

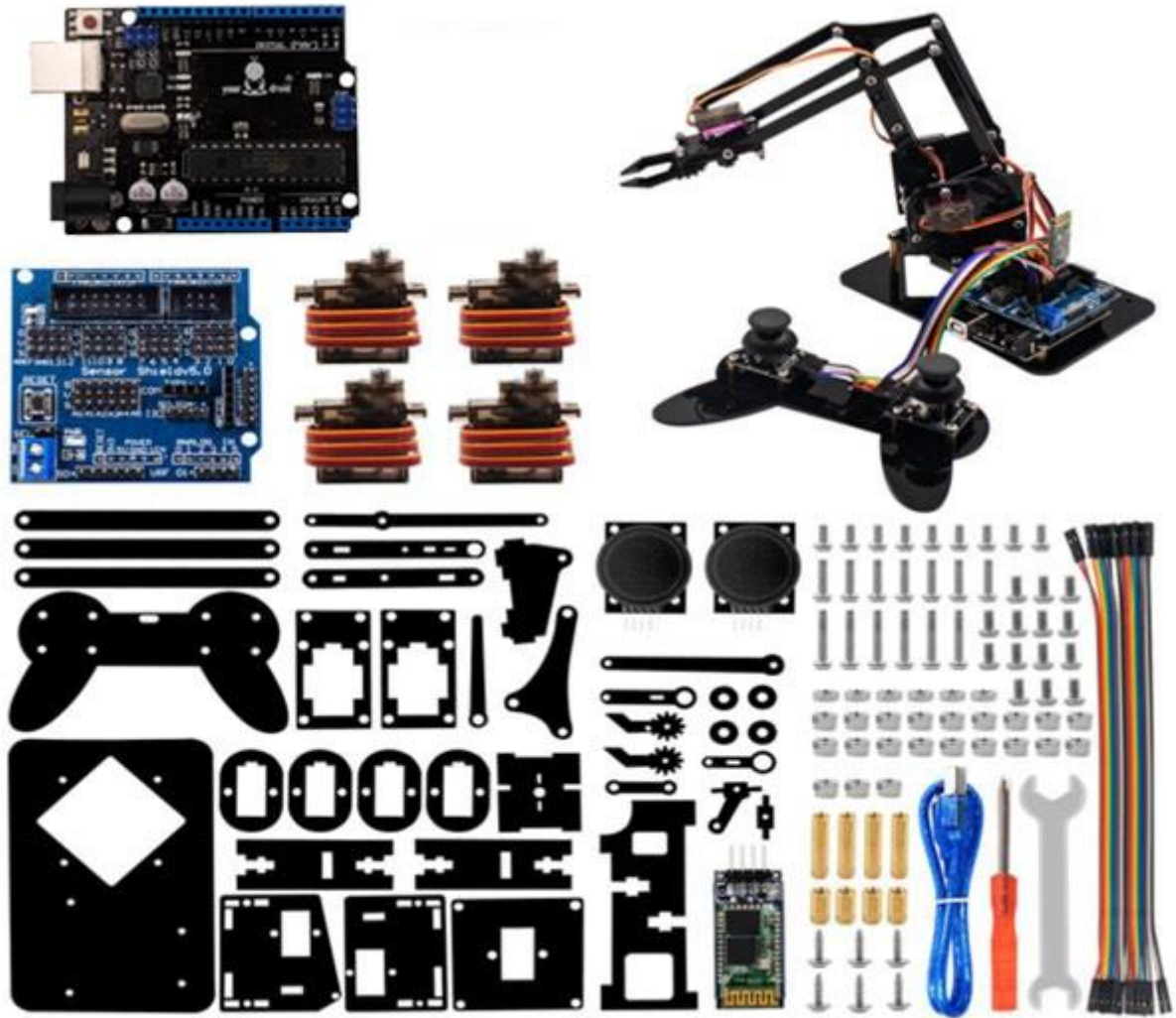
Anleitung yourDroid Roboterarm für Arduino UNO



Inhaltsverzeichnis

1.	Arduino IDE installieren	2
2.	Bibliotheken hinzufügen und den seriellen Monitor öffnen	8
3.	Blink Beispiel	14
4.	Aufbau	19
5.	Servo	43
6.	Analoges Joystick Modul	45
7.	Roboterarm mit zwei Joysticks steuern	46
8.	Roboterarm mit Bluetooth steuern	48

Packliste



- 1 Stück Uno r3 Board
- 4 Stück MG90S Servo
- 1 Stück V5.0 extension board
- 1 Stück USB-Kabel
- 2 Stück Joystick Modul
- 1 Stück 20PIN Buchse auf Buchse Dupont Kabel
- 1 Stück Bluetooth Modul
- 1 Stück Schraubendreher
- 1 Stück Schraubenschlüssel
- 1 Set Acrylteile
- 1 Set Schrauben
- 2 Stück Bunding belt
- 4 Stück Gummifüße

1. Arduino IDE installieren

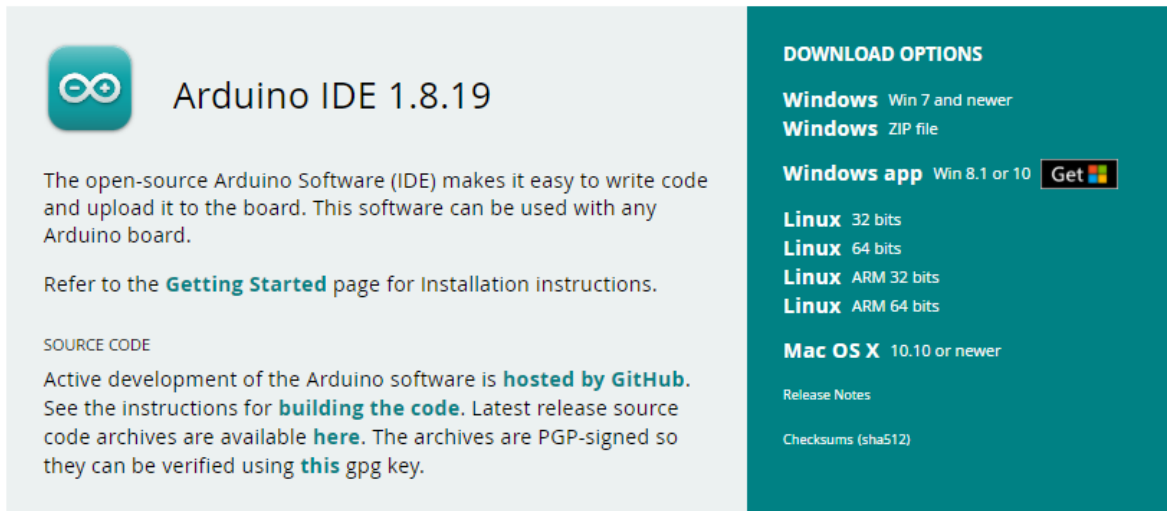
Die Arduino Integrated Development Environment ist die Entwicklungsumgebung und Software der Arduino Plattform.

In dieser Lektion lernen Sie Ihren Computer einzurichten, um Arduino Boards nutzen zu können.

Die Arduino Software erlaubt es Ihnen Arduinos und andere Entwicklungsplatinen zu programmieren. Die Software ist verfügbar für Windows, Mac und Linux. Der Installationsprozess ist für alle Plattformen unterschiedlich.

Arduino IDE herunterladen

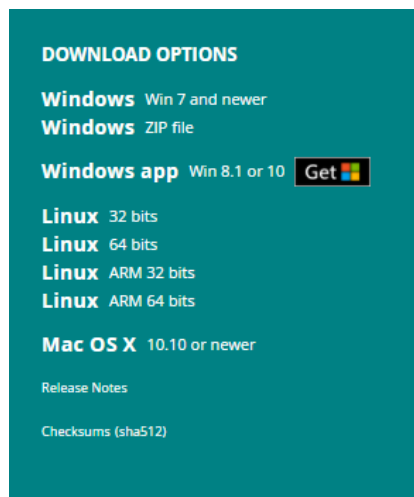
Schritt 1: <https://www.arduino.cc/en/software> aufrufen



The screenshot shows the Arduino IDE 1.8.19 download page. On the left, there's a section with the Arduino logo and the text "Arduino IDE 1.8.19". Below this, it says "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board." and "Refer to the **Getting Started** page for Installation instructions." There's also a "SOURCE CODE" section mentioning GitHub. On the right, a teal sidebar titled "DOWNLOAD OPTIONS" lists various download links for Windows (Win 7 and newer, ZIP file, Windows app), Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), and Mac OS X (10.10 or newer). It also includes links for "Release Notes" and "Checksums (sha512)".

Auf dieser Webseite finden Sie immer die aktuelle Version. Die aktuelle Version ist vermutlich neuer als die aus dieser Anleitung.

Schritt 2: Passenden Installer herunterladen



This is a zoomed-in view of the "DOWNLOAD OPTIONS" sidebar from the previous screenshot. It lists the following options: Windows (Win 7 and newer, ZIP file, Windows app), Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), and Mac OS X (10.10 or newer). It also includes links for "Release Notes" and "Checksums (sha512)".

„Win 7 and newer“ auswählen, falls Sie nicht die App aus dem Microsoft Store installieren möchten.

Schritt 3: Download Starten

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **58.947.663** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD



Learn more about [donating to Arduino](#).

„Just Download“ auswählen.

