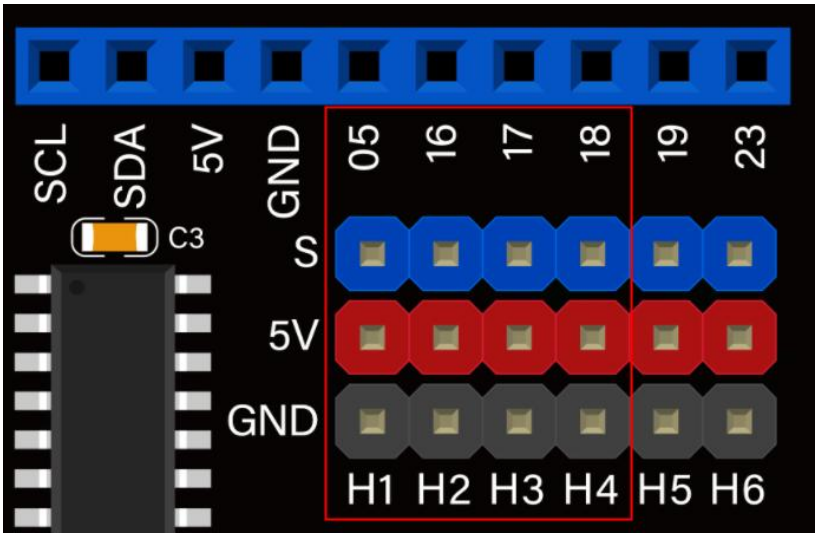



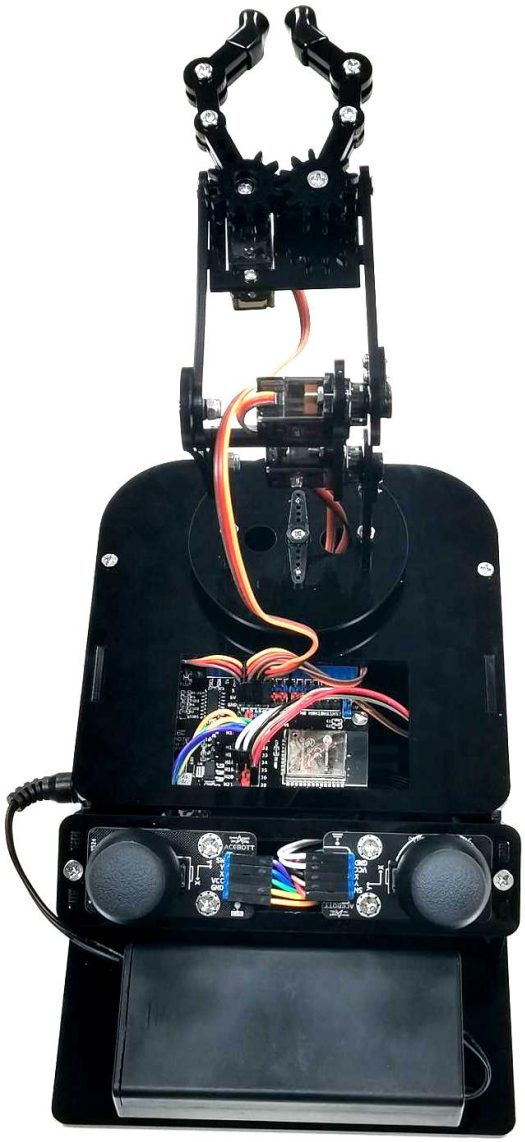
**16. Ordnen Sie die Servomotorkabel.**

Bitte vergewissern Sie sich, dass die Dupont-Kabel der vier Servomotoren korrekt mit dem Hauptboard verbunden sind.




Hinweis: Bitte befolgen Sie die Verdrahtungsanweisungen genau, um das Modul mit dem ESP32-Entwicklungsboard zu verbinden. Eine falsche Verdrahtung kann einen Kurzschluss sowie Beschädigungen des ESP32-Entwicklungsboards verursachen.

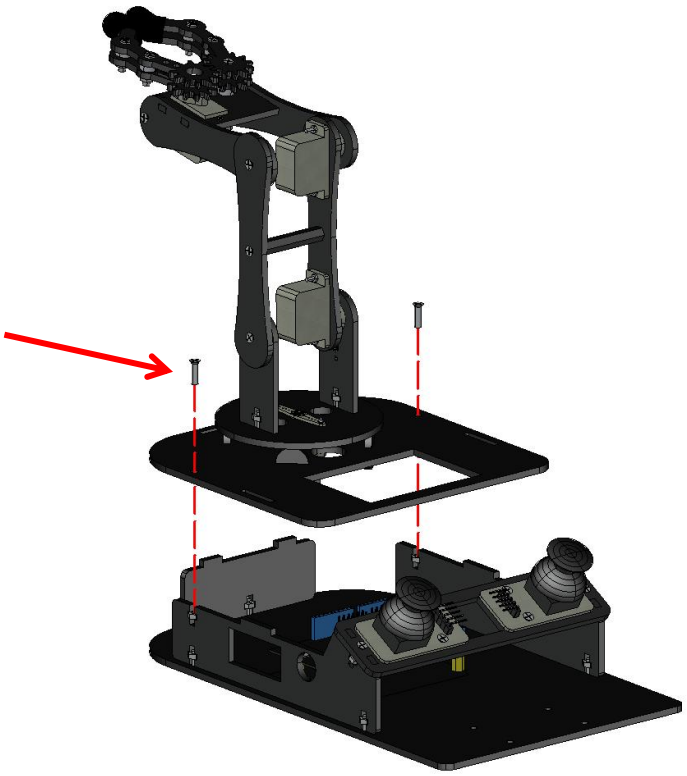




## 17. Roboterarm-Basis befestigen

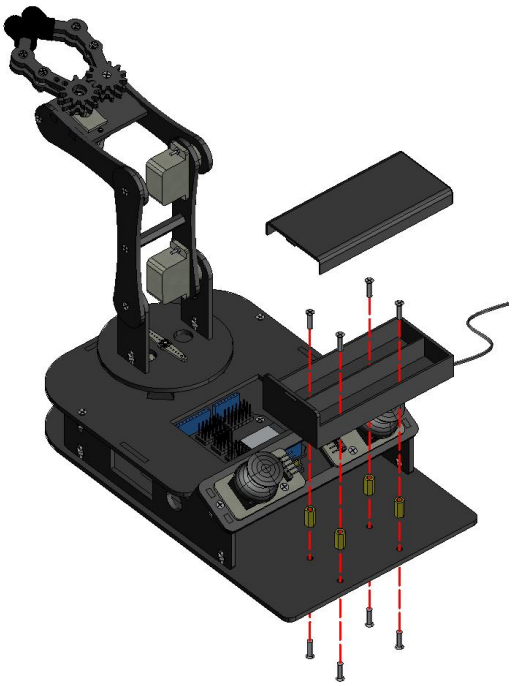
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| M3*10 Flat Head Screws | 2 |
| M3 Nickel-Plated Lock Nuts | 2 |

Hinweis: Stellen Sie in diesem Schritt sicher, dass die Kabel der vier Servomotoren bereits mit dem Hauptboard verbunden sind, und ziehen Sie erst dann die Schrauben fest.



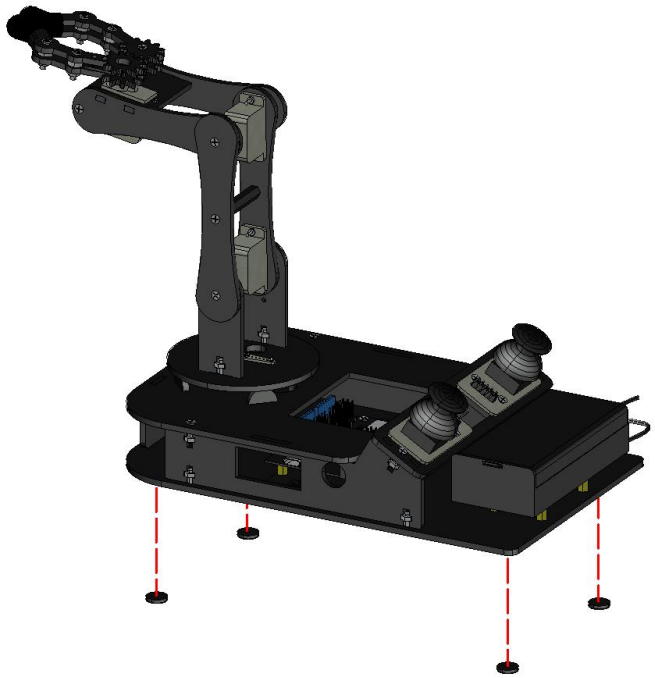
## 18. Batteriefach Befestigen

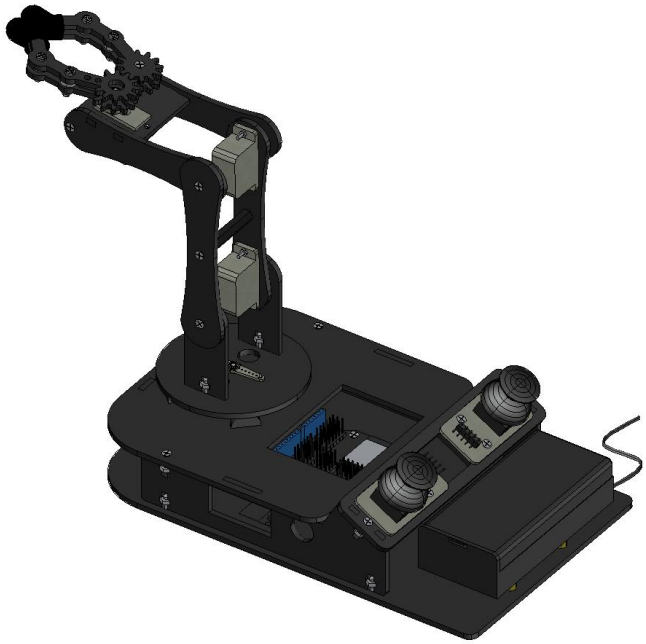
| Teileliste | |
|---|---|
| **Name** | **Menge** |
| 18650 Battery Holder | 1 |
| M3*8 Flat Head Screws | 8 |
| M3*12MM Dual-pass Copper Pillar | 4 |



## 19. Installieren Sie die Gummipads

Ziehen Sie die Schutzfolie der Gummipads ab und kleben Sie die Gummipads an die vier Ecken an der Unterseite der Roboterarm-Basis.





Der Roboterarm ist nun montiert!

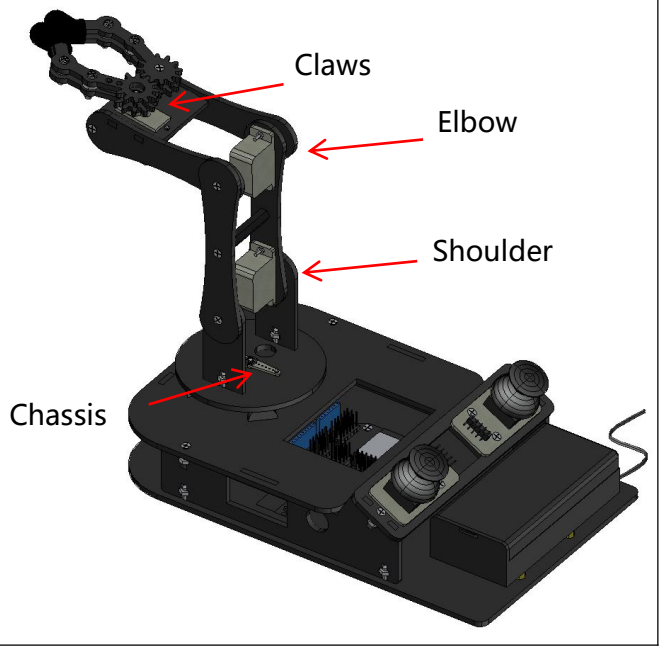
# Lektion 3 Joystick-Steuerung für Roboterarm

## I. Servomotorsteuerung des Roboterarms

### 1. Erklärung der Servomotor-Pins des Roboterarms

Nach dem Einbau der Servomotoren des Roboterarms übernimmt jeder Servomotor eine andere Funktion des Roboterarms. Wenn wir den Winkel jedes Servomotors per Programm steuern und unterschiedliche Funktionen realisieren möchten, müssen wir wissen, welchem Pin jeder Servomotor zugeordnet ist.