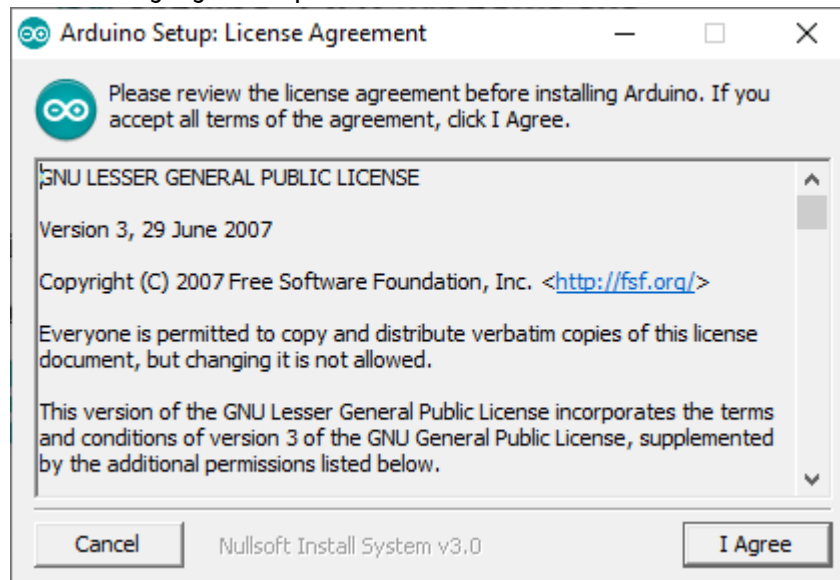
Installation unter Windows

1. Installationsdatei starten

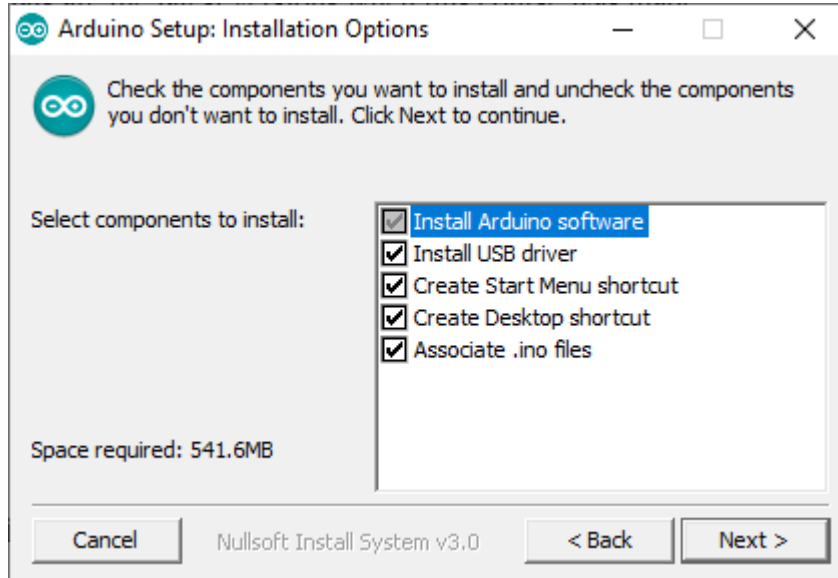


arduino-1.8.19-windows.exe

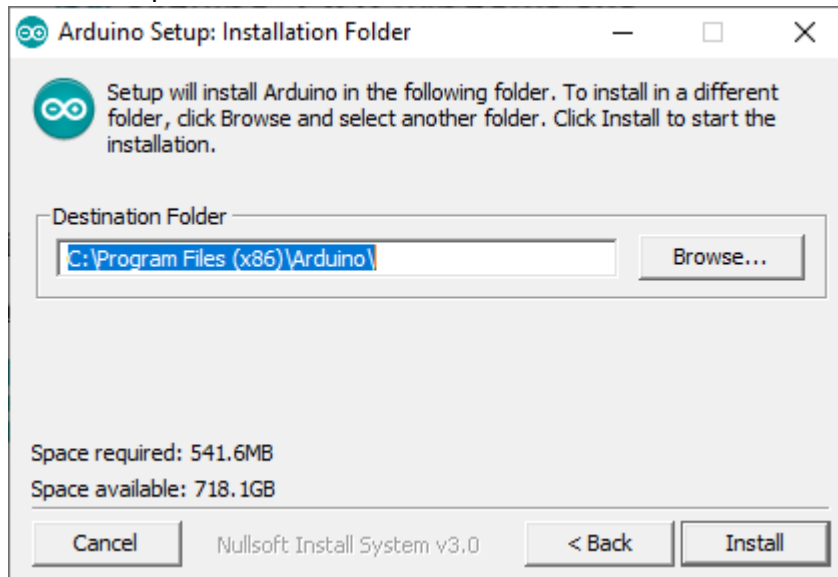
2. Lizenzbedingungen akzeptieren



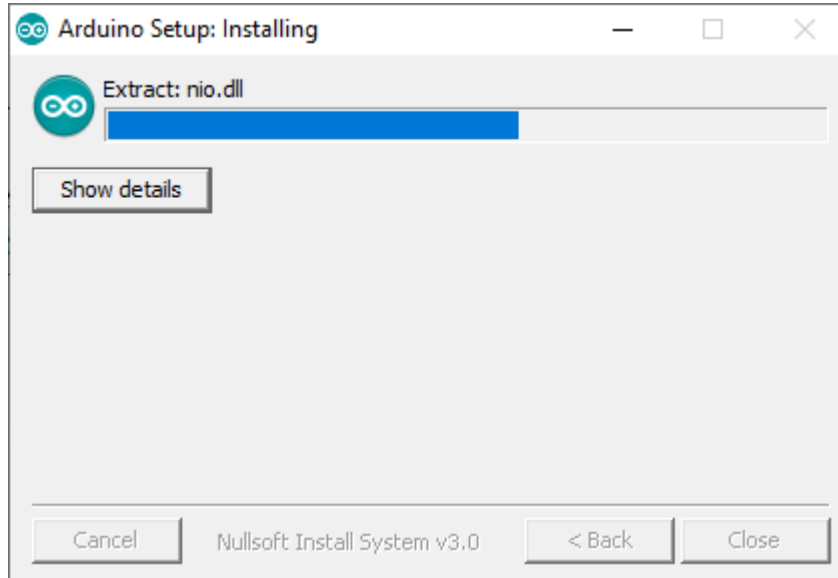
3. Auf „Next“ klicken



4. Installationspfad auswählen oder „Install“ drücken



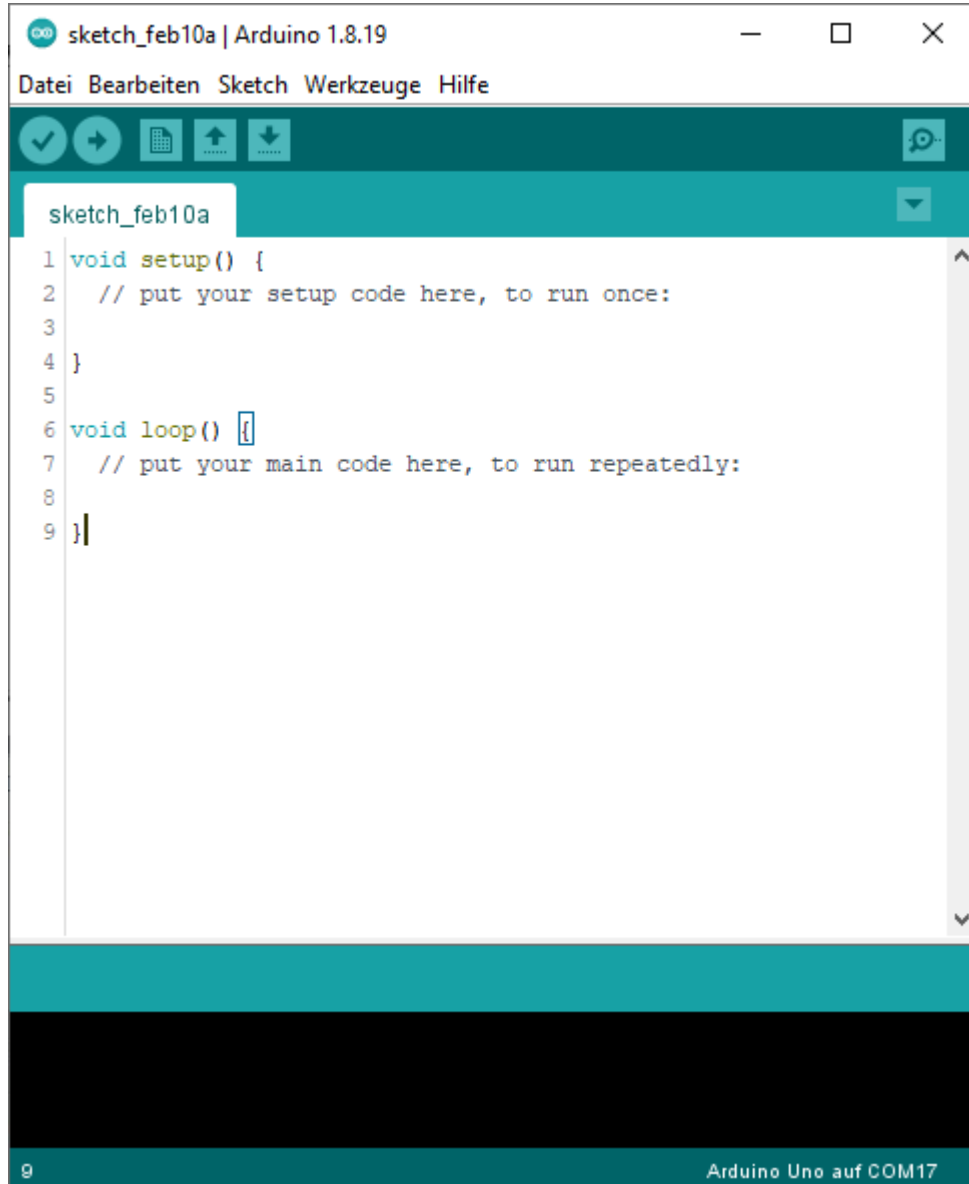
5. Installation abwarten



6. Arduino USB-Treiber installieren lassen



7. Zurück zum Desktop und die Verknüpfung der Arduino IDE doppelklicken



8. Das war die Installation unter Windows.

Installation unter Linux

1. Package extrahieren
2. Install Script starten
Entweder im entpackten Archiv die „install.sh“ starten oder im Terminal zum Dateipfad navigieren und „./install.sh“ eingeben
3. Die Arduino IDE kann nun gestartet werden. Auf dem Desktop finden Sie die Verknüpfung.

Empfohlen: Wenn Sie ein Ubuntu System nutzen, können Sie stattdessen die Installation über das Software-Center von Ubuntu durchführen.

Arduino Installation unter MacOS

1. Die Installationsdatei ist im Zip-Format. Wenn Sie Safari nutzen, wird der Inhalt automatisch entpackt. Die Installationsdatei doppelklicken, falls noch nicht bereits installiert, werden Sie aufgefordert die Java Runtime Library zu installieren.
2. Die Arduino IDE kann nun gestartet werden.

2. Bibliotheken hinzufügen und den seriellen Monitor öffnen

Installieren zusätzlicher Arduino Bibliotheken

Sobald Sie sich mit der Arduino IDE vertraut gemacht haben und die integrierten Beispiele ausprobiert haben, können Sie mit Bibliotheken mehr Potenzial aus Ihrem Arduino Board schöpfen.

Was sind Bibliotheken?

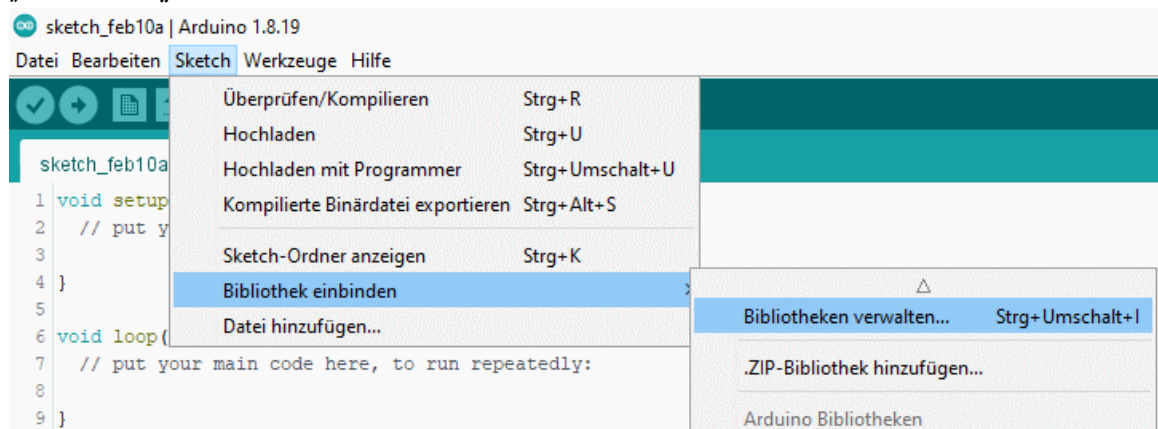
Bibliotheken sind Codesammlungen um Sensoren, Module, Displays oder Funktionen wesentlich einfacher programmieren zu können. Zum Beispiel vereinfacht die integrierte LiquidCrystal.h Bibliothek die Kommunikation zu LCD Displays, um Zeichenfolgen unkompliziert ausgeben können. Es gibt tausende zusätzliche Bibliotheken im Internet kostenfrei zum Herunterladen. Die integrierten Bibliotheken besitzen Beispiele, die Sie unter „Datei – Beispiele“ aufrufen können. Zusätzliche Bibliotheken müssen heruntergeladen und installiert werden.

Wie installiere ich weitere Bibliotheken?

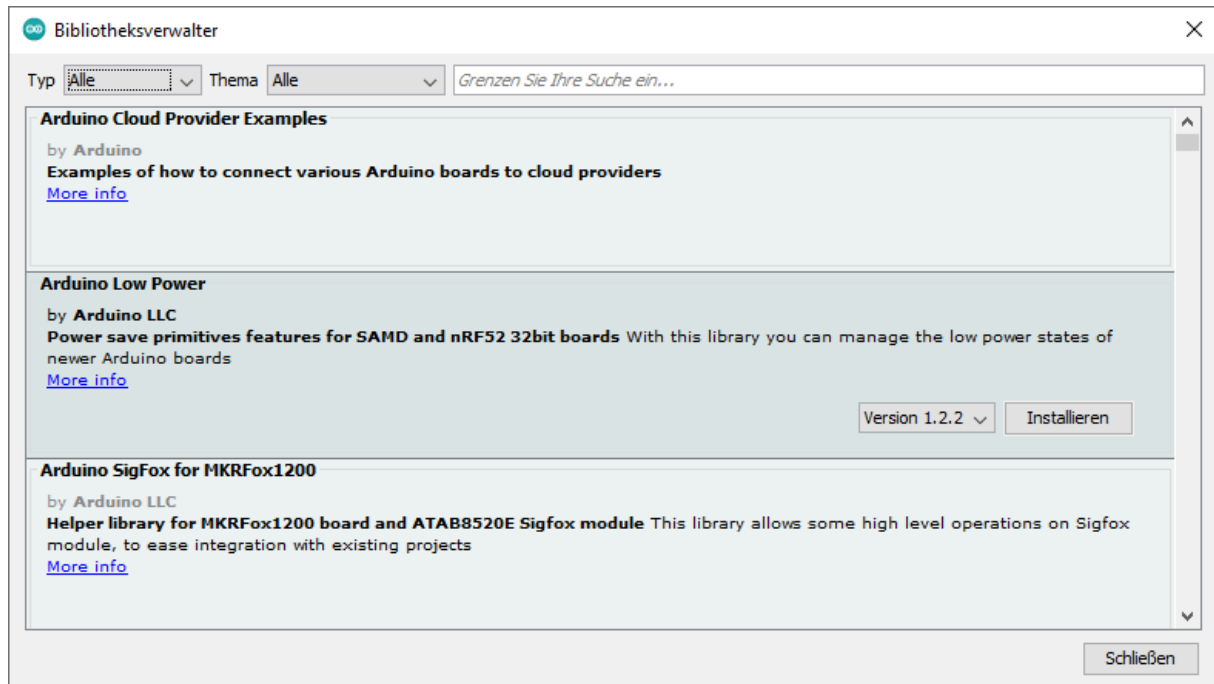
Für die Installation von weiteren Bibliotheken gehen wir hier auf die verschiedenen Möglichkeiten ein.

Bibliotheken mit der Arduino IDE installieren

Seit Version 1.8.0 gibt es den praktischen Library Manager mit dem Sie im Handumdrehen Bibliotheken suchen, herunterladen und installieren können. Öffnen Sie die Arduino IDE und klicken Sie im Reiter „Sketch“ auf **„Bibliotheken verwalten...“**

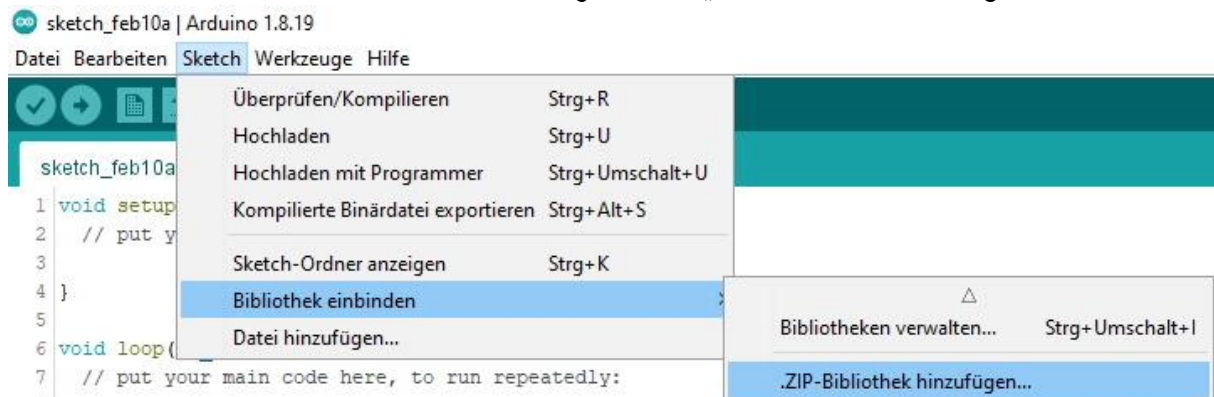


Es öffnet sich der Library Manager, mit dem Sie Bibliotheken installieren, aktualisieren oder entfernen können. Gelegentlich werden Sie in der Arduino IDE unten rechts eine Meldung erhalten, dass für installierte Bibliotheken Aktualisierungen verfügbar sind.

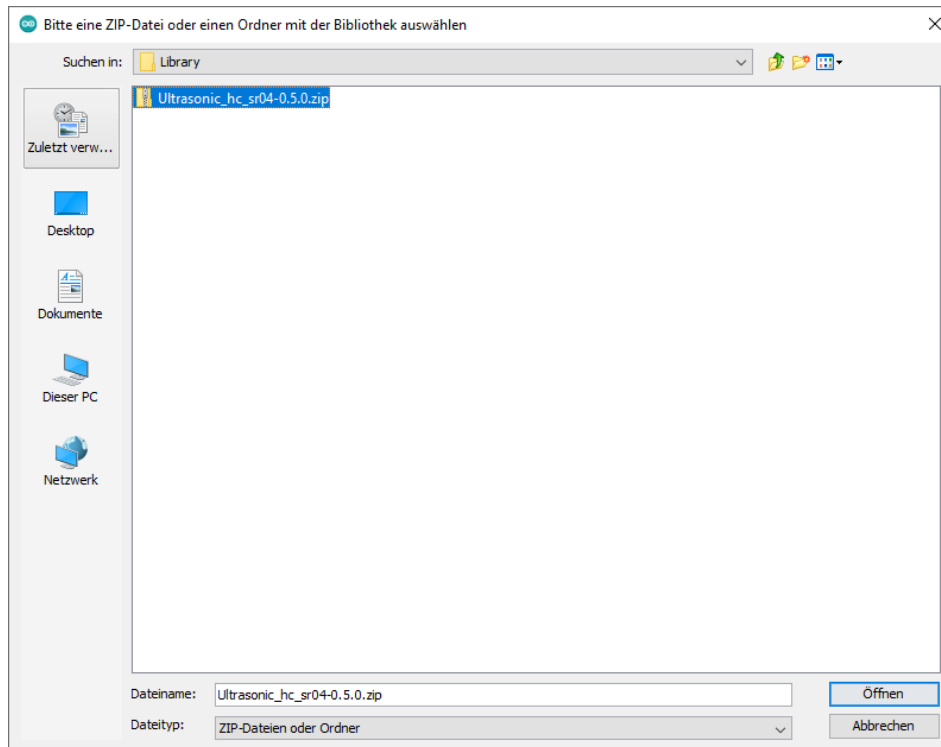


.zip Bibliothek importieren

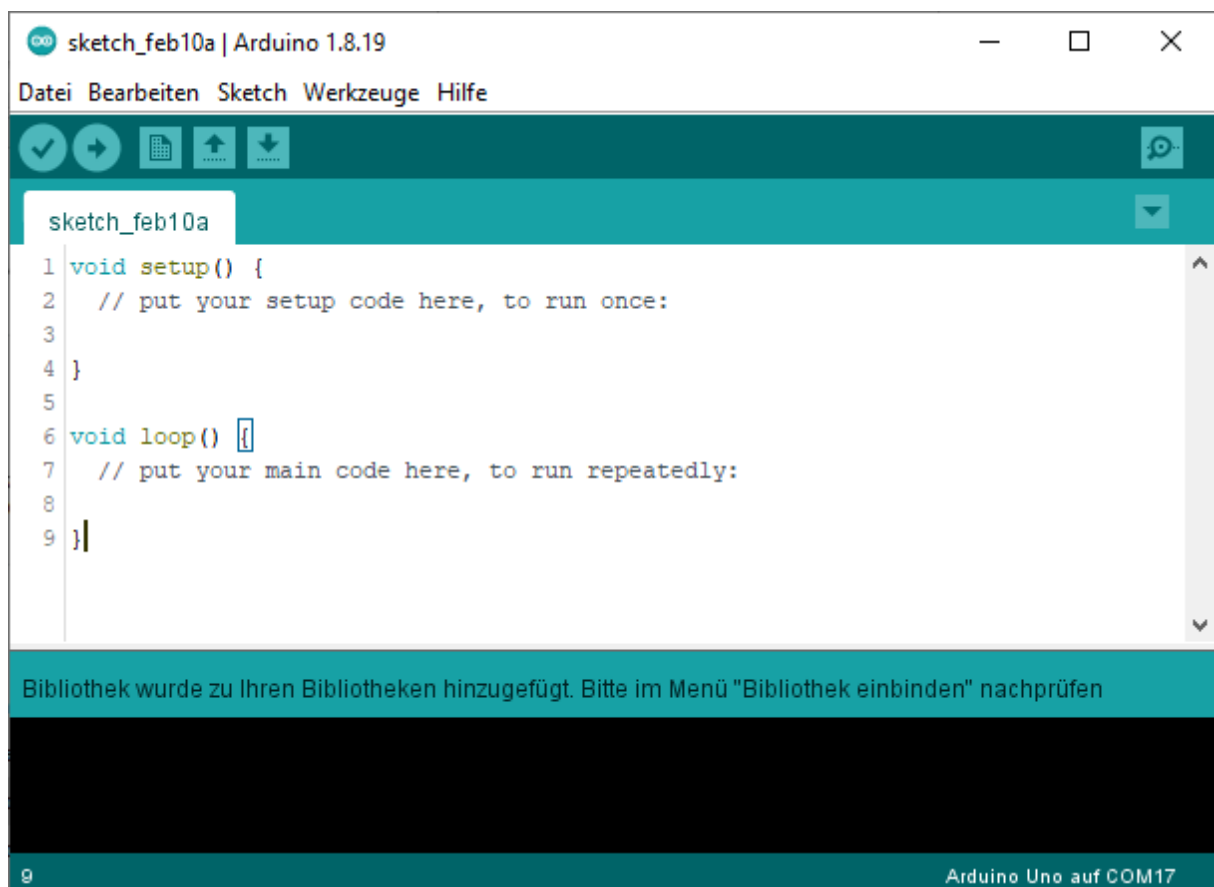
Es kann vorkommen, dass eine Bibliothek nicht im Bibliotheksverwalter zu finden ist. In diesem Fall muss man auf eine externe Quelle zurückgreifen und die Bibliothek importieren. Hierzu in der Arduino IDE unter „Sketch“ zu „Bibliothek einbinden“ navigieren und **„.ZIP Bibliothek hinzufügen...“** anklicken.



Sie werden aufgefordert mit dem Dateexplorer eine ZIP Bibliothek zu öffnen. Navigieren Sie zum entsprechenden Dateipfad und wählen Sie eine ZIP Bibliothek aus.



Wenn die Bibliothek erfolgreich eingebunden wurde, erscheint im Hauptfenster folgende Nachricht:



Nun Können Sie, wie aufgefordert, unter „Sketch -> Bibliotheken einbinden“ nach unten scrollen um zu schauen ob die Bibliothek aufgelistet wird.

Hinweis: Die Beispiele unter „Datei – Beispiele“ tauchen erst nach einem Neustart der Arduino auf.

Bibliotheken manuell installieren

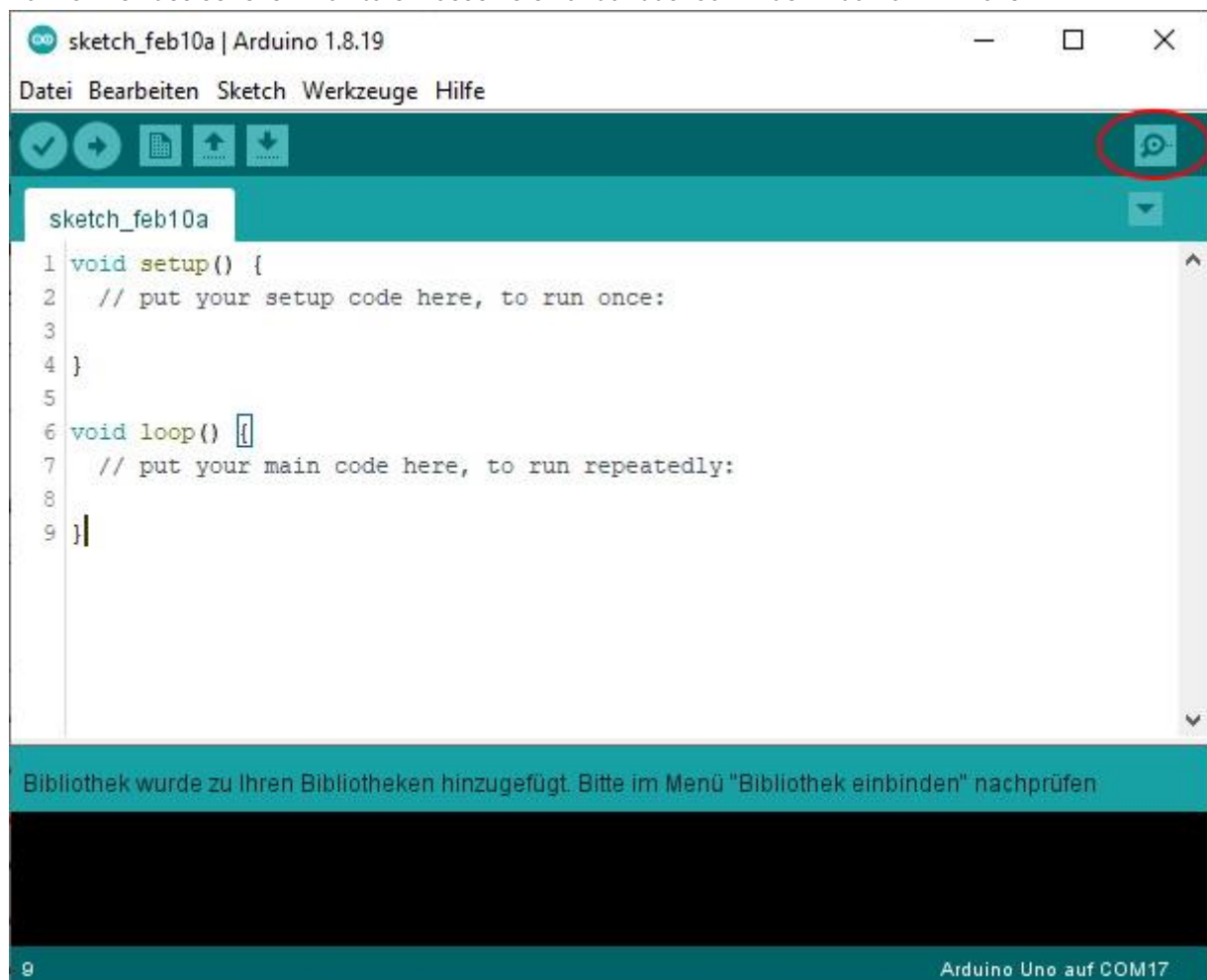
Um Bibliotheken manuell zu installieren, müssen Sie die .ZIP Datei entpacken und den Ordner nach „Dokumente/Arduino/Libraries“ kopieren. Wichtig ist, dass der Ordner wie die Bibliothek heißt und eine .cpp und .h Datei enthält. Nach einem Neustart befindet sich die Bibliothek nun ebenfalls unter „Sketch – Bibliotheken einbinden“ in der Liste.

Serieller Monitor

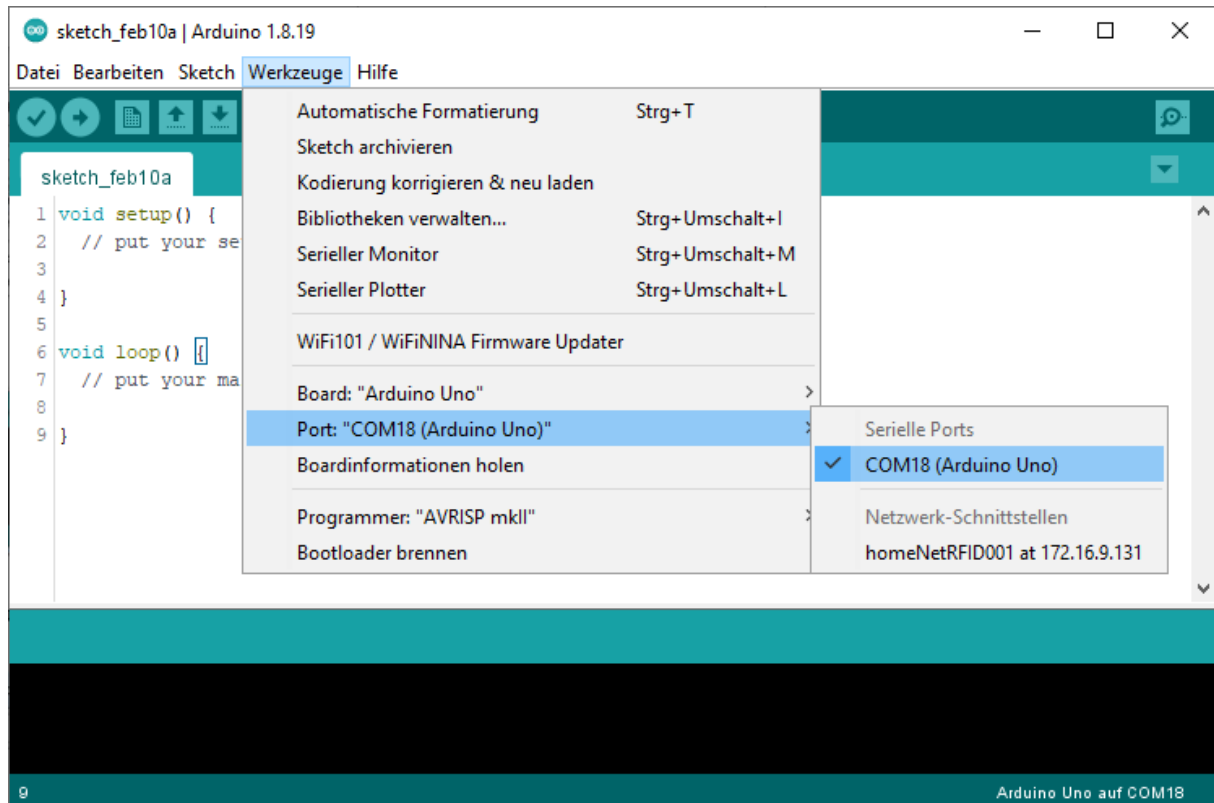
Der serielle Monitor ist ein wichtiger Teil der Arduino IDE, um mit angeschlossenen Microcontrollern kommunizieren und sich auf Fehlersuche begeben zu können. Mit diesem Werkzeug können Sie ganz schnell Messwerte ausgeben lassen, dem Arduino Befehle erteilen oder Programmfunktionen auf Fehler überprüfen.