Der Roboterarm verwendet insgesamt 4 Servomotoren. Die entsprechenden Pins sind in der folgenden Tabelle aufgeführt:

| Sequenz nummer | Pin-Nummer | Servomotor position | Diagramm |
|---|---|---|---|
| 1 | GPIO18 | Claws |   |
| 2 | GPIO17 | Elbow | |
| 3 | GPIO16 | Shoulder | |
| 4 | GPIO5 | Chassis | |

### 2. Bewegungsgesetze Der Einzelnen Gelenke Des Roboterarms

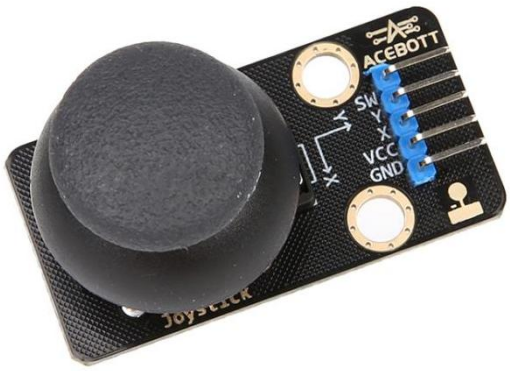
| Sequenz nummer | Pin-Nummer | Servomotor position | Bewegungsregel |
|---|---|---|---|
| 1 | GPIO18 | Claws | Je größer der Winkel des Servomotors, desto weiter öffnen sich die Greifer. |
| 2 | GPIO17 | Elbow | Je größer der Winkel des Servomotors, desto weiter hebt sich der Ellbogen des Roboterarms nach oben. |

| | | | |
|---|---|---|---|
| 3 | GPIO16 | Shoulder | Je größer der Winkel des Servomotors, desto weiter senkt sich die Schulter des Roboterarms nach unten. |
| 4 | GPIO5 | Chassis | Je größer der Winkel des Servomotors, desto weiter dreht sich das Chassis nach links. |

## II. Joystick-Modul kennenlernen

### 1. Einführung in das Joystick-Modul

Das Joystick-Module besteht aus zwei Schiebepotentiometern und einer Taste. Wenn der Joystick bewegt wird, verändert sich der Widerstandswert der Schiebepotentiometer, und die entsprechenden X/Y-Spannungswerte ändern sich ebenfalls entsprechend. Wenn der Joystick nach unten gedrückt wird, wird die Taste ausgelöst, und das entsprechende SW-Signal wird auf einen niedrigen Pegel gesetzt. Es wird häufig in Schiffsmodellen, Spielkonsolen, ferngesteuerten Fahrzeugen, Gimbals und anderen Geräten eingesetzt.




### 2. Test der seriellen Portwerte des Joystick-Moduls

Um den entsprechenden X/Y-Wert der Joystick-Bewegung zu ermitteln, kann dieser in Echtzeit im seriellen Monitor von Arduino IDE ausgegeben werden.

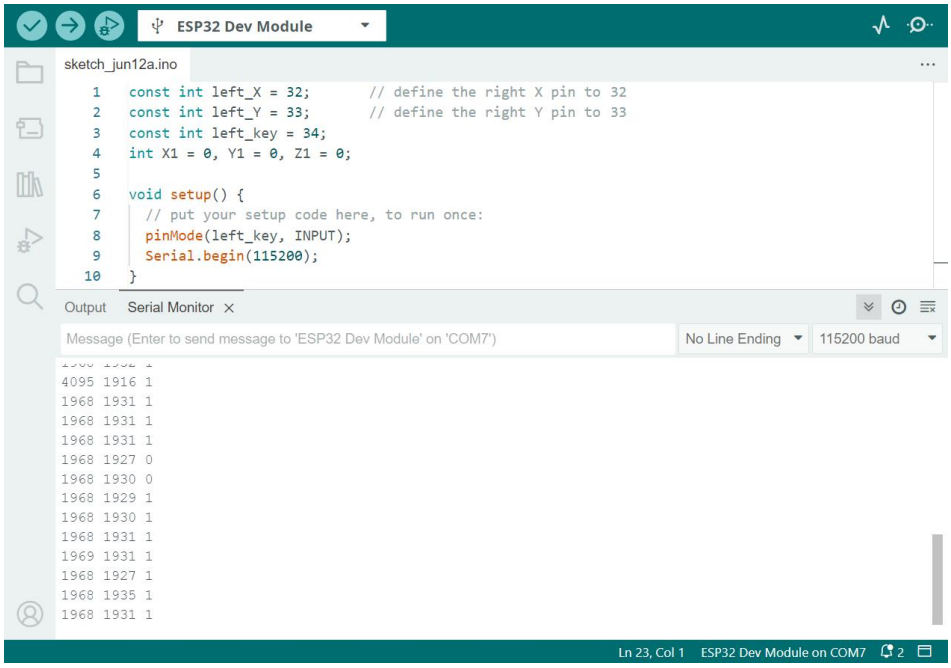
Öffnen Sie „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 3\Joystick_test“ und darin „Joystick_test.ino“. Verbinden Sie das ESP32-Entwicklungsboard über ein USB-Kabel mit dem Computer, wählen Sie das richtige Entwicklungsboard und den

richtigen Port aus und laden Sie den Code auf das ESP32-Entwicklungsboard hoch.

Das Referenzprogramm lautet wie folgt:

```
const int left_X = 32;//Define the right X pin to 32
const int left_Y = 33;//Define the right Y pin to 33
const int left_key = 34;
int X1 = 0, Y1 = 0, Z1 = 0;

void setup() {
  pinMode(left_key, INPUT);
  Serial.begin(115200);
}
void loop() {
  X1 = analogRead(left_X);//Read the right X value
  Serial.print(X1);
  Y1 = analogRead(left_Y);//Read the right Y value
  Serial.print(" ");
  Serial.print(Y1);
  Z1 = digitalRead(left_key);//Read the right Z value
  Serial.print(" ");
  Serial.println(Z1);
  delay(1000);
}
```



## III. Grundlegende Bewegungen des joystickgesteuerten Roboterarms

Die grundlegenden Steuerungsbewegungen des Roboterarms umfassen hauptsächlich Links- und Rechtsbewegungen, Rechtsdrehung, Auf- und Abwärtsbewegung des Ellbogens sowie Öffnen und Schließen der Klaue. Sobald diese Grundbewegungen beherrscht werden, können weitere Bewegungen darauf aufbauend kombiniert und ausgeführt werden.

**Hinweis: Nachdem der Roboterarm eingeschaltet wurde, ist es verboten, den Servomotor direkt mit der Hand zu drehen, um Schäden am Servomotor zu vermeiden.**

### 1. Joystick-Steuerungsprogramm

Öffnen Sie „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 3\JoyStick_Controlled_Robot_Arm“ und darin „JoyStick_Controlled_Robot_Arm.ino“. Verbinden Sie das ESP32-Entwicklungsboard über ein USB-Kabel mit dem Computer, wählen Sie das richtige Entwicklungsboard und den richtigen Port aus und laden Sie den Code auf das ESP32-Entwicklungsboard hoch. Das Entwicklungsboard muss an die Stromversorgung des Batteriefachs angeschlossen werden und der Schalter des Batteriefachs ist auf „ON“ zu stellen.

Das Referenzprogramm lautet wie folgt:

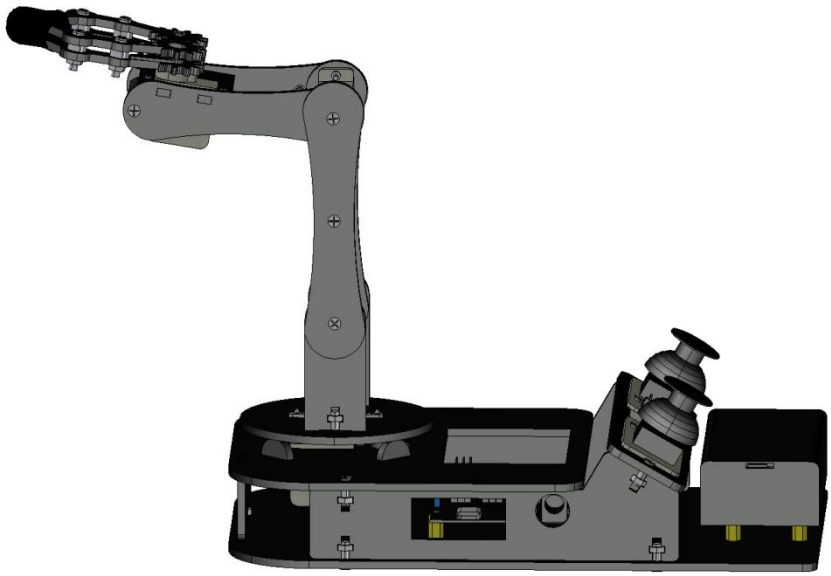
```
#include <ACB_ARM.h>//Add the Robot arm libraries
ACB_ARM ARM;

void setup() {
  ARM.ARM_init(5,16,17,18);//The parameters are four Servo pins
  ARM.JoyStick_init(32,33,34,35,36,39);//Joystick initialization
}
void loop() {
```

```
  ARM.get_JoyStick();
  if (ARM.JoyY1 < 50) {//chassis left
    ARM.chassis_angle = ARM.chassis_angle + 1;
    ARM.JoyChassisCmd(ARM.chassis_angle);
  }
  if (ARM.JoyY1 > 3500) {//chassis right
    ARM.chassis_angle = ARM.chassis_angle - 1;
    ARM.JoyChassisCmd(ARM.chassis_angle);
  }
  if (ARM.JoyX1 < 50) {//Shoulder down
    ARM.shoulder_angle = ARM.shoulder_angle + 1;
    ARM.JoyShoulderCmd(ARM.shoulder_angle);
  }
  if (ARM.JoyX1 > 4000) {//Shoulder up
    ARM.shoulder_angle = ARM.shoulder_angle - 1;
    ARM.JoyShoulderCmd(ARM.shoulder_angle);
  }
  if (ARM.JoyX2 < 50) {//Elbow up
    ARM.elbow_angle = ARM.elbow_angle + 1;
    ARM.JoyElbowCmd(ARM.elbow_angle);
  }
  if (ARM.JoyX2 > 4000) {//Elbow dwon
    ARM.elbow_angle = ARM.elbow_angle - 1;
    ARM.JoyElbowCmd(ARM.elbow_angle);
  }
  if (ARM.JoyY2 > 4000) {// Claws open
    ARM.claws_angle = ARM.claws_angle + 1;
    ARM.JoyClawsCmd(ARM.claws_angle);
  }
  if (ARM.JoyY2 < 50) {// Claws close
    ARM.claws_angle = ARM.claws_angle - 1;
    ARM.JoyClawsCmd(ARM.claws_angle);
  }
}
```

## 2. Joystick-Steuerungsregeln

Nach dem Hochladen des Programms befinden sich die vier Arme des Roboterarms in Form einer „7“; dies ist die Ausgangsposition. Anschließend kann der Roboterarm mithilfe der Joysticks gesteuert werden.