Eine Verbindung herstellen

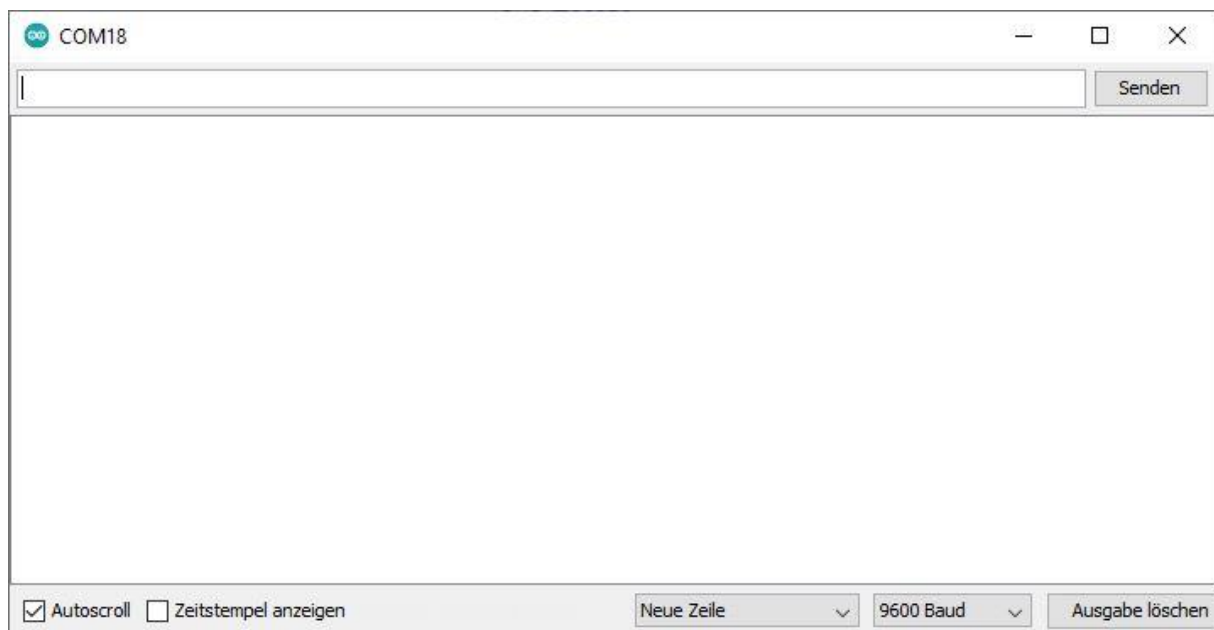
Zum Öffnen des seriellen Monitors müssen Sie nur auf das Icon in der Arduino IDE klicken



Die Port-Auswahl vom seriellen Monitor funktioniert wie beim Hochladen von Arduino Codes.

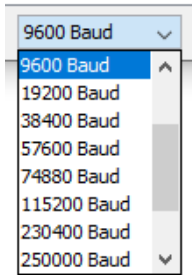


Einmal geöffnet sieht er ungefähr so aus:



Einstellungen des seriellen Monitors

Der serielle Monitor der Arduino IDE hat nur wenige Einstellungen, die aber für die meisten Anwendungen ausreichen dürften. Die wichtigste Einstellung ist die baud rate.



Mit der baud rate wird die **Datenrate in Bit pro Sekunde** für die serielle Datenübertragung festgelegt. Wenn die baud rate falsch eingestellt ist, werden die Daten nicht korrekt angezeigt und die Kommunikation wird nicht funktionieren. **Autoscroll** ist sehr praktisch, kann aber bei Bedarf ausgeschaltet werden.

Vorteile vom seriellen Monitor

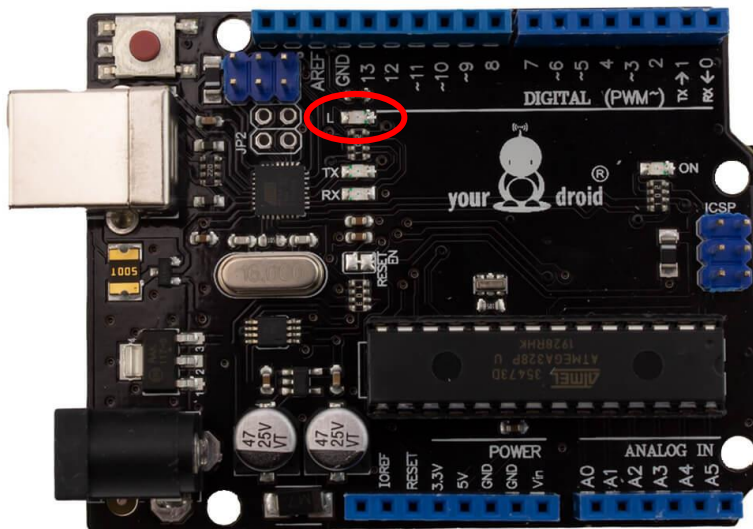
Der serielle Monitor ist eine gute und sehr einfache Möglichkeit eine serielle Verbindung mit Ihrem Arduino herzustellen. Wenn Sie sowieso in der Arduino IDE programmieren, liegt es nahe die integrierten Werkzeuge zu verwenden.

Nachteile vom seriellen Monitor

Durch die dürftigen Einstellungsmöglichkeiten eignet sich der serielle Monitor nicht für fortgeschrittene Anwendungen. Die Grenzen bereits sind schnell erreicht, wenn Ihre Anwendung eine nicht aufgeführte baud rate verlangt.

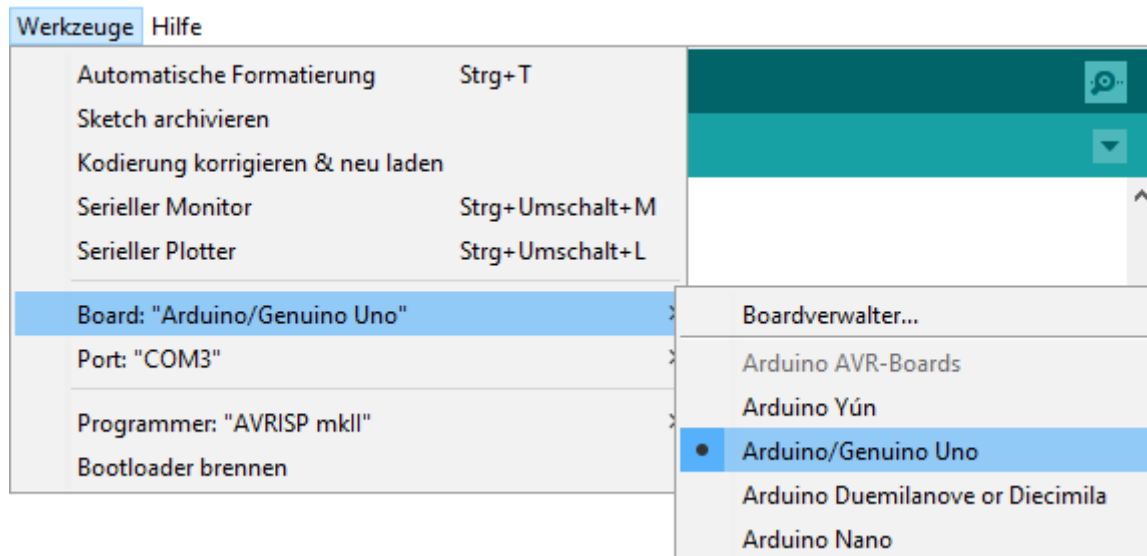
3. Blink Beispiel

In diesem Kapitel zeigen wir Ihnen durch das beliebte Blink Beispiel, wie Sie das Arduino UNO kompatible Board über die offizielle Software programmieren. Auf dem Arduino UNO ist eine kleine LED angebracht, die wir steuern möchten. Standardmäßig wird das Board bereits mit dem Blink Sketch ausgeliefert, also sollte die LED bereits blinken sobald es angeschlossen wird.



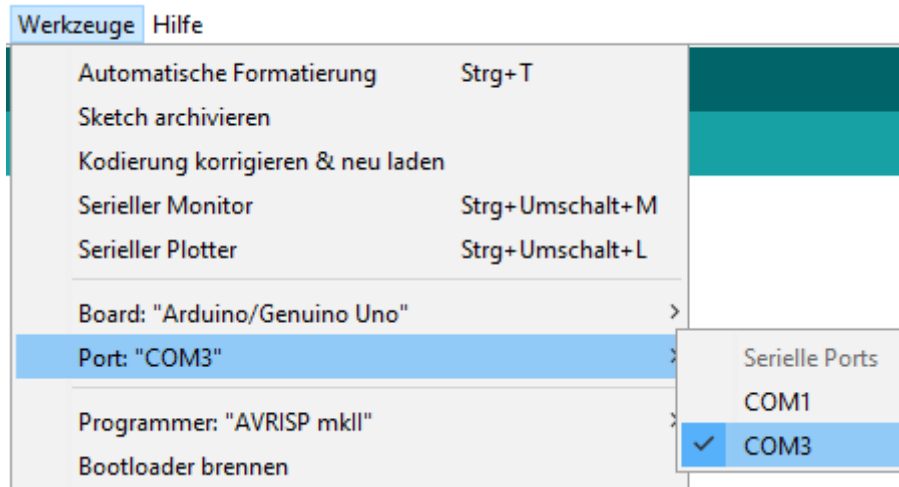
Nachdem wir die Arduino-Software installiert haben, schließen wir das Arduino UNO Board über USB an den Computer an.

Nun öffnen wir die Arduino-Software und stellen sicher, dass bei "Werkzeuge - Board" der Arduino UNO ausgewählt ist. und der serielle Port richtig eingestellt ist. Unter Windows ist es meistens Port "COM 3" oder höher, denn Port 1-2 sind für das System reserviert.



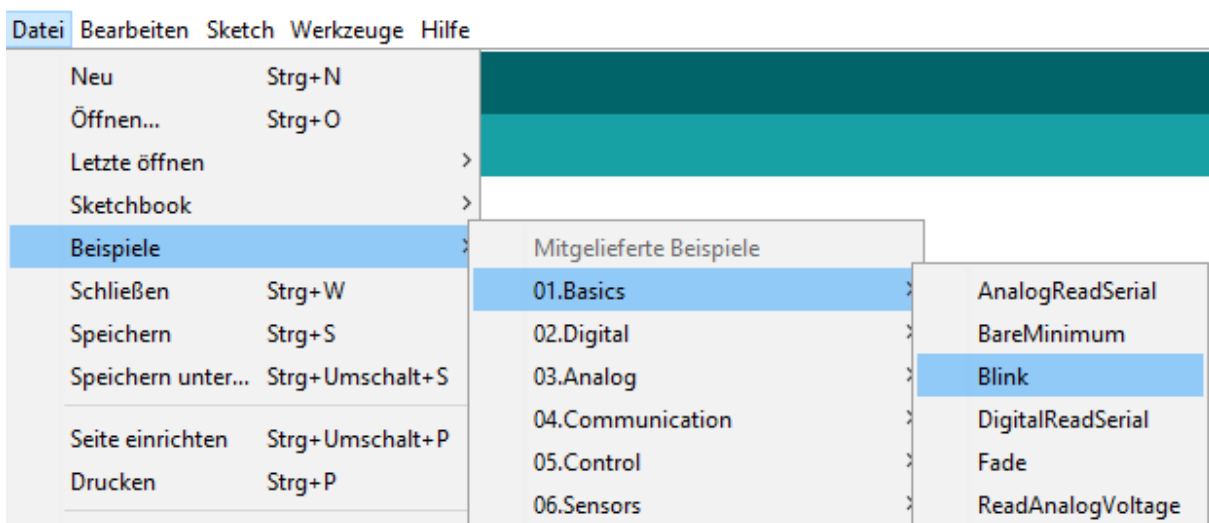
Als nächstes stellen wir den richtigen Port ein. Unter Windows ist es meistens Port "COM 3" oder höher, denn Port 1-2 sind in der Regel für das System reserviert.

Achtung: Der richtige Port auf Ihrem System wird vermutlich nicht derselbe wie auf den Screenshots sein!



Die Arduino-Software liefert einige Programmbeispiele mit, für dieses Projekt benutzen wir einfach das "Blink"-Beispiel.

Für dieses Beispiel klicken wir oben links auf "Datei" und wählen unter dem Reiter "Beispiele - 01.Basics - Blink" aus.



Dann öffnet sich ein neuer Sketch, welcher hier nochmal mit deutschen Kommentaren versehen ist:

```

/*
  Blink
  Schaltet eine LED für 1 Sekunde an und für eine Sekunde aus.
*/

// Die Setup-Funktion wird einmalig nach jedem Start oder Reset ausgeführt
void setup() {
  // digitaler pin 13 wird als Output deklariert
  pinMode(13, OUTPUT);
}

// Die Loop-funktion wird immer und immer wieder ausgeführt
void loop() {

```

```
digitalWrite(13, HIGH); // Schaltet die LED an (HIGH ist die Spannungsversorgung)
delay(1000);           // 1000 ms (= 1 Sekunde) Verzögerung
digitalWrite(13, LOW); // Schaltet die LED aus indem die Spannungsversorgung auf LOW gesetzt wird
delay(1000);           // 1 Sekunde Verzögerung
}
```

Hinweis: Die grau hinterlegten **Kommentare nach //** haben keine Programmfunktion. Sie dienen nur der Erklärung des Programms. Zwischen **/*** und ***/** befinden sich Block-Kommentare, die sich über mehrere Zeilen erstrecken können.

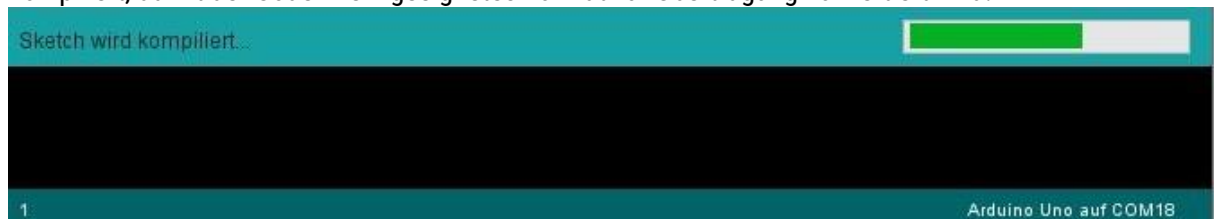
Jeder Arduino Sketch benötigt eine **„Setup“-Funktion**, die einmalig beim Start ausgeführt wird. In unserem Beispiel wird hier nur die Anweisung erteilt, dass der Pin 13 als Ausgang deklariert wird. Weitere Anweisungen können Sie zwischen den geschweiften Klammern hinzufügen.

Die **„Loop“-Funktion** ist ebenfalls Pflicht und wird nach dem Setup immer wieder ausgeführt. In der Loop-Funktion wird die LED angeschaltet (Pin 13 auf HIGH), eine Pause von 1000ms eingelegt und die LED ausgeschaltet (Pin 13 auf LOW).

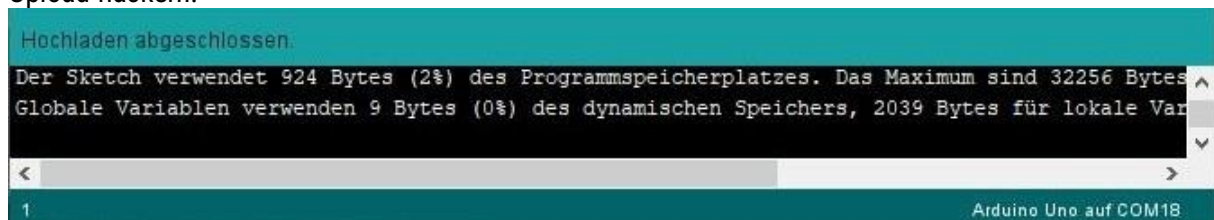
Dieses kleine Programm laden wir nun auf unser Board, indem wir auf die Schaltfläche klicken:



Im unteren Bereich können Sie den Fortschritt des Vorgangs verfolgen. Der Sketch wird zunächst kompiliert, damit der Code in ein geeignetes Format zur Übertragung konvertiert wird.

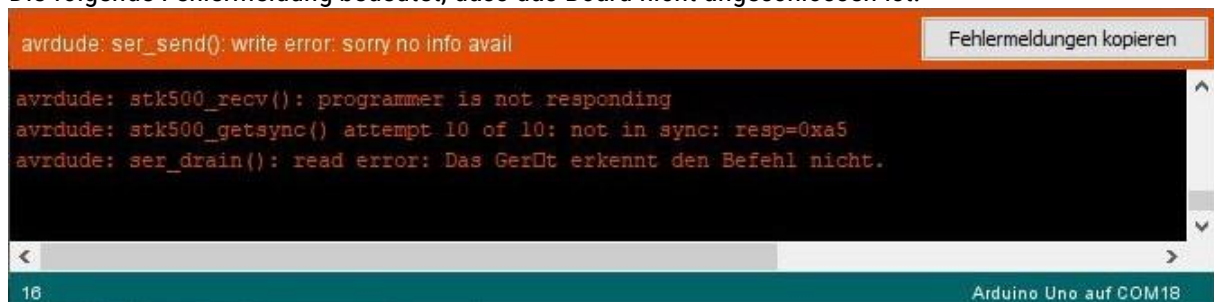


Im Anschluss wird der Sketch hochgeladen, die anderen LEDs auf dem Board sollten während dem Upload flackern.



Fehlerbehebung

Die folgende Fehlermeldung bedeutet, dass das Board nicht angeschlossen ist.



Mögliche Fehlerursachen und Lösungen

- **Board nicht angeschlossen.** Manchmal hilft es auch einen anderen USB-Port auszuprobieren.
- **Treiber nicht richtig installiert.** In diesem Fall müssen Sie im Geräte-Manager Ihr Arduino finden und im Fenster „Eigenschaften“ den Treiber deinstallieren und anschließen neu installieren.
- **Falscher Port ausgewählt.** In der Arduino IDE unter „Werkzeuge“ den Port erneut überprüfen.
- **Falsches Board ausgewählt.** In der Arduino IDE unter „Werkzeuge“ das Board erneut überprüfen.



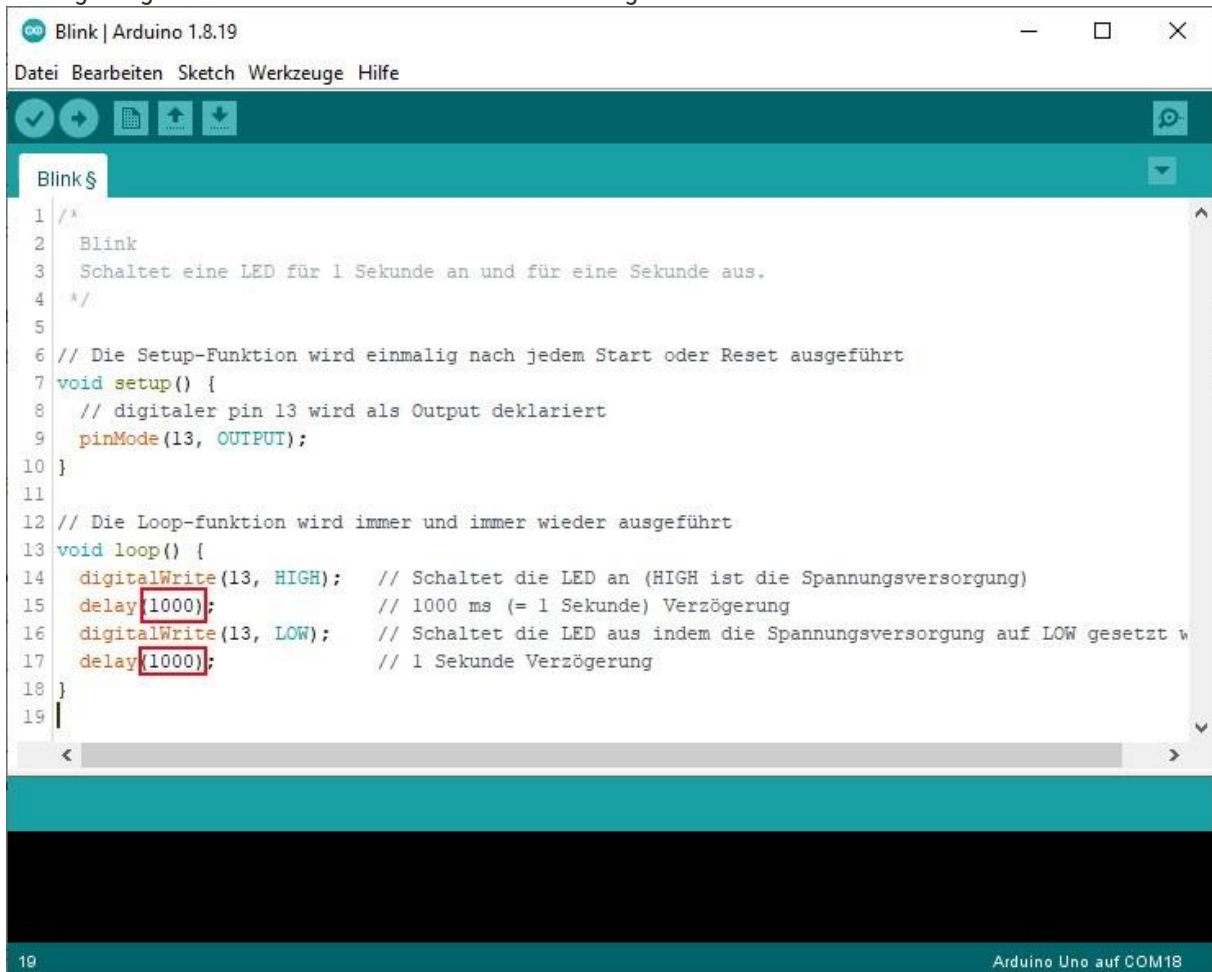
The screenshot shows the Arduino IDE's error console. The top bar is orange and contains the text "expected ';' before '}' token" and a button "Fehlermeldungen kopieren". The main area is black with white text showing the error message: "expected ';' before '}' token". Below this, it says "exit status 1" and "expected ';' before '}' token". The bottom status bar is teal and shows "29" and "Arduino Uno auf COM18".

Diese Fehlermeldung wird jeder früher oder später erhalten. In der Programmiersprache von Arduino muss hinter jeder Zeile, also jedem Befehl ein Semikolon „;“ stehen. Andernfalls bricht die Kompilierung ab und erzeugt diese Fehlermeldung.

Lösung: Die entsprechende Zeile wird in der Arduino rot hinterlegt und kann korrigiert werden.

Ergebnis

Wenn alles funktioniert hat, blinkt unsere LED im 1-Sekunden-Takt. Als nächstes können Sie die Verzögerungen ändern und die LED schneller oder langsamer blinken lassen.



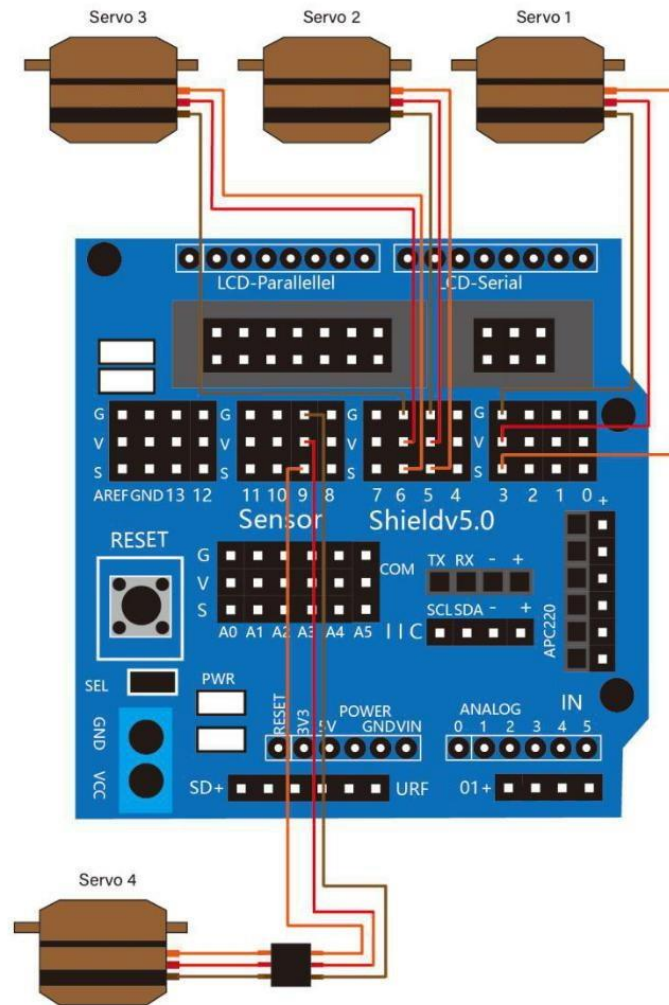
```
1 /*
2  Blink
3  Schaltet eine LED für 1 Sekunde an und für eine Sekunde aus.
4  */
5
6  // Die Setup-Funktion wird einmalig nach jedem Start oder Reset ausgeführt
7  void setup() {
8    // digitaler pin 13 wird als Output deklariert
9    pinMode(13, OUTPUT);
10 }
11
12 // Die Loop-funktion wird immer und immer wieder ausgeführt
13 void loop() {
14   digitalWrite(13, HIGH); // Schaltet die LED an (HIGH ist die Spannungsversorgung)
15   delay(1000); // 1000 ms (= 1 Sekunde) Verzögerung
16   digitalWrite(13, LOW); // Schaltet die LED aus indem die Spannungsversorgung auf LOW gesetzt w
17   delay(1000); // 1 Sekunde Verzögerung
18 }
19 |
```

Ändern Sie den Delay von 1000 auf 500, um die LED alle 500 ms blinken zu lassen. Nach der Änderung den Sketch erneut uploaden und Sie müssten sehen, dass die LED doppelt so schnell blinkt.