Im Folgenden finden Sie eine Einführung in die Joystick-Funktionen:

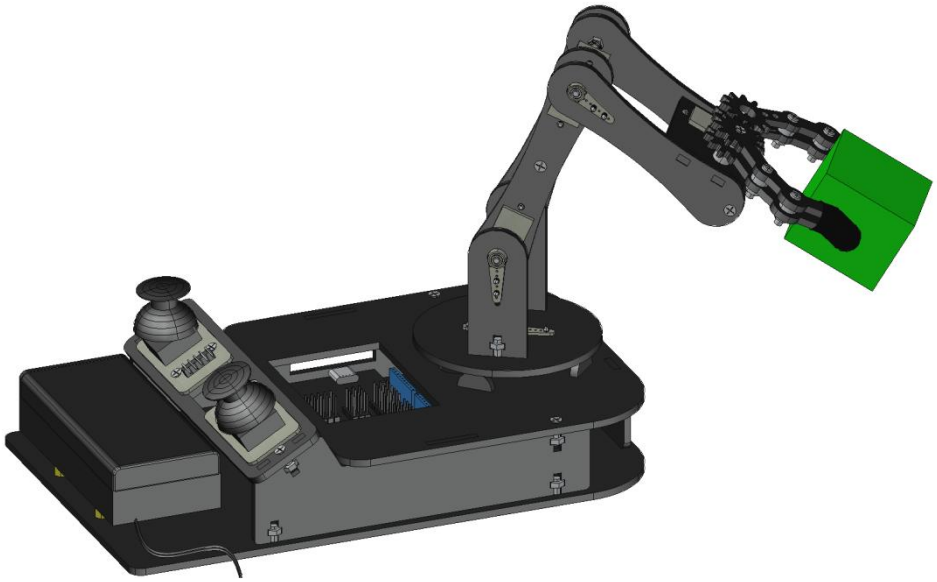
| Joystick | Richtung | Gelenkposition | Bewegungsgesetze |
|---|---|---|---|
| Linker Joystick | Links | Chassis | Nach links drehen |
| | Rechts | Chassis | Rechts drehen |
| | Hoch | Shoulder | Nach oben bewegen |
| | Runter | Shoulder | Nach unten bewegen |
| Rechter Joystick | Hoch | Elbow | Nach oben bewegen |
| | Runter | Elbow | Nach unten bewegen |
| | Links | Claws | Greifer öffnen |
| | Rechts | Claws | Greifer schließen |

## IV. Erweiterte Aufgaben

Basierend auf den grundlegenden Bedienregeln des Joysticks für den Roboterarm können wir nun die Funktion implementieren, den Roboterarm mithilfe des Joysticks zum Tragen von Objekten zu steuern.

**Aufgabenbeschreibung:**

Verwenden Sie das Joystick-Module, um den Roboterarm zu steuern und den Block von Punkt A aufzunehmen und zu Punkt B zu bewegen. (Hinweis: Die Positionen von Punkt A und B können Sie selbst festlegen.)

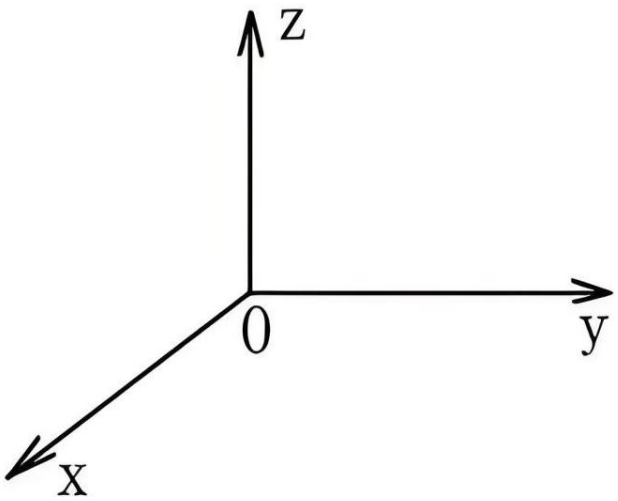


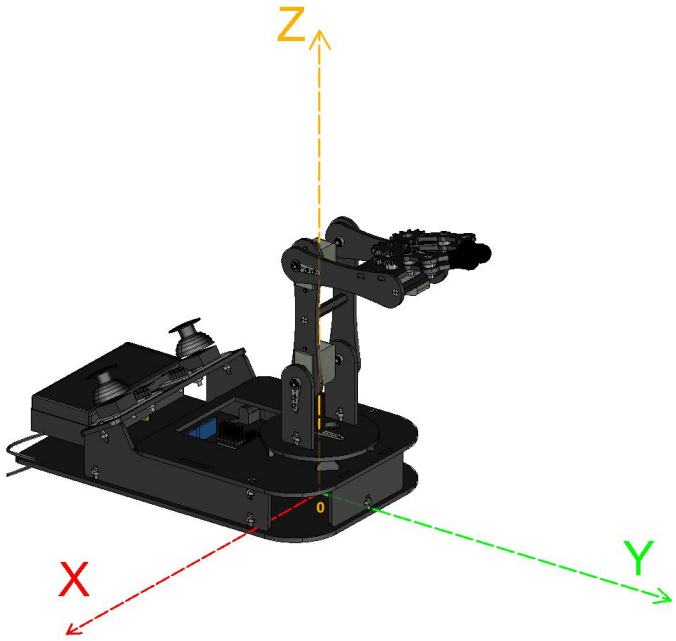
# Lektion 4 Räumliche Koordinaten des Roboterarms

Die räumlichen Koordinaten des Roboterarms spielen eine entscheidende Rolle für seine Steuerung und Programmierung. Präzise räumliche Koordinaten ermöglichen eine genaue Positionierung, optimierte Bewegungsplanung, effektive Hindernisvermeidung und präzise Bedienung des Roboterarms und verbessern gleichzeitig den Automatisierungs- und Intelligenzgrad.

## I. Kartesisches Koordinatensystem

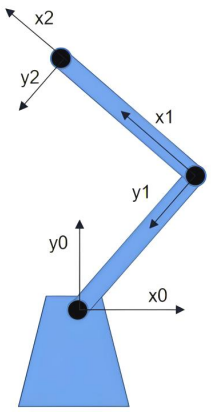
In der räumlichen Bewegungssteuerung von Roboterarmen ist das kartesische Koordinatensystem eines der am häufigsten verwendeten Koordinatensysteme. Es ist ein mathematisches System, das die Position eines Punktes im Raum beschreibt. Im dreidimensionalen Raum besteht das kartesische Koordinatensystem aus drei zueinander senkrechten Koordinatenachsen (x, y, z). Der Schnittpunkt der drei Koordinatenachsen ist der Ursprung (O) des Koordinatensystems. In diesem Tutorial befindet sich der Ursprung des kartesischen Koordinatensystems in der Mitte der Chassis-Servoscheibe des Roboterarms.






## II. Gelenkkoordinatensystem

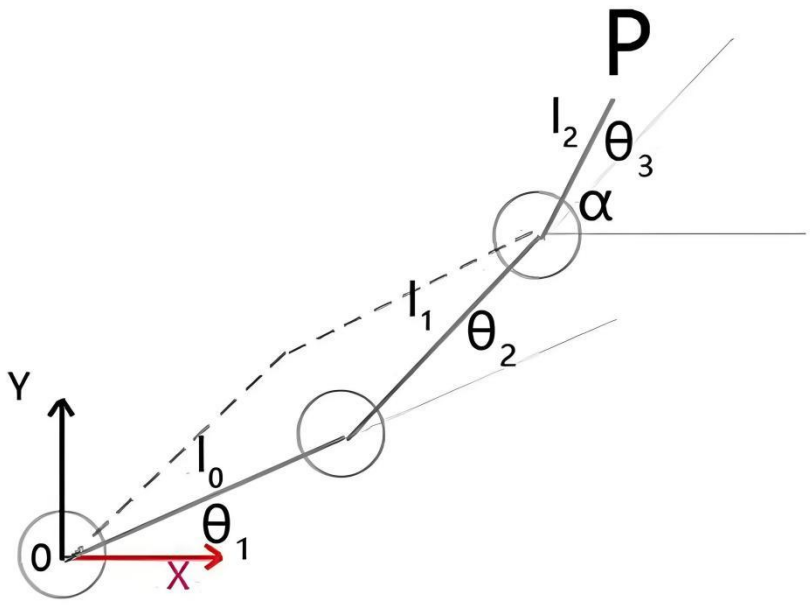
Zusätzlich zum kartesischen Koordinatensystem verfügt jedes Gelenk des Roboterarms über ein eigenes Koordinatensystem, das sogenannte Gelenkkoordinatensystem. Der Ursprung liegt üblicherweise am Gelenk, und die Koordinatenachsen verlaufen entlang der Rotationsachse des Gelenks. Jedem Gelenkkoordinatensystem ist ein eigenes Gelenkkoordinatensystem zugeordnet, das den Rotationswinkel oder die Streckung dieses Gelenks beschreibt. Da sich jedes Gelenk des Roboterarms drehen oder strecken kann, lässt sich das Gelenkkoordinatensystem entsprechend der aktuellen Position des Roboterarms transformieren.



## III. Vorwärts- und inverse Kinematik

Die Vorwärtskinematik berechnet die Position und Haltung des Roboterarms im kartesischen Koordinatensystem anhand der Gelenkwinkel. Die Inverse Kinematik hingegen verwendet kartesische Koordinaten, um die Winkel jedes Gelenks zu berechnen und so die Bewegung des Roboterarms zu realisieren.

Probleme der Vorwärtskinematik lassen sich durch Matrixtransformation lösen, d. h. durch Vergleich der Transformationsmatrix jedes Gelenkkoordinatensystems mit der Koordinatentransformationsformel des Roboterarms, um die Winkel jedes Gelenks zu bestimmen. Probleme der Inversen Kinematik sind in der Regel komplexer und erfordern die Lösung eines Systems nichtlinearer Gleichungen mit mehreren Lösungen.

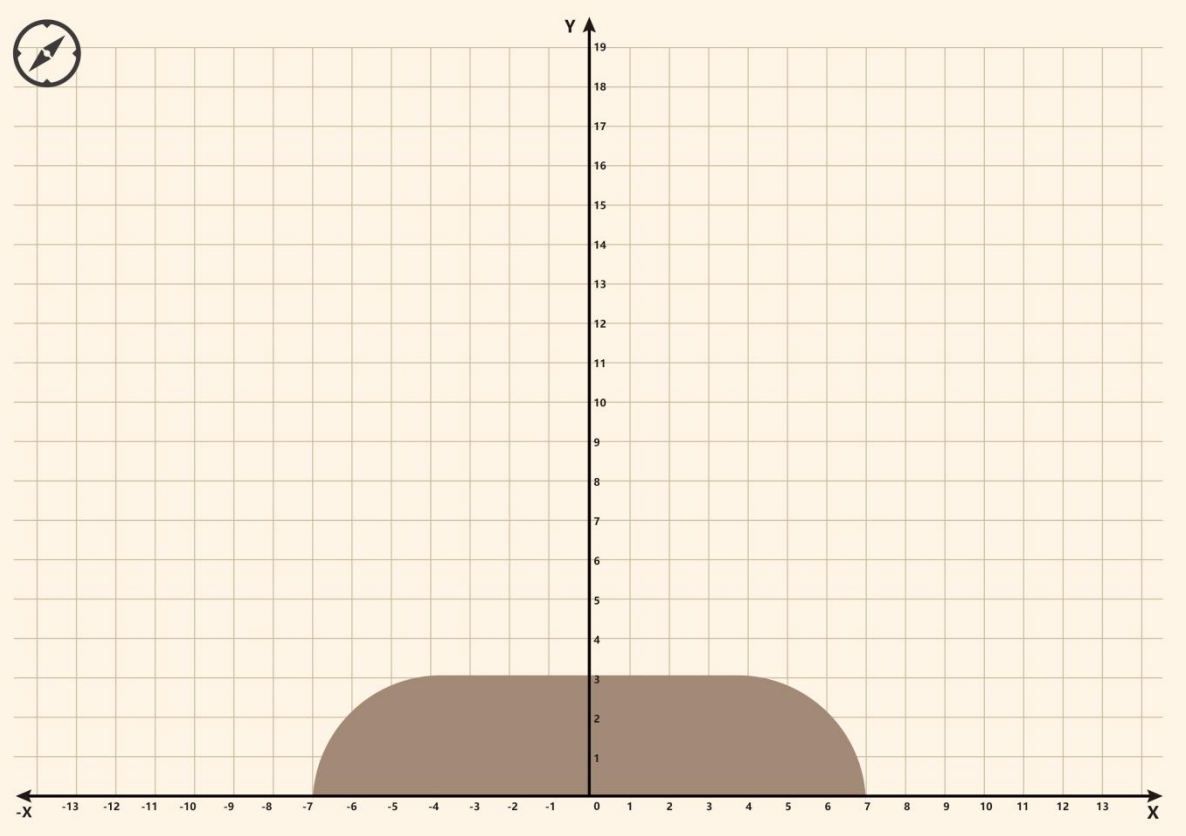



In diesem Tutorial verwenden wir hauptsächlich das Zentrum der Chassis-Servomotor-Scheibe als Ursprung, um ein kartesisches Koordinatensystem aufzubauen. Wenn die angegebenen Positionskoordinaten (X, Y, Z) eingegeben werden, verwendet die Bibliotheksdatei im Tutorial automatisch den Algorithmus der inversen Kinematik, um die Koordinaten der einzelnen Gelenke des Roboterarms

zu berechnen. Anschließend werden diese in die Winkel der Servomotoren der einzelnen Gelenke umgerechnet, sodass das Ende des Roboterarms den Zielpunkt im Raumkoordinatensystem erreichen kann.

## IV. Koordinatendiagramm des Roboterarms

Beim Zusammenbau des Roboterarms treten zwangsläufig Fehler auf. Um den Roboterarm besser zu kalibrieren, benötigen wir eine Koordinatenkarte. Die Koordinatenkarte des Roboterarms besteht aus X- und Y-Koordinaten. Der Schnittpunkt von X und Y ist der Koordinatenpunkt der Karte, wobei der X-Koordinatenbereich [-13, 13] und der Y-Koordinatenbereich [0, 19] ist. Das schattierte, abgeschrägte Rechteck ist die Referenzposition des Roboterarms. Platzieren Sie die Oberkante des Roboterarm-Chassis an dieser Form.



**[Klicken Sie auf die Koordinatenkarte des Roboterarms PDF-Datei]**

Hinweis: Bitte drucken Sie die Karte des Roboterarms selbst anhand der PDF-Datei aus. Sie können sie auf A4-Papier ausdrucken.

## V. Kalibrierungsanweisungen für den Roboterarm

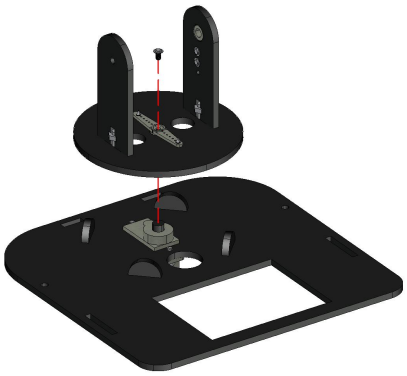
Der Roboterarm ist eine Maschine, die aus mehreren Servomotoren besteht. Im tatsächlichen Betrieb treten gewisse Abweichungen auf. Um diese Abweichungen zu verringern, haben wir dem Steuerprogramm des Roboterarms Befehle hinzugefügt, mit denen sich die Fehler des Roboterarms anpassen lassen.

```
ARM.Chassis_angle_adjust(8);  // Default 0
ARM.Slight_adjust(0,2);   // Default 0,0
```

Zunächst dient der Befehl „**ARM.Chassis_angle_adjust(8)**“ dazu, die Zentrumswinkelabweichung des Chassis-Servomotors anzupassen. Der Standard-Zentrumswinkel beträgt 90 Grad, und der Standardabweichungswert des Mittelpunkts ist 0. Die Winkel auf der linken Seite (-X-Achse) liegen zwischen 90 und 180 Grad, während die Winkel auf der rechten Seite (X-Achse) zwischen 0 und 90 Grad liegen.

Aufgrund von Problemen mit der Zahnradgenauigkeit bei der Installation stimmt die Montageposition des Chassis-Servomotors und der Servohorns jedoch nicht unbedingt genau mit 90 Grad überein. Es kann leicht nach links oder rechts abweichen. Daher müssen wir es per Programm kalibrieren. Angenommen, es weicht um 8 Grad nach rechts ab, dann muss der Abweichungswert erhöht werden, weshalb im Klammern des Befehls „**ARM.Chassis_angle_adjust()**“ eine 8 eingetragen wird. Wenn es sich um eine Verschiebung um 8 Grad nach links handelt, muss der Abweichungswert subtrahiert werden, und im

Klammern des Befehls „**ARM.Chassis_angle_adjust()**“ wird eine -8 eingetragen.



Dann folgt der Befehl „**ARM.Slight_adjust(0,2)**“. Dieser Befehl hat zwei Parameter, deren Standardwerte beide 0 sind. Angenommen, wenn das Ende des Roboterarms den angegebenen Raumpunkt erreicht, treten immer noch geringfügige Verschiebungsfehler auf, dann kann dieser Befehl zur Feinjustierung verwendet werden.

Der erste Parameter ist, dass der Roboterarm direkt 1 schreibt, wenn das Ende noch 1 Grad nach rechts abweicht, wenn es 1 Grad nach links abweicht, -1 schreibt, wenn es nicht abweicht, 0 schreibt.

Der zweite Parameter gibt an, wann sich der Roboterarm auf der negativen X-Achse befindet. Wenn der Endeffektor um 1 Grad nach rechts ausgelenkt wird, schreiben Sie 1; wenn er um 1 Grad nach links ausgelenkt wird, schreiben Sie -1; wenn er nicht ausgelenkt wird, schreiben Sie 0.

Passen Sie anschließend mithilfe der Koordinatenkarte die Kalibrierungsparameter entsprechend den unten stehenden räumlichen Koordinaten und Ihrer tatsächlichen Situation an.

## VI. Punktbewegungen in räumlichen Koordinaten

Öffnen Sie „Inverse_Kinematics.ino“ in „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion

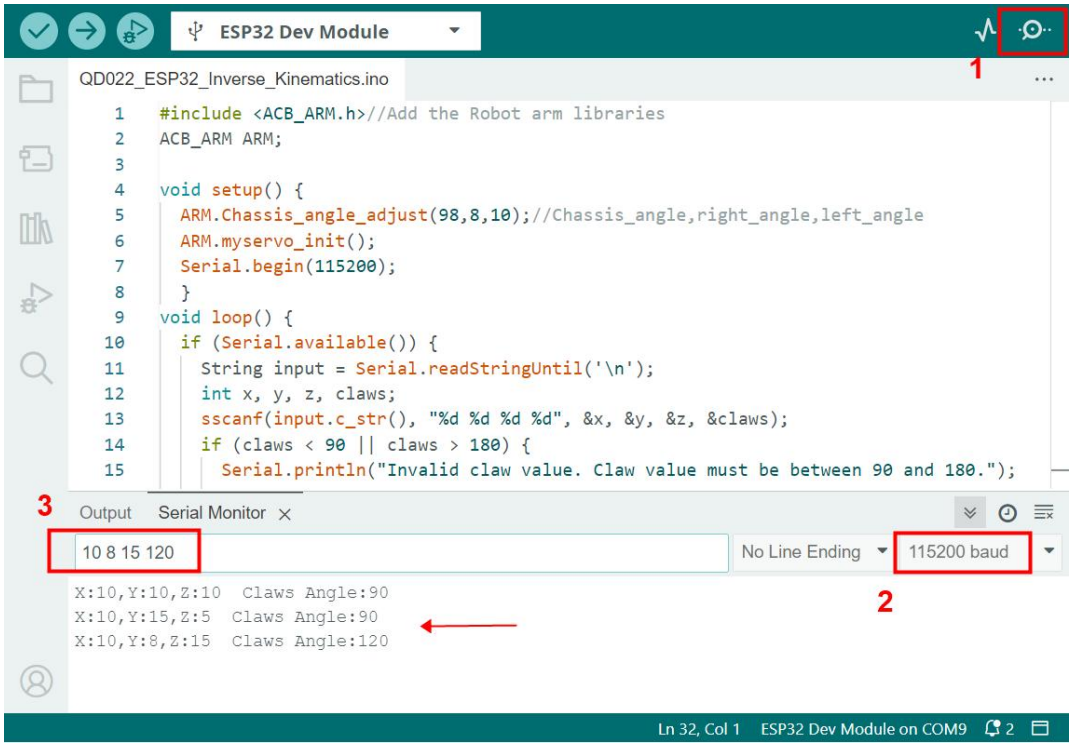
4\Inverse_Kinematics“, verbinden Sie die ESP32-Entwicklungsplatine mit dem Computer über ein USB-Kabel, wählen Sie den richtigen Entwicklungsplatinen-Typ und die richtige Portnummer aus, laden Sie den Code auf die ESP32-Entwicklungsplatine hoch, versorgen Sie die Entwicklungsplatine mit Strom vom Batteriefach und schalten Sie den Schalter am Batteriefach auf die Position „ON“.

Das Referenzprogramm lautet wie folgt:

```
#include <ACB_ARM.h>//Add the Robot arm libraries
ACB_ARM ARM;

void setup() {
  ARM.Chassis_angle_adjust(8);//Default 0
  ARM.Slight_adjust(0,2);//Default(0,0)
  ARM.ARM_init(5,16,17,18);//The parameters are four Servo pins
  Serial.begin(115200);
}
void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    int x, y, z, claws;//Define variables for coordinates x, y, z, and claws
    sscanf(input.c_str(), "%d %d %d %d", &x, &y, &z, &claws);
    if (claws < 90 || claws > 180) {
      Serial.println("Invalid claw value. Claw value must be between 90 and 180.");
      return;
    }
    Serial.print("X:");
    Serial.print(x);
    Serial.print(",Y:");
    Serial.print(y);
    Serial.print(",Z:");
    Serial.print(z);
    Serial.print("    Claws Angle:");
    Serial.print(claws);
    Serial.println(" ");
```

```
    ARM.ClawsCmd(claws);
    delay(1000);
    ARM.PtpCmd(x, y, z);
  }
}
```




Nach dem Hochladen des Programms geben Sie vier Werte in der seriellen Schnittstelle ein: die Koordinate X, die Koordinate Y, die Koordinate Z sowie der Öffnungswinkel der Klaue (Bereich 90 bis 180 Grad). Diese vier Werte müssen durch Leerzeichen getrennt werden; drücken Sie nach der Eingabe Enter.

„**ARM.PtpCmd(x, y, z)**“ ist der Koordinatensteuerbefehl für den Roboterarm, mit dem die Bewegung des Endeffektors des Arms zu einer angegebenen Position im Raum gesteuert wird, wobei x die x-Achsenkoordinate, y die y-Achsenkoordinate und z die z-Achsenkoordinate ist.

„**ARM.ClawsCmd(claws)**“ ist der Steuerbefehl für die Klaue des Roboterarms, mit dem das Öffnen und Schließen der Endklaue gesteuert wird, wobei „claws“ den Winkelparameter des Klauen-Servos darstellt, der im Bereich von 90 bis 180 eingegeben wird.

Wenn die Koordinaten korrekt sind, bewegt sich der Roboterarm zum Raumpunkt. Wenn nach der Eingabe der Koordinaten in der seriellen Schnittstelle die Meldung „Out of range!“ erscheint, liegt dies daran, dass der Bewegungsbereich des Arms innerhalb einer Kugel liegt, sodass eingegebene Werte außerhalb des erreichbaren Bereichs des Arms liegen können. In diesem Fall passen Sie die Daten an und geben Sie sie erneut ein.

# Lektion 5 Palettierung mit dem Roboterarm

In der heutigen, sich rasant entwickelnden Technologie sind Roboterarme zu einem unverzichtbaren Bestandteil moderner Industrie, Dienstleistungen und des Alltags geworden. Sie haben traditionelle Arbeitsweisen mit ihrer erstaunlichen Flexibilität, Genauigkeit und Effizienz grundlegend verändert.

Insbesondere die Roboterarm-Palettierungstechnologie findet in verschiedenen Bereichen breite Anwendung, vor allem dort, wo ein effizientes Handling großer Mengen an Gütern erforderlich ist und ihr Nutzen daher besonders groß ist. Beispielsweise kann die Roboterarm-Palettierungstechnologie in der Logistiklagerhaltung die Effizienz und Genauigkeit der Warenhandhabung deutlich verbessern und die Kosten für manuelle Tätigkeiten sowie menschliche Fehler reduzieren. Im Vergleich zur manuellen Palettierung bietet die Roboterarm-Palettierungstechnologie Vorteile wie hohe Effizienz, gute Stabilität und einfache Bedienung.