4. Aufbau

Vor dem Aufbau müssen wir die Servos kalibrieren, so dass Ihre Startposition 90° beträgt. Ansonsten startet der Roboterarm in falschen Positionen.

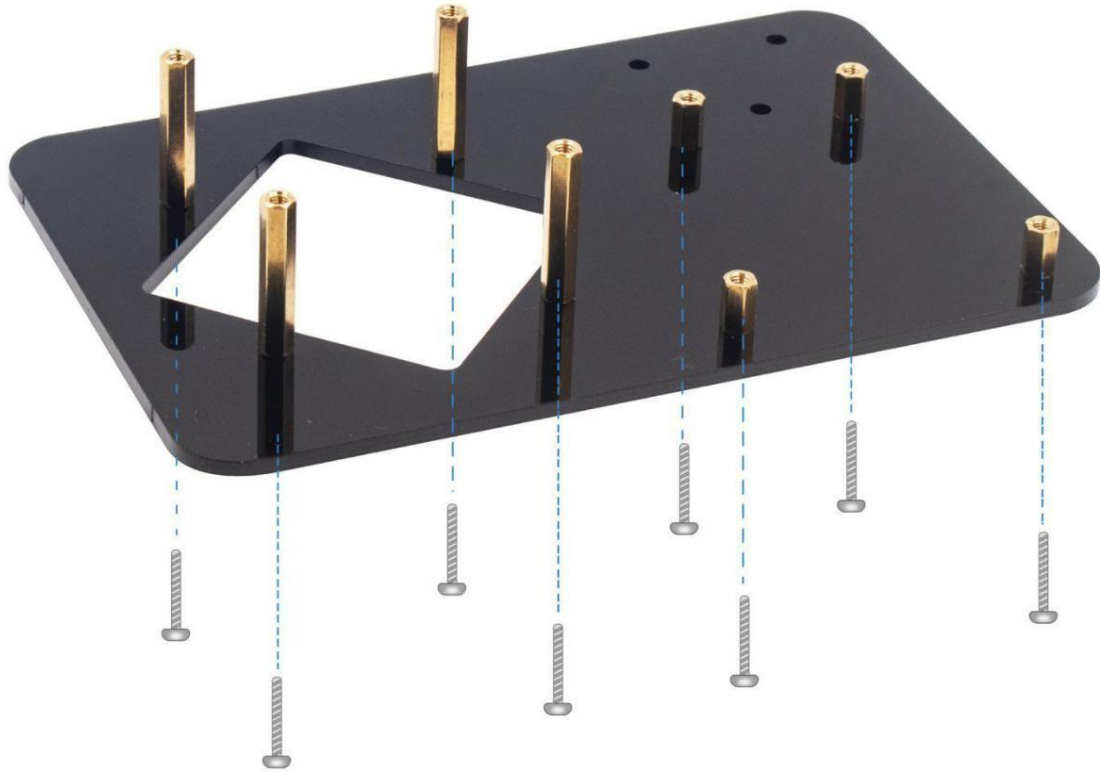
Stecken Sie zuerst das Sensor Shield auf das Arduino UNO Board und schließen Sie dann die Servomotoren an.



Verbinden Sie das Board mit Ihrem Computer und laden Sie den Install Sketch hoch. Die Motoren fahren dann ihre Position an. Nun können Sie das Board vom Computer trennen und mit dem Aufbau fortfahren.