Mit dem technologischen Fortschritt und der Marktnachfrage wird die Roboterarm-Palettierungstechnologie in Zukunft noch breitere Anwendung finden. Daher lernen wir in dieser Lektion, wie man grundlegende Palettierungsfunktionen mithilfe eines Roboterarms implementiert.

**I. Palettierung mit dem Roboterarm**

Öffnen Sie „Robot_Arm_Stacking.ino“ in „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 5\Robot_Arm_Stacking“, verbinden Sie die ESP32-Entwicklungsboard mit dem Computer über ein USB-Kabel, wählen Sie die richtige Entwicklungsplatine und den richtigen Port aus, laden Sie den Code auf die ESP32-Entwicklungsboard hoch, versorgen Sie die Entwicklungsplatine mit Strom vom Batteriefach und schalten Sie den Schalter am Batteriefach auf die Position „ON“.

Nachdem zwei Blöcke im Voraus vertikal gestapelt wurden, werden sie an den Koordinaten (-7, 13) auf der Roboterarmkarte platziert. Dabei ist darauf zu achten, dass der Mittelpunkt der Blöcke mit dem Koordinatenpunkt übereinstimmt.



Das Referenzprogramm lautet wie folgt:

```
#include <ACB_ARM.h>//Add the Robot arm libraries
ACB_ARM ARM;

bool RunningState = true;

int startx = 7, starty = 13, startz = 2;//Initial coordinates
int midz = 15;//Height of middle point
int endx = -7, endy = 13, endz = 0;//End point coordinates
int openAngle = 130, closeAngle = 90;
int count = 2;//Number of blocks
int i = 0;

void setup() {
  ARM.Chassis_angle_adjust(8);//Chassis error calibration
  ARM.Slight_adjust(0,2);//Small calibration error
  ARM.ARM_init(5,16,17,18);//The parameters are four Servo pins
  Serial.begin(115200);
}

void loop() {
  while(i < count) {
    ARM.ClawsCmd(openAngle);//open claws
    delay(1000);
    ARM.PtpCmd(startx, starty, startz-i*2);//The target point is the upper object
    delay(1000);
    ARM.ClawsCmd(closeAngle);//close claws
    delay(1000);
    ARM.PtpCmd(startx, starty, midz);//Lift the object after picking it up
    delay(1000);
    ARM.PtpCmd(endx, endy, midz);//Rotate above the target point
    delay(1000);
    ARM.PtpCmd(endx, endy, endz+i*2);//Placing Objects
    delay(1000);
    ARM.ClawsCmd(openAngle);
    delay(1000);
```

```
      ARM.PtpCmd(endx, endy, midz);//Raise arms
      delay(1000);
      i = i + 1;
    }
    ARM.Zero();//Servo initialization
}
```

Nach dem Hochladen des Programms ist zu sehen, dass das Ende des Roboterarms die Ausgangskoordinaten der Blöcke erreicht, den oberen Block greift, ihn zu den Endpunktkoordinaten bewegt, ihn absetzt, dann zu den Ausgangskoordinaten zurückkehrt, den unteren Block greift, ihn zu den Endpunktkoordinaten bewegt und ihn schließlich auf den ersten Block stapelt.

## II. Erweiterungsaufgaben

Anhand der Koordinatenkarte des Roboterarms ist das Stapelmuster bereits bekannt. Sobald die Positionskoordinaten auf der Karte bekannt sind, kann der Roboterarm die Stapelfunktion ausführen. Im nächsten Schritt wird die Karte kombiniert, um die Stapelfunktion des Roboterarms zu vervollständigen.

**Aufgabenbeschreibung:**

Versuchen Sie, basierend auf dem Stapelprogramm im Beispiel, die Koordinaten der Anfangs- und Endposition des Objekts im Programm so zu ändern, dass der Roboterarm an verschiedenen Positionen Stapeleffekte erzielt.

(Hinweis: Die beiden Blöcke müssen nebeneinander platziert werden, und die Koordinateneingabe darf den Kartenbereich nicht überschreiten. Außerdem können aufgrund von Platzierungs- oder Genauigkeitsproblemen Fehler beim Greifen des Roboterarms auftreten.)

# Lektion 6 Teach-Funktion des Roboterarms

Die Programmierung durch Teach-in beim Roboterarm ist eine Methode, bei der der Bediener dem Roboterarm feste Bewegungsbahnen vorgibt, sodass der Roboterarm die Aufgaben gemäß den voreingestellten Schritten ausführt.

Die Teach-in-Programmierung des Roboterarms lässt sich in drei Schritte unterteilen: Der erste Schritt ist das Einlernen der Bewegungen, bei dem der Bediener dem Roboterarm feste Bewegungsbahnen vorgibt; der zweite Schritt ist das Speichern der Bewegungen, bei dem das Steuerungssystem des Roboterarms die eingelernten Bewegungen aufzeichnet; der dritte Schritt ist die Wiedergabe der Teach-in-Bewegungen, bei der der Roboterarm die zuvor aufgezeichneten Bewegungen erneut ausführt.

In dieser Lektion werden wir das Joystick-Module verwenden, um den Roboterarm im Teach-in-Verfahren zu erlernen.

## I. Teach-Programm

Öffnen Sie „Memory_Controlled_Robot_Arm.ino“ in „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 6\Memory_Controlled_Robot_Arm“, verbinden Sie die ESP32-Entwicklungsplatine mit dem Computer über ein USB-Kabel, wählen Sie die richtige Entwicklungsplatine und den richtigen Port aus, laden Sie den Code auf die ESP32-Entwicklungsplatine hoch, versorgen Sie die Entwicklungsplatine mit Strom vom Batteriefach und schalten Sie den Schalter am Batteriefach auf die Position „ON“.

Das Referenzprogramm lautet wie folgt:

```
#include <ACB_ARM.h>//Add the Robot arm libraries
ACB_ARM ARM;

void setup() {

  ARM.ARM_init(5,16,17,18);//The parameters are four Servo pins
  ARM.JoyStick_init(32,33,34,35,36,39);//Joystick initialization
  Serial.begin(115200);//set the baud rate to 115200

}

void loop() {

  ARM.JoyStick_Memory();//JoyStick memory mode

}
```

Hinweis:

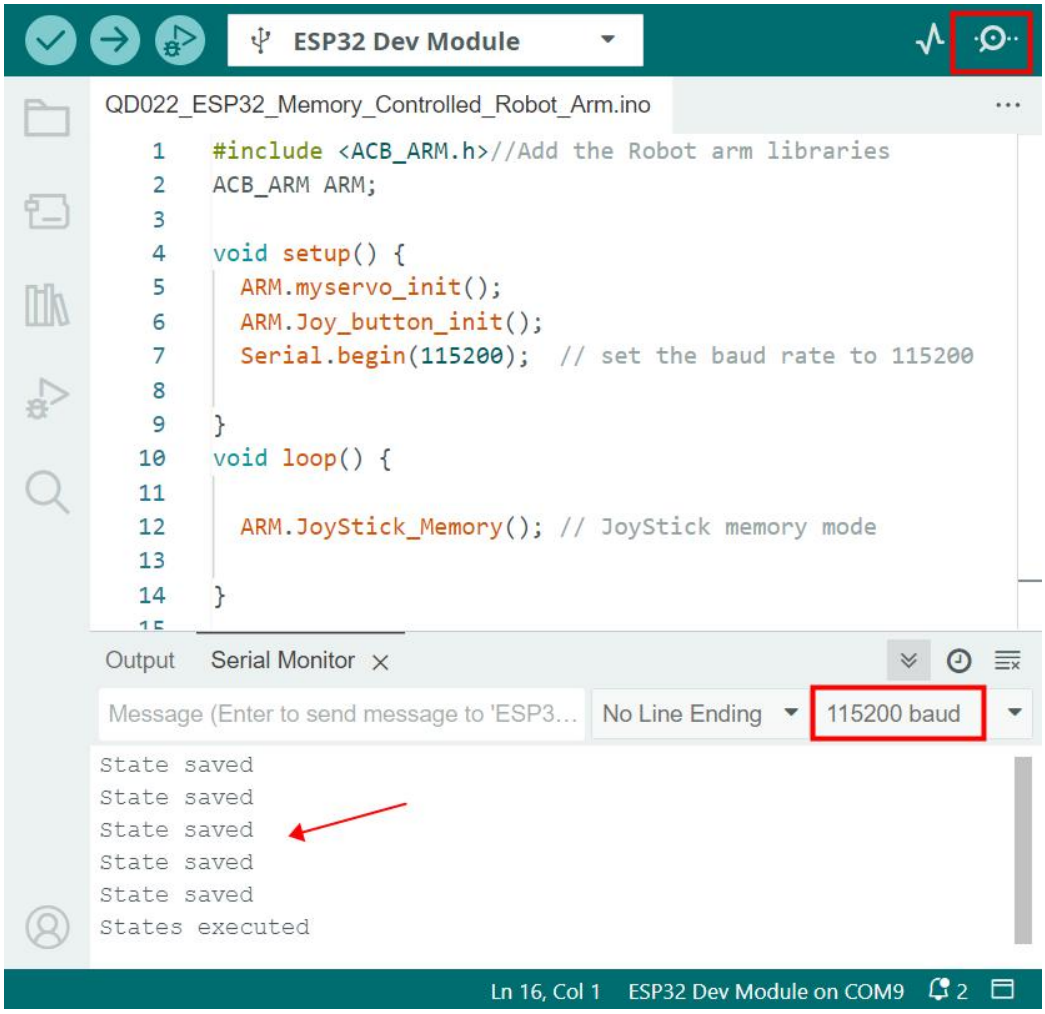
① Die Start- und Endposition des Roboterarms müssen umgehend gespeichert werden.

② Die Speicherfunktion kann maximal 20 Aktionsgruppen gleichzeitig speichern.

**Anweisungen zur Trainingssteuerung:**

| Joystick | Joystick-Bedienung | Aktion |
|---|---|---|
| Linker Joystick | Kurz drücken | Aktion speichern |
| | Lang drücken | Aktion löschen |
| Rechter Joystick | Kurz drücken | Aktion ausführen |

Hinweis: Im seriellen Portmonitor werden die Eingabeaufforderungen für die Aktionsausführung des Roboterarms angezeigt.

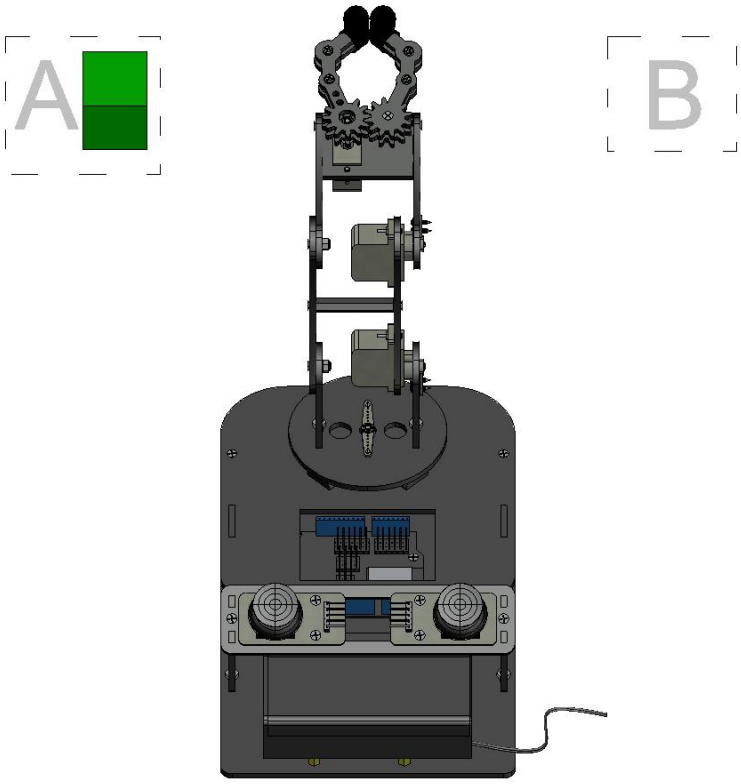



## II. Erweiterungsaufgaben

Basierend auf den grundlegenden Betriebsregeln für das Lehren und Lernen von Roboterarmen können wir nun die Lehr- und Lernfunktion implementieren, indem wir ein Joystick-Module zur Steuerung des Roboterarms und zum Bewegen von Objekten verwenden.

**Aufgabenbeschreibung:**

Der Lernprozess zur Steuerung des Roboterarms mithilfe des Joystick-Moduls: Ein Block wird von Punkt A im Koordinatendiagramm aufgenommen, zu Punkt B bewegt, dort abgesetzt und anschließend zur Ausgangsposition zurückgeführt. (Hinweis: Die Positionen von Punkt A und B können frei gewählt werden.)


A
B

# Lektion 7 Websteuerung des Roboterarms

Mit der kontinuierlichen Entwicklung der drahtlosen Kommunikationstechnologie und der IoT-Technologie wird die Fernsteuerungstechnologie in vielen Bereichen weit verbreitet und ermöglicht es den Benutzern, Endgeräte aus langer Entfernung präzise zu steuern. Es gibt viele Arten von drahtlosen Kommunikationstechnologien, und dieses Tutorial beschreibt hauptsächlich, wie Sie die Fernsteuerung des Roboterarms mit der WiFi-Kommunikationstechnologie erreichen können.

Die WiFi-Kommunikationstechnologie ist eine drahtlose lokale Netzwerktechnologie (WLAN), die es elektronischen Geräten wie Smartphones, Tablets und Laptops ermöglicht, sich drahtlos mit dem Internet oder einem lokalen Netzwerk zu verbinden. Die WiFi-Kommunikationstechnologie verbindet Geräte über einen drahtlosen Router oder einen Zugangspunkt (AP) mit demselben Netzwerk, sodass die Geräte untereinander Daten senden und empfangen können.

Webbasierte Steuergeräte sind eine wichtige Anwendung der WLAN-Kommunikationstechnologie und finden breite Anwendung in Smart Homes und der intelligenten Industrie. Diese Geräte verbinden sich über das Internet mit einem Steuerterminal. Die Interaktion zwischen Gerät und Controller erfolgt über das einfache HTTP-Protokoll. Sobald ein Gerät mit dem Controller verbunden ist, stellt dieser eine benutzerfreundliche Weboberfläche bereit, über die das Gerät gesteuert werden kann.

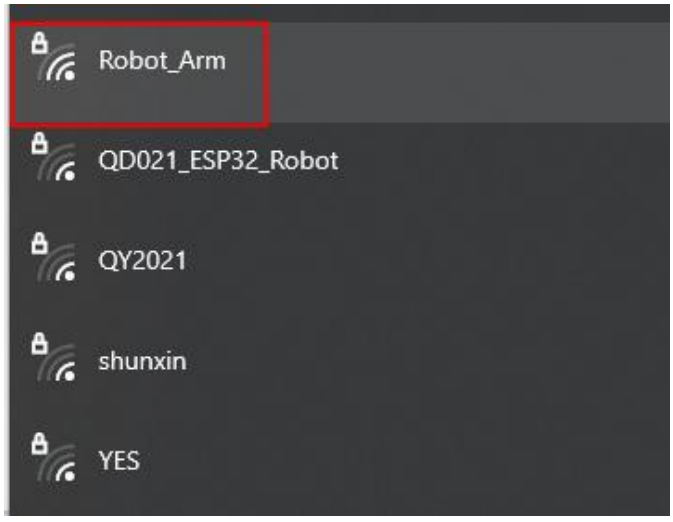
Als Nächstes verwenden wir eine Webseite zur Fernsteuerung des Roboterarms.

## I. Webbasierte Steuerungssoftware

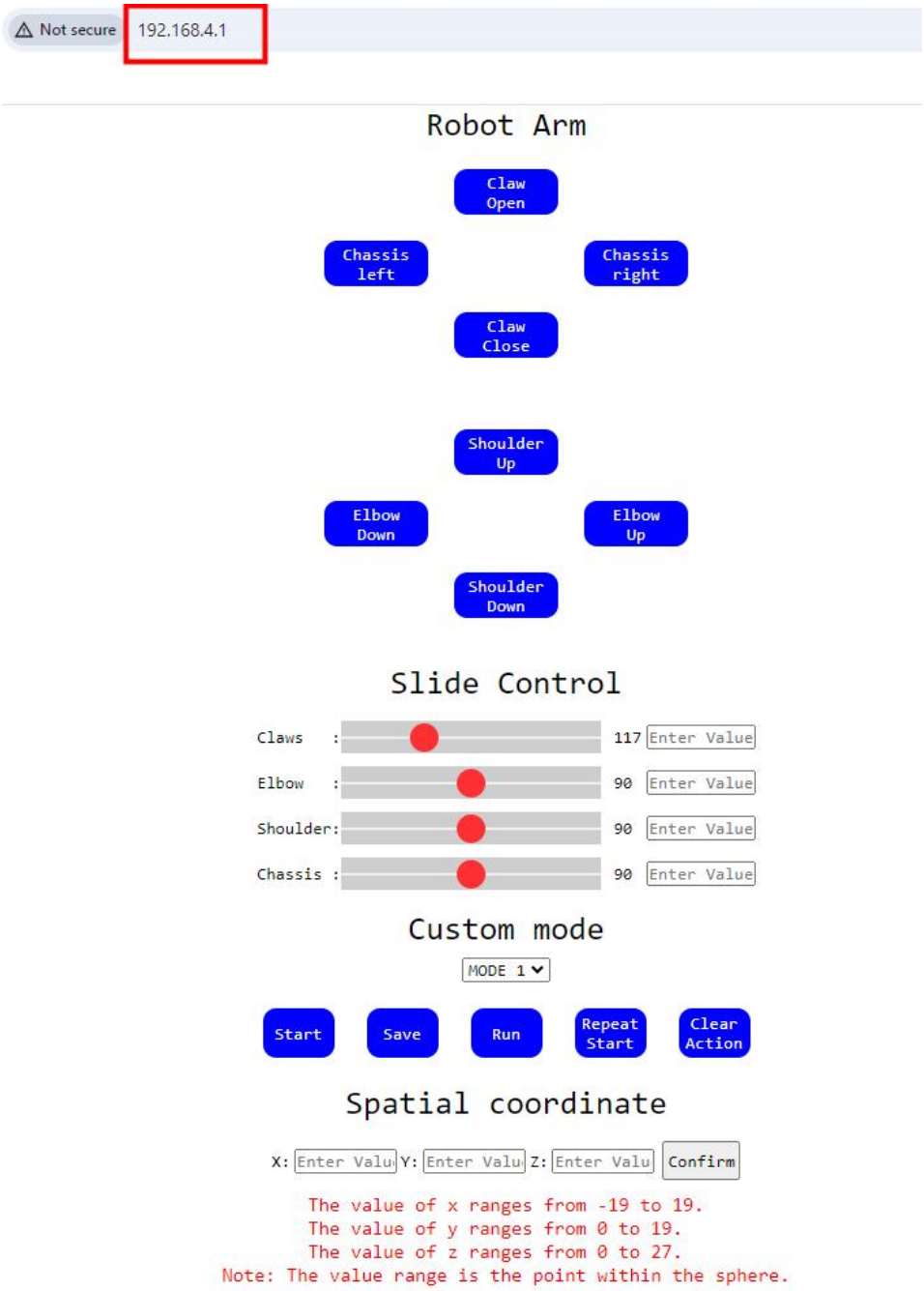
Öffnen Sie die Datei „Web_Controlled_Robot_Arm.ino“ im Verzeichnis „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 7\Web_Controlled_Robot_Arm“, verbinden Sie das ESP32-Entwicklungsboard per USB-Kabel mit dem Computer, wählen Sie das richtige Board und den richtigen Port aus, laden Sie den Code auf das ESP32-Entwicklungsboard hoch, versorgen Sie das Board mit Strom über das Batteriefach und schalten Sie den Schalter des Batteriefachs auf die Position „ON“.

## II. Anmeldung auf der Webseite

Nach erfolgreichem Upload suchen Sie mit Ihrem Computer oder Mobiltelefon nach WLAN-Netzwerken und verbinden Sie sich mit dem WLAN-Hotspot „Robot_Arm“. Das Passwort lautet 12345678, wie in der Abbildung unten gezeigt.




Nach erfolgreicher Verbindung geben Sie „192.168.4.1“ in die Adresszeile Ihres Browsers ein. Die Benutzeroberfläche der Webseite ist in der folgenden Abbildung dargestellt:




| Sequenznummer | Funktionen der Webseite | Funktionsbeschreibung |
|---|---|---|
| 1 | Tastensteuerung | Steuern Sie die Bewegung des Roboterarms über die Tasten auf der Webseite. |
| 2 | Schiebereglersteuerung | Steuern Sie die Bewegung des Roboterarms, indem Sie den Schieberegler bewegen oder einen Winkel im Eingabefeld eingeben. Hinweis: Bewegen Sie den Schieberegler langsam; je schneller Sie schieben, desto schneller bewegt |

| | | |
|---|---|---|
| | | sich der Roboterarm. |
| 3 | Aktion speichern | Es können insgesamt 6 Sätze von Roboterarmaktionen (Modus 1–6) gespeichert werden, wobei jeder Satz 20 verschiedene Aktionen enthalten kann. Die genaue Vorgehensweise ist wie folgt:<br>① Klicken Sie auf „Start“. Die Schaltfläche wechselt dann zu „End“. Führen Sie anschließend die „Save“-Aktionen aus und klicken Sie entsprechend Ihrem Bewegungsablauf schrittweise auf „Save“. Achten Sie darauf, sowohl die Startposition als auch die Endposition mit „Save“ zu speichern;<br>② Klicken Sie auf „End“, um das Speichern der Bewegung abzuschließen;<br>③Klicken Sie auf „Run“, um die gespeicherte Bewegung einmal auszuführen;<br>④ Klicken Sie auf „Repeat Start“, um die gespeicherte Bewegung wiederholt auszuführen;<br>⑤ Klicken Sie auf „Clear Action“, um die Aktionsgruppe zurückzusetzen. |
| 4 | Räumliche Positionierung | Geben Sie die räumlichen Koordinatenwerte x, y und z ein und klicken Sie anschließend auf „Bestätigen“. Der Roboterarm bewegt sich entsprechend den angegebenen räumlichen Koordinatenpunkten.<br>Hinweis: Unterhalb der Eingabefelder für x, y und z befinden sich Beschreibungen der entsprechenden Wertebereiche. Liegt der Wert außerhalb des Bereichs, muss er erneut eingegeben werden. |

## III. Erweiterte Aufgaben

Basierend auf der Webseitensteuerung des Roboterarms verwenden wir nun drei Steuerungsmethoden auf der Webseite, um die

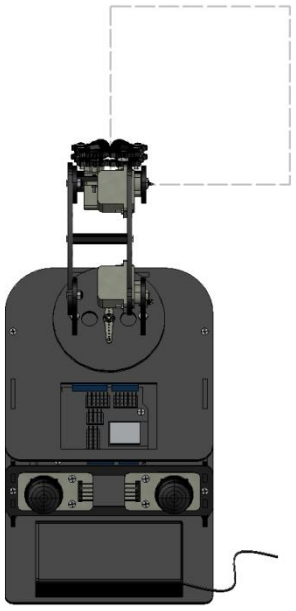
Funktion zum Speichern von Aktionen für den Roboterarm zu implementieren.

**Aufgabenbeschreibung:**

(1) Verwenden Sie die Schaltfläche, um die Greifklaue des Roboterarms zu steuern und eine quadratische Aktion auf der Karte zu zeichnen. Speichern Sie diese im Modus 1.

(2) Verwenden Sie den Schieberegler, um die Greifklaue des Roboterarms zu steuern und eine quadratische Aktion auf der Karte zu zeichnen. Speichern Sie diese im Modus 2.

(3) Verwenden Sie die räumliche Positionierung, um die Greifklaue des Roboterarms zu steuern und eine quadratische Aktion auf der Karte zu zeichnen. Speichern Sie diese im Modus 3.