Step 1

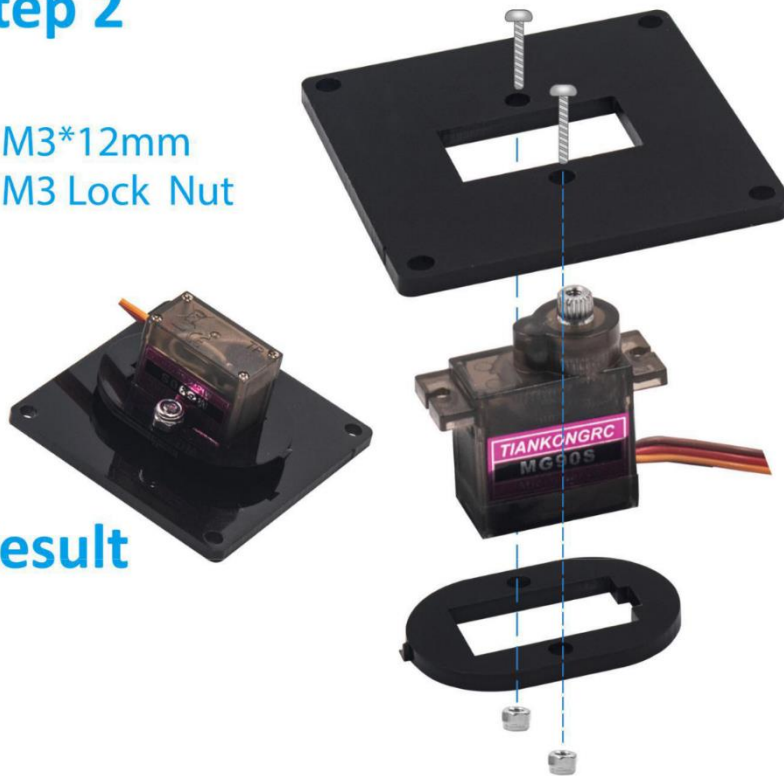
1. M3*6mm



Step 2

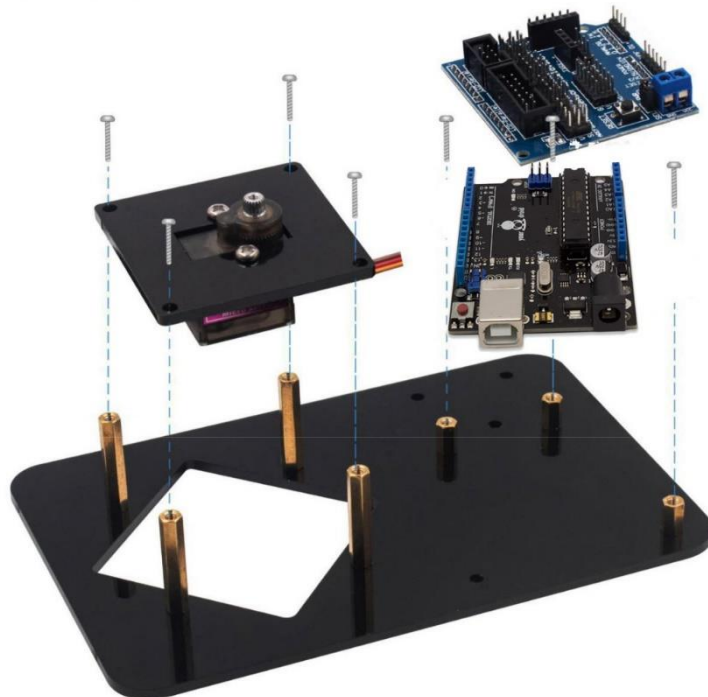
1. M3*12mm
2. M3 Lock Nut

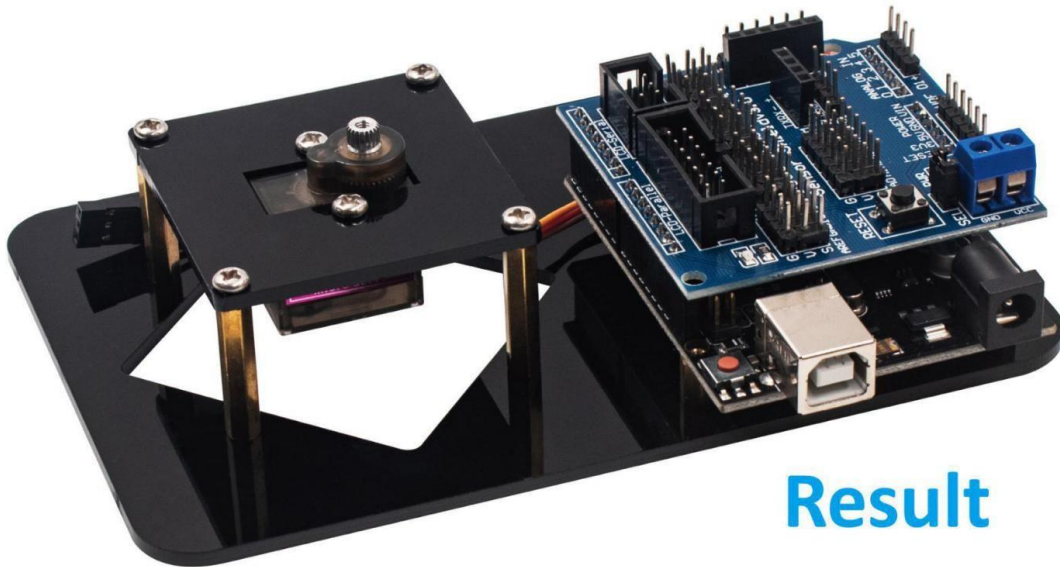
Result



Step 3

1. M3*6mm

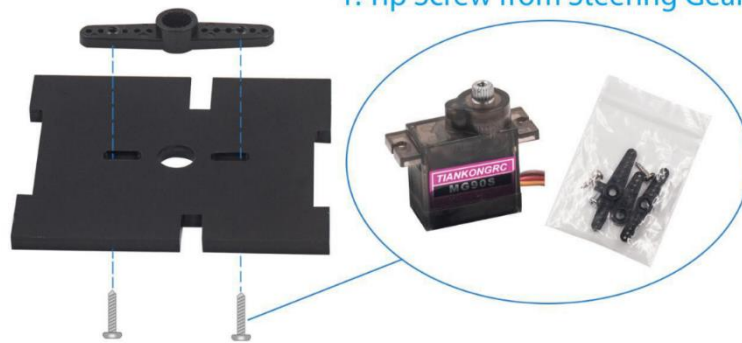




Result

Step 4

1. Tip Screw from Steering Gear

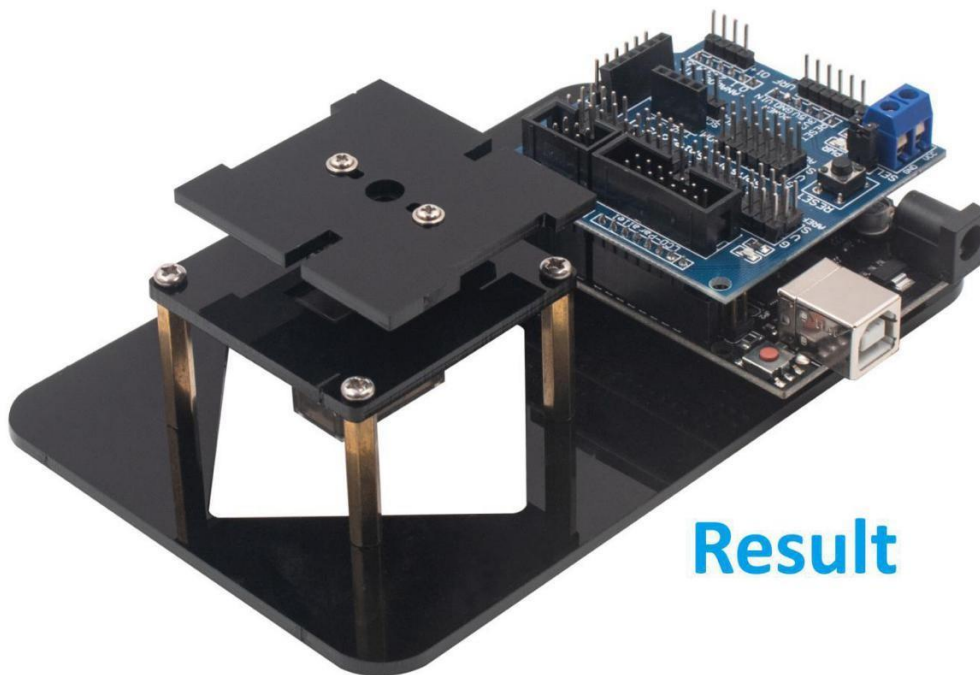


Result



Step 5

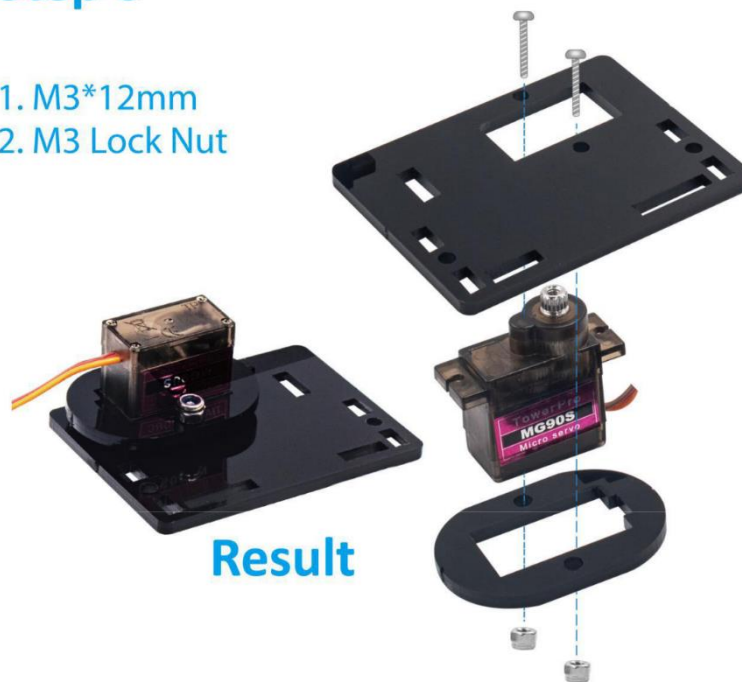
1. M2.5*5mm



Result

Step 6

1. M3*12mm
2. M3 Lock Nut



Step 7



1. Tip Screw from Steering Gear

Step 8

1. M2.5*5mm



Step 9

1. M3*12mm
2. M3 Lock Nut



Step 10



Result



Step 11

1. M2.5*5mm



Step 12

- 1. M3*10mm
- 2. M3 Lock Nut



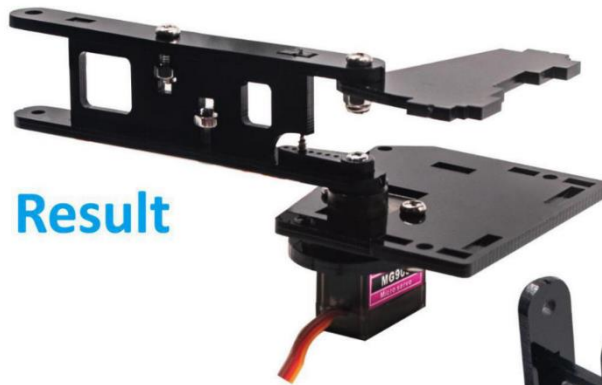
Result

Step 13

- 1. M3*10mm
- 2. M3 Nut



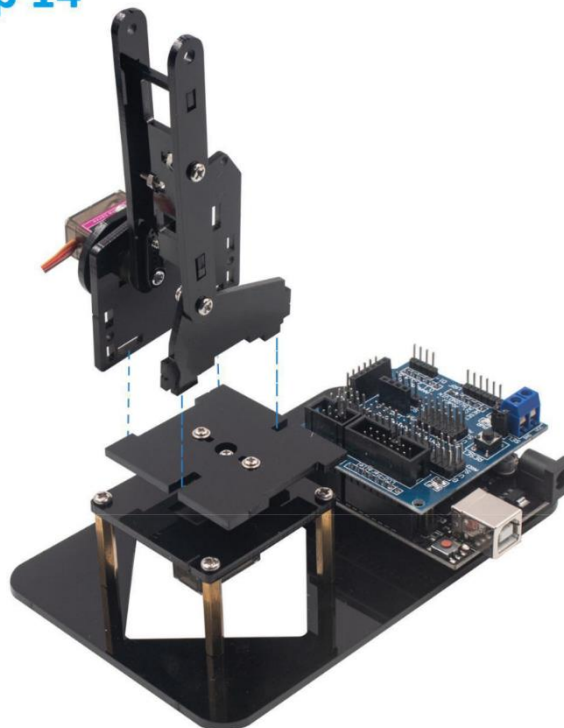
Result

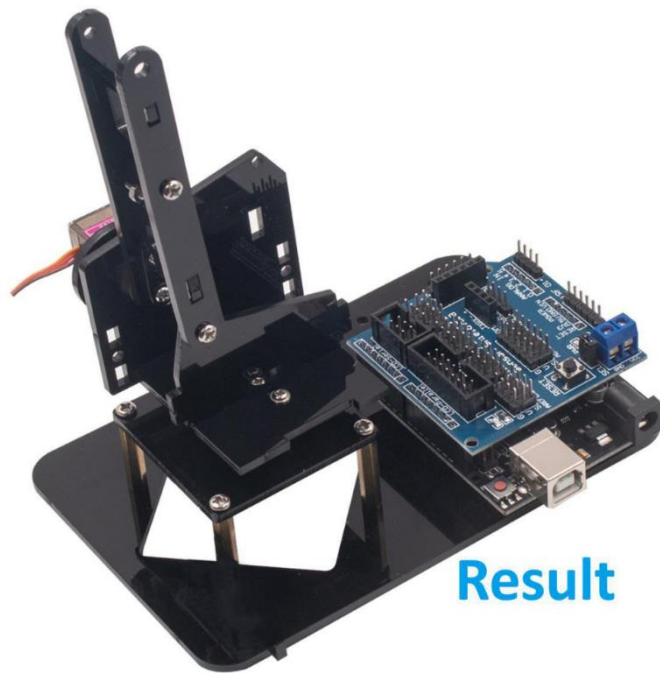
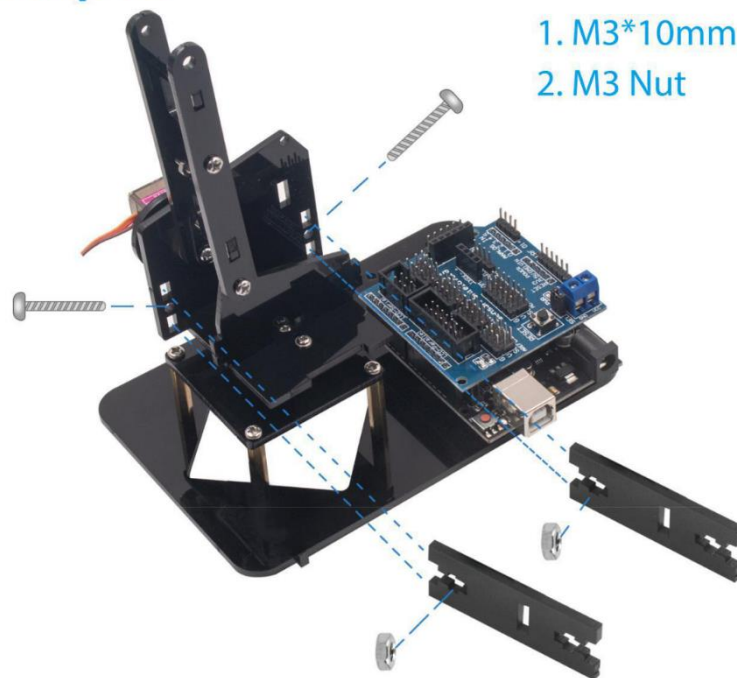