# Lektion 8 APP-Steuerung des Roboterarms

In den vorherigen Lektionen haben wir bereits gelernt, den Roboterarm mit einem Joystick und über eine Webseite zu steuern. Um den Roboterarm noch bequemer steuern zu können, verwenden wir eine Smartphone-App als Benutzeroberfläche, über die der Roboterarm gesteuert wird. Im Folgenden werden wir lernen, wie man den Roboterarm über eine Smartphone-App steuert.

## I. App herunterladen

(1) Wenn Sie ein iOS-Telefon verwenden, müssen Sie im App Store nach dem Stichwort „ACEBOTT“ suchen und die App herunterladen; wenn Sie ein Android-Telefon verwenden, müssen Sie im Google Play Store nach dem Stichwort „ACEBOTT“ suchen und die App herunterladen; das Symbol ist im Bild unten dargestellt.




Hinweis:

1. Diese Anleitung gilt für die ACEBOTT-App ab Version 2.0. Die Softwareversionsnummer finden Sie in den Einstellungen oben links in der App. Bitte stellen Sie sicher, dass Ihre Version den Anforderungen entspricht.

2. Falls Sie die ACEBOTT-Software aktualisieren müssen, finden Sie in dieser Anleitung eine Anleitung zum Herunterladen der neuesten Version.

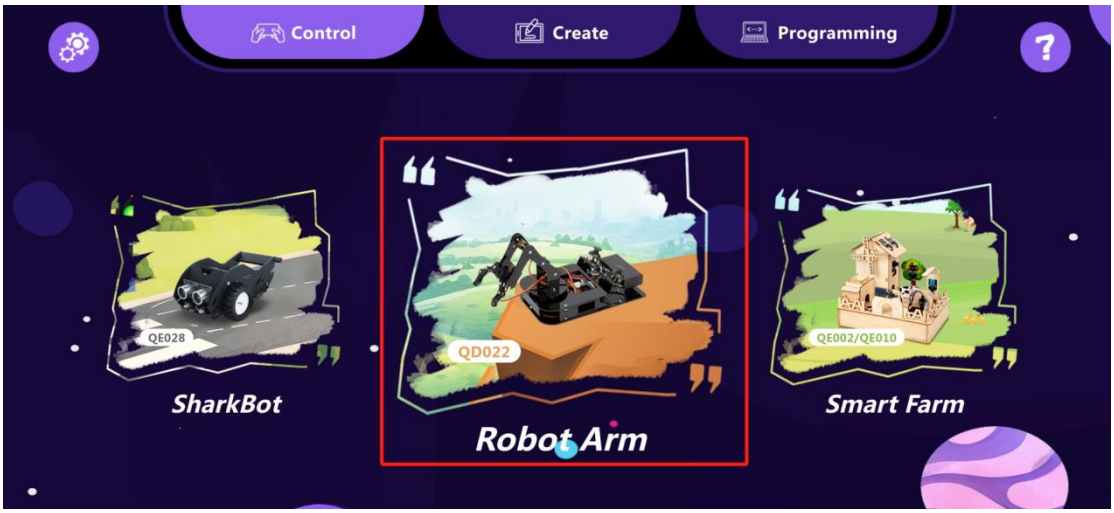
(2) Klicken Sie nach dem Öffnen der App auf die Bildschirmoberfläche.




(3) Rufen Sie die Auswahloberfläche auf und wählen Sie den Roboterarm aus.

Hinweis: Wenn Sie ein video über den betrieb der app ansehen möchten, Klicken Sie Bitte Auf Den Untenstehenden Link.

https://youtu.be/OJtV29RbKQs

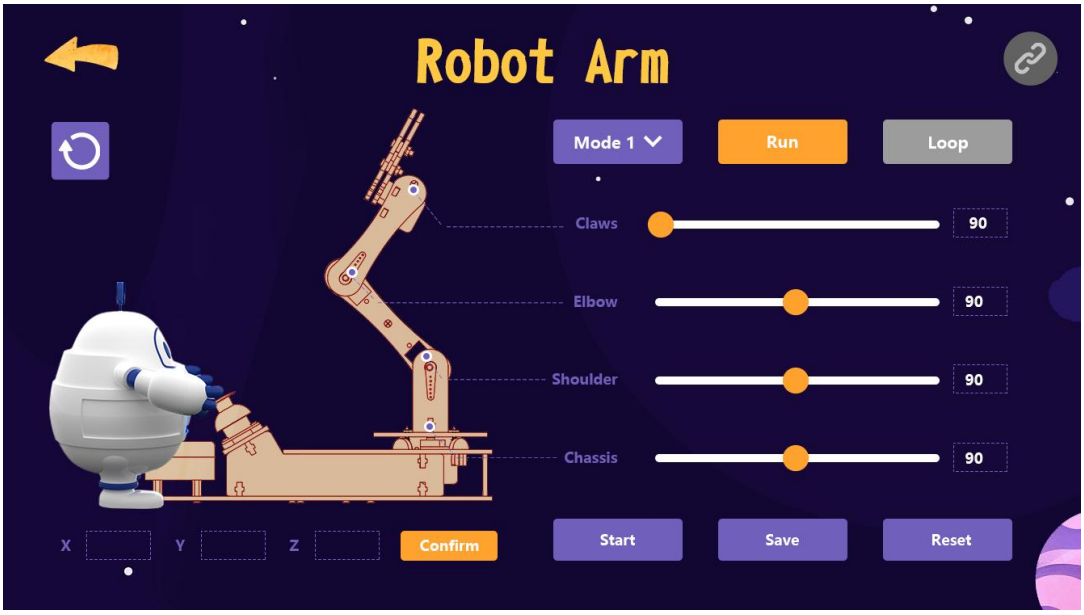



Klicken Sie auf eingang und wählen Sie „Control“, um auf die Kontrollseite zu gelangen.




Hinweis: Sie können auf die Schaltfläche „Build“ rechts klicken, um das Erstellungsvideo dieses Projekts anzusehen.

(4) Rufen Sie die Steuerungsoberfläche des Roboterarms auf (er kann noch nicht direkt gesteuert werden; Sie müssen das Programm hochladen).




## II. App-Steuerung des Roboterarms

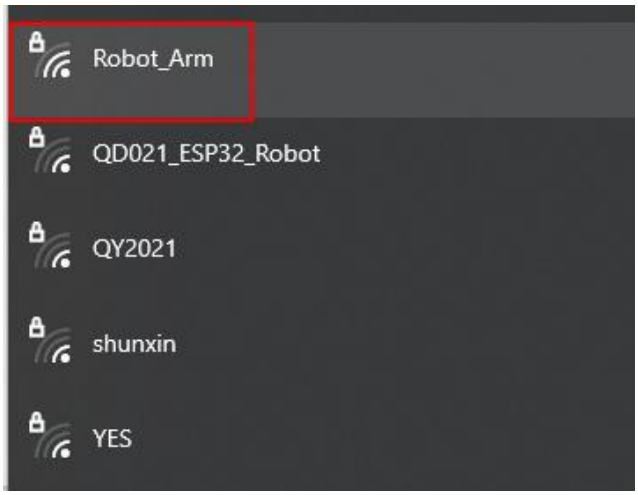
### 1. Arduino-Programm zum Steuern des Roboterarms per APP hochladen

Bevor der Roboterarm per APP gesteuert werden kann, muss zunächst das Arduino-Programm für die Kommunikation zwischen dem Roboterarm und der APP auf den Arm übertragen werden.

Öffnen Sie die Datei „APP_Controlled_Robot_Arm.ino“ im Verzeichnis „Deutsch\Arduino(Erfahrener Lernender)\2.Arduino Programm\Lektion 8\APP_Controlled_Robot_Arm“, verbinden Sie das ESP32-Entwicklungsboard per USB-Kabel mit dem Computer, wählen Sie das richtige Board und den richtigen Port aus, laden Sie den Code auf das ESP32-Entwicklungsboard hoch, versorgen Sie das Board mit Strom über das Batteriefach und schalten Sie den Schalter des Batteriefachs auf die Position „ON“.

**2. WiFi mit Roboterarm verbunden**

Verwenden Sie Ihren Computer oder Ihr Mobiltelefon, um nach WLAN-Netzwerken zu suchen und sich mit dem WLAN-Hotspot mit dem Namen „Robot_Arm“ und dem Passwort „12345678“ zu verbinden, wie in der Abbildung unten gezeigt.

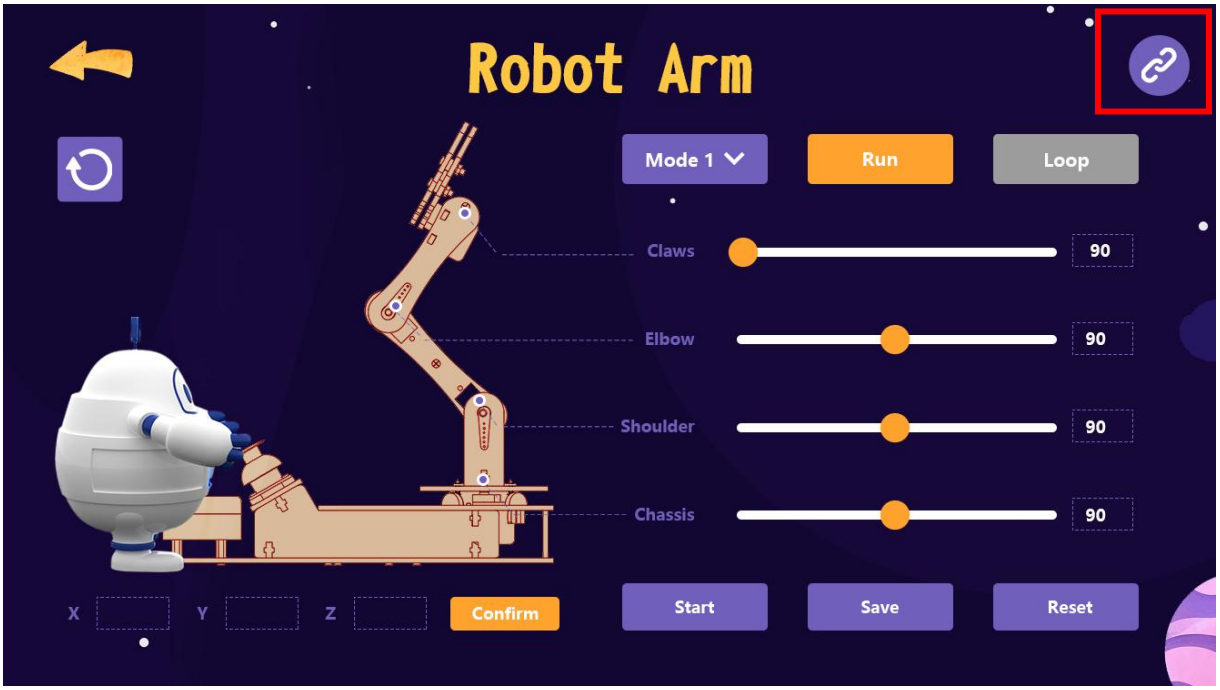



Hinweis: Der Name und das Passwort des Hotspots sind bereits im Programm definiert, aber der Benutzer kann sie anpassen und wenn wir mehrere Roboterarme haben, können wir jeden Roboterarm durch einen anderen WiFi-Namen unterscheiden.

```
const char* ssid = "Robot_Arm";// Wifi name
const char* password = "12345678";// WiFi password
```

### 3. App-Kontrolle Verwenden

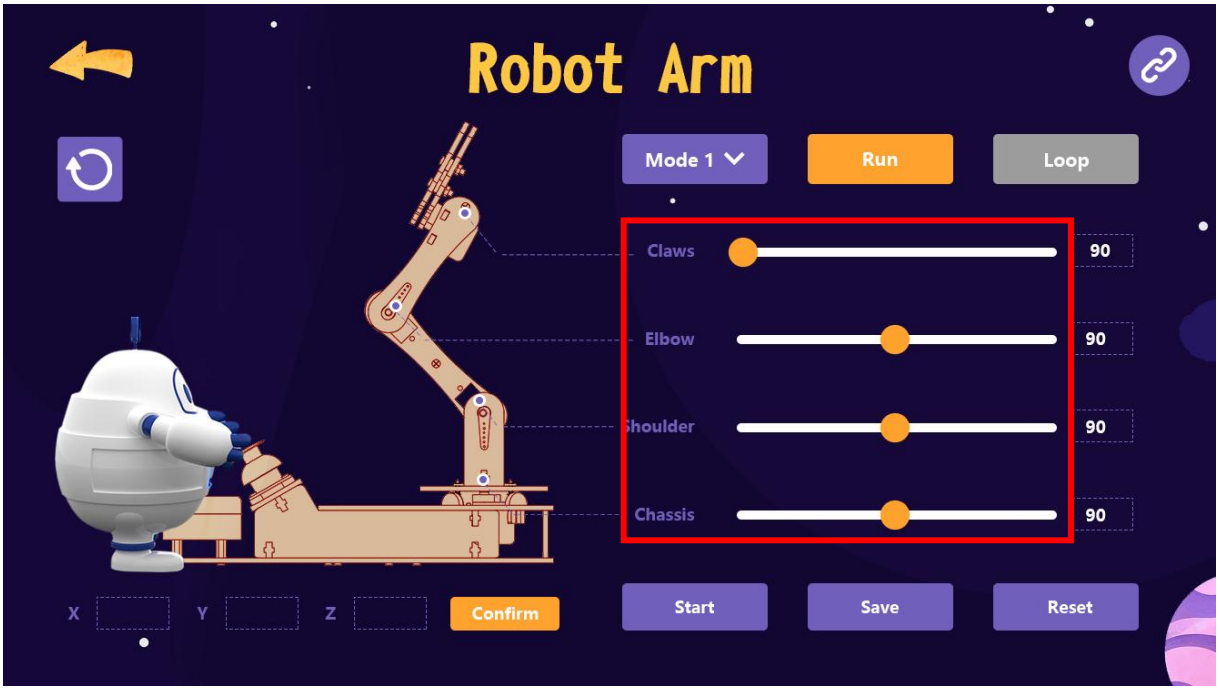
Nachdem Sie sich mit dem WLAN verbunden haben, klicken Sie einfach auf das Verbindungssymbol in der oberen rechten Ecke der App, um die Verbindung abzuschließen.



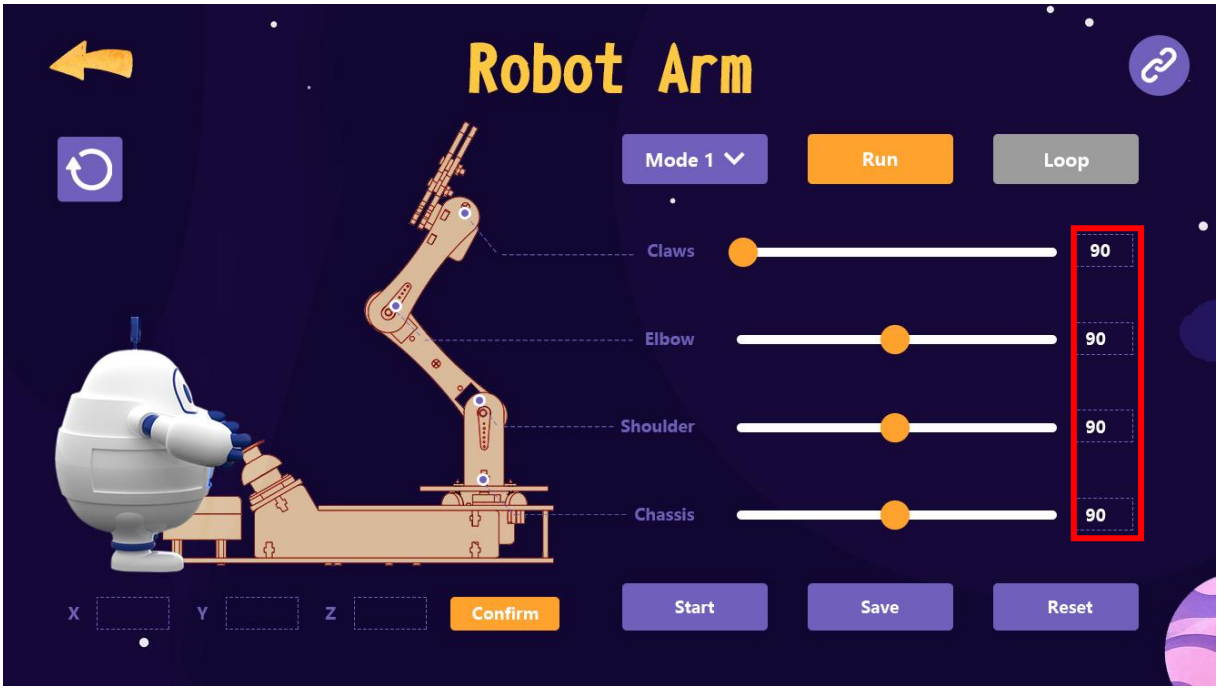

Nach Abschluss der oben genannten Schritte kehren Sie zur unten gezeigten Benutzeroberfläche zurück und können nun den Roboterarm steuern. Die wichtigsten Steuerungsfunktionen umfassen: Schiebereglersteuerung, Eingabefeldsteuerung, benutzerdefinierter Modus (Start, Ende, Speichern, Ausführen, Zurücksetzen), räumliche Positionierung und Positionswiederherstellung.

**Funktionen der Roboterarm-App:**

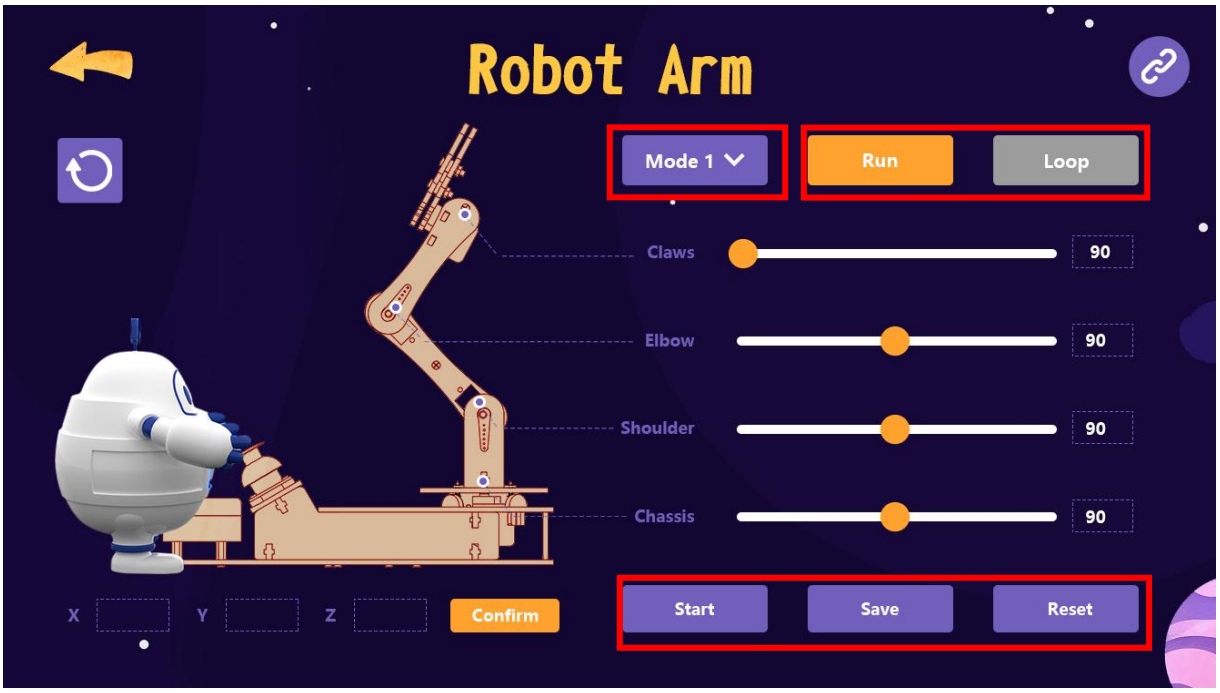
**(1) Schiebereglersteuerung**: Bewegen Sie den Schieberegler, der dem Servo der verschiedenen Teile des Roboterarms entspricht, um die Positionsänderung des Roboterarms zu steuern.



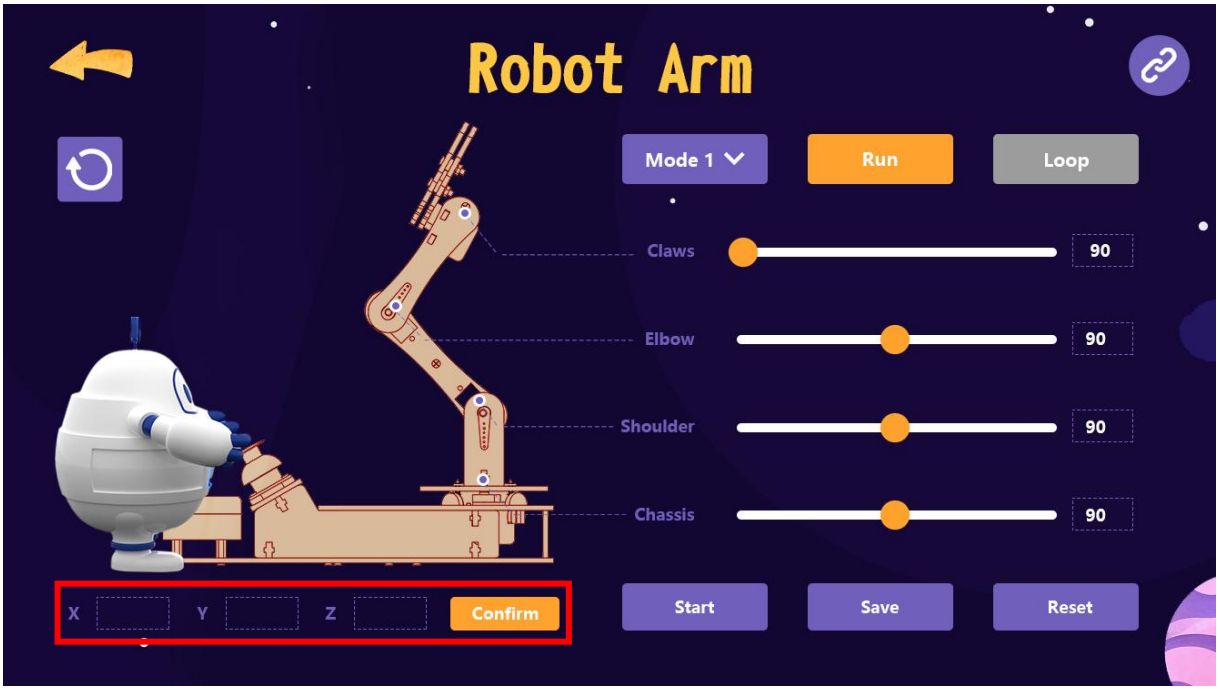

**(2) Eingabefeldsteuerung**: Rechts neben dem jeweiligen Schieberegler befindet sich ein Eingabefeld, in das Sie den entsprechenden Servowinkel eingeben können, um die Positionsänderung des Roboterarms zu steuern.



**(3) Aktionen speichern**: Klicken Sie auf „Mode 1“, um einen der 6 Modi (Modus 1–6) auszuwählen. In jedem Modus können 20 verschiedene Aktionsgruppen gespeichert werden. Klicken Sie auf „Run“, um die gespeicherte Aktion einmalig auszuführen, auf „Loop“, um die gespeicherte Aktion zu wiederholen, und auf „Reset“, um die gespeicherte Aktion zu löschen. Die Bedienung entspricht der Steuerung auf der Webseite.




**(4) Räumliche Positionierung**: Geben Sie die räumlichen Koordinaten x, y, z ein. Der Roboterarm bewegt sich entsprechend den angegebenen Koordinatenpunkten. (Hinweis: Aufgrund der Struktur des Roboterarms ist sein Bewegungsbereich begrenzt. Sollten die eingegebenen Koordinaten den Bewegungsbereich überschreiten, geben Sie sie bitte erneut ein.)




(5) Positionsinitialisierung: Klicken Sie auf das Aktualisierungssymbol in der oberen linken Ecke, um die Position des Roboterarms auf die Ausgangsposition zurückzusetzen.