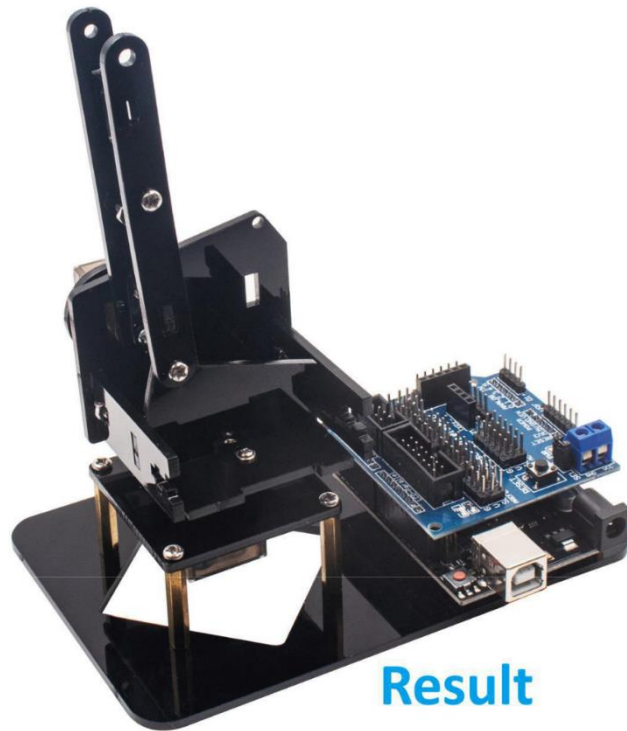
Result



Step 14



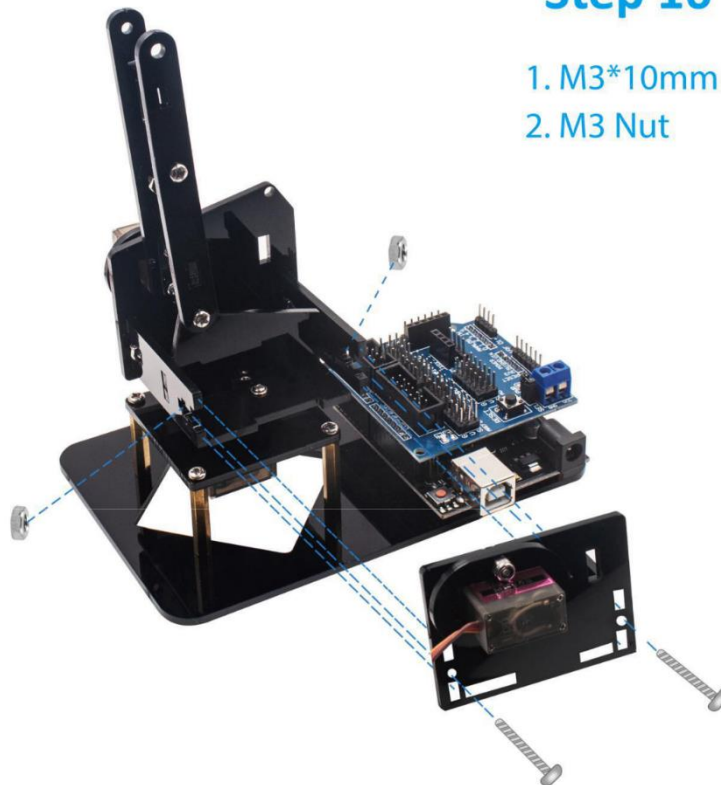
**Result****Step 15**

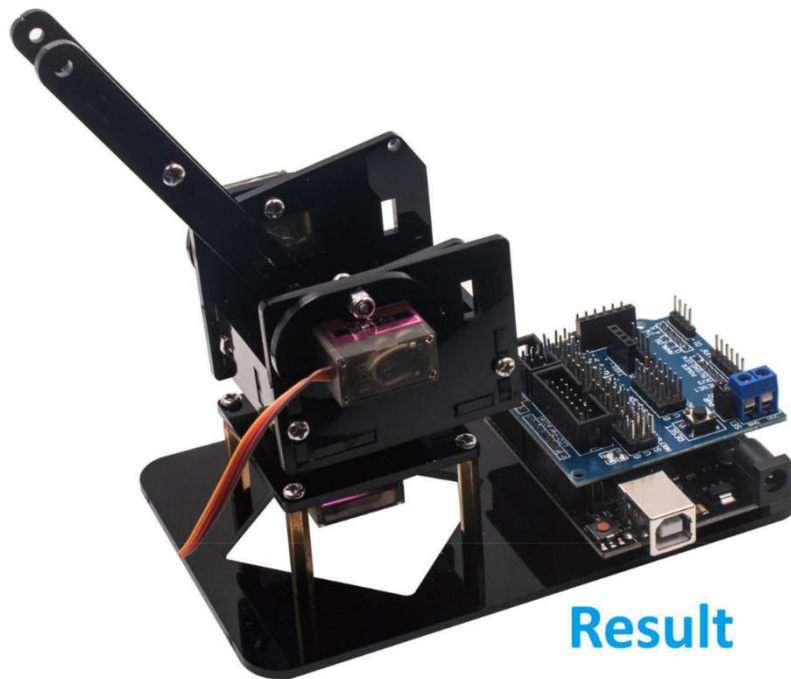


Result

Step 16

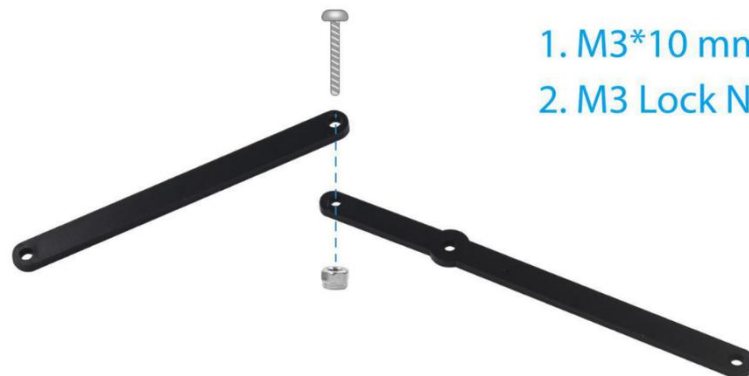
1. M3*10mm
2. M3 Nut





Result

Steps 17



1. M3*10 mm
2. M3 Lock Nut



Result

Steps 18

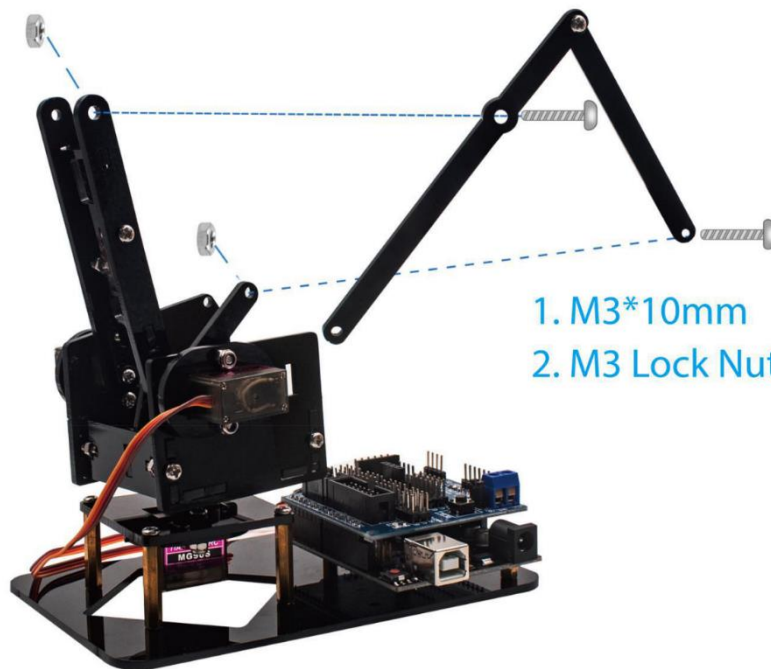
1. M3*12mm
2. M3 Lock Nut

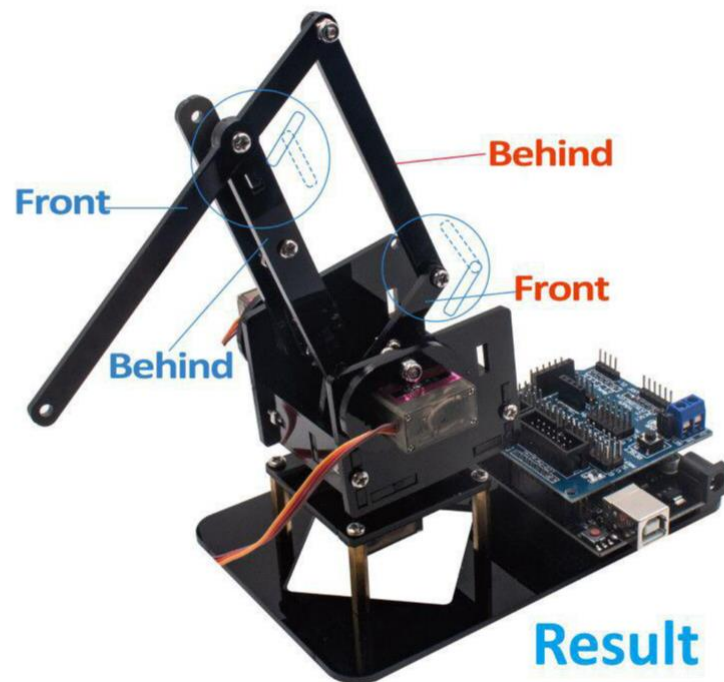


Result

Step 19

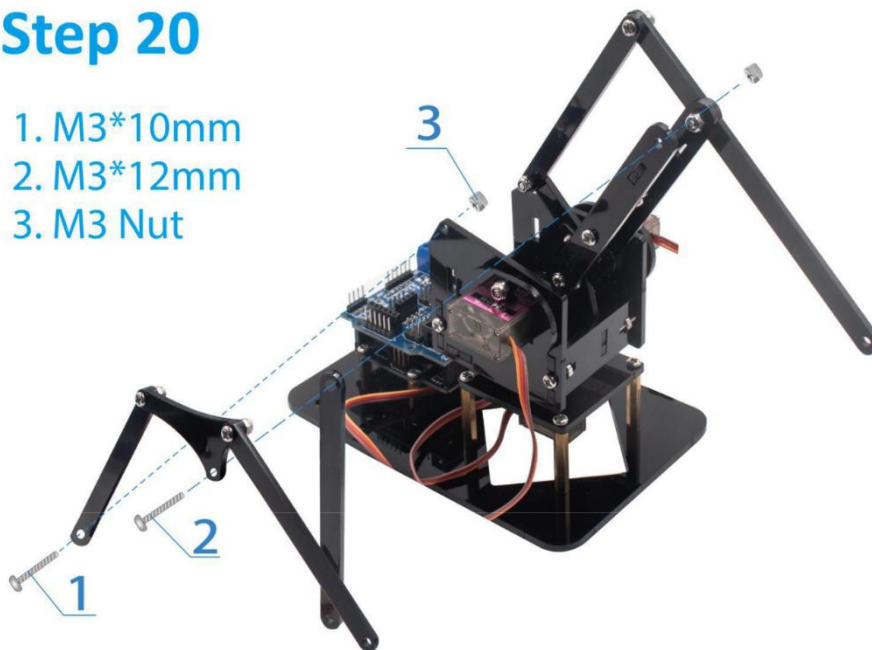
1. M3*10mm
2. M3 Lock Nut



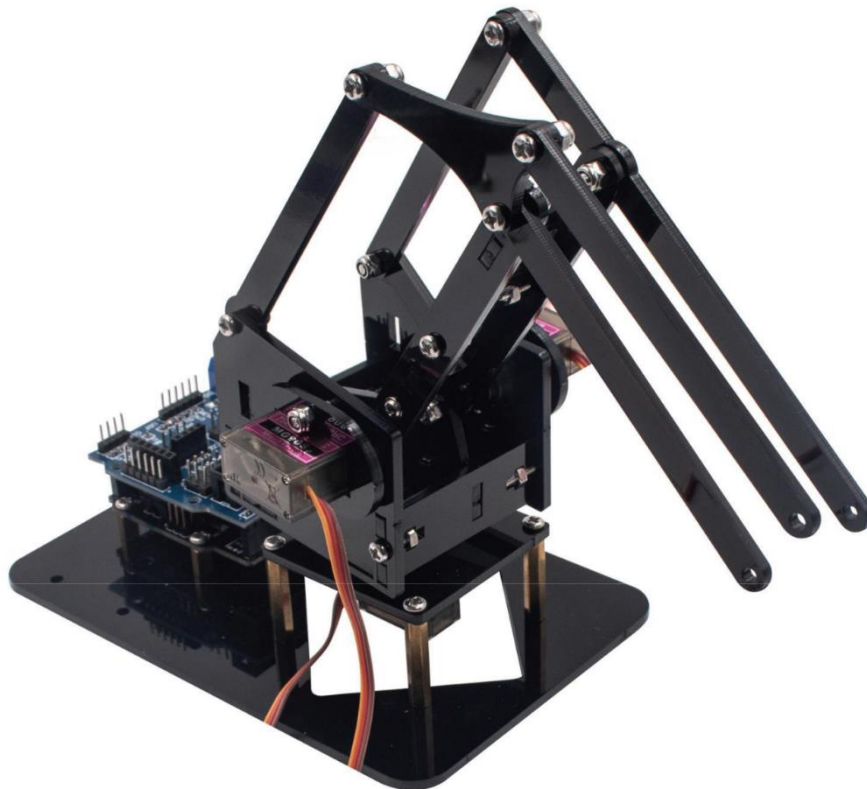


Step 20

- 1. M3*10mm
- 2. M3*12mm
- 3. M3 Nut



Result



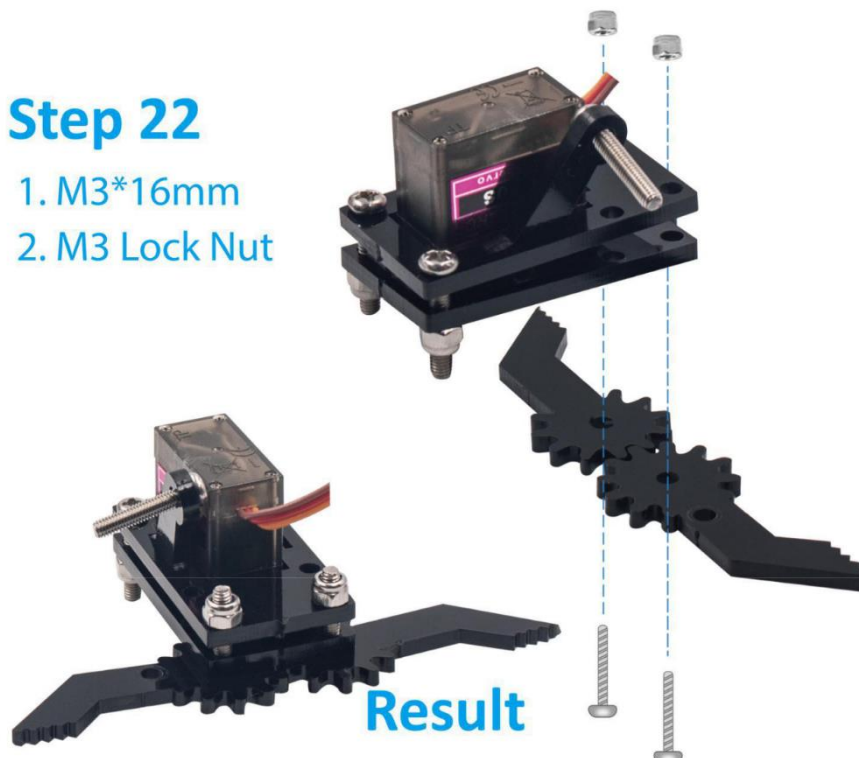
Step 21

- 1. M3*16mm
- 2. M3 Lock Nut



Step 22

- 1. M3*16mm
- 2. M3 Lock Nut

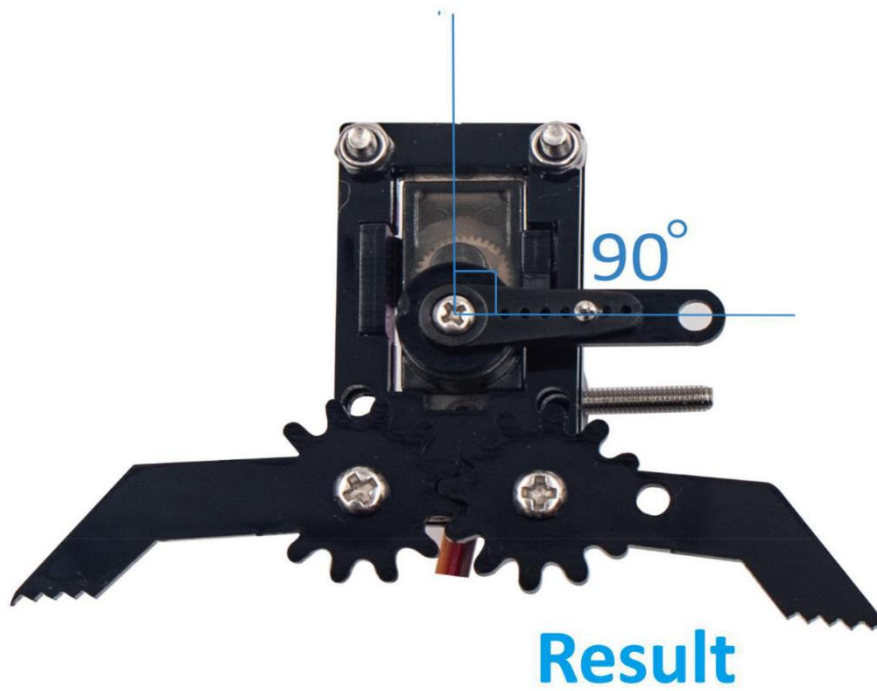


Step 23



Step 24

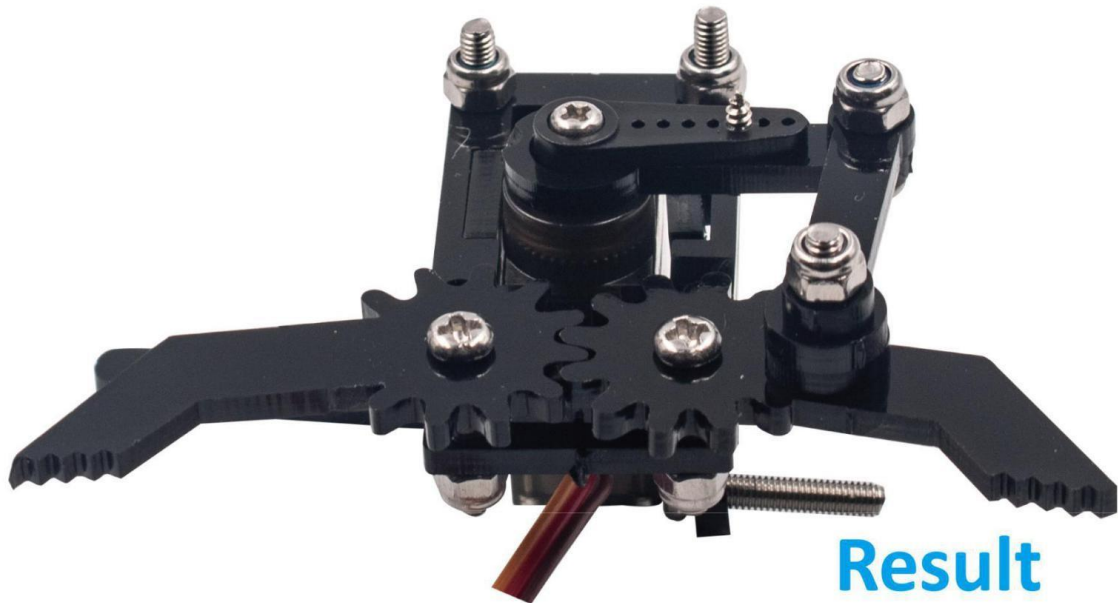




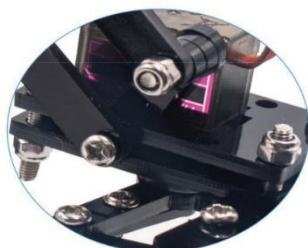
Steps 25

- 1. M3*10mm
- 2. M3*16mm
- 3. M3 Lock Nut
- 4. Gasket

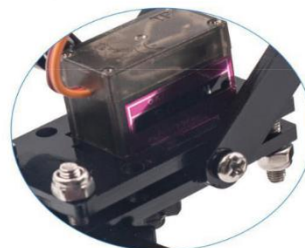




Step 26



Result



Result

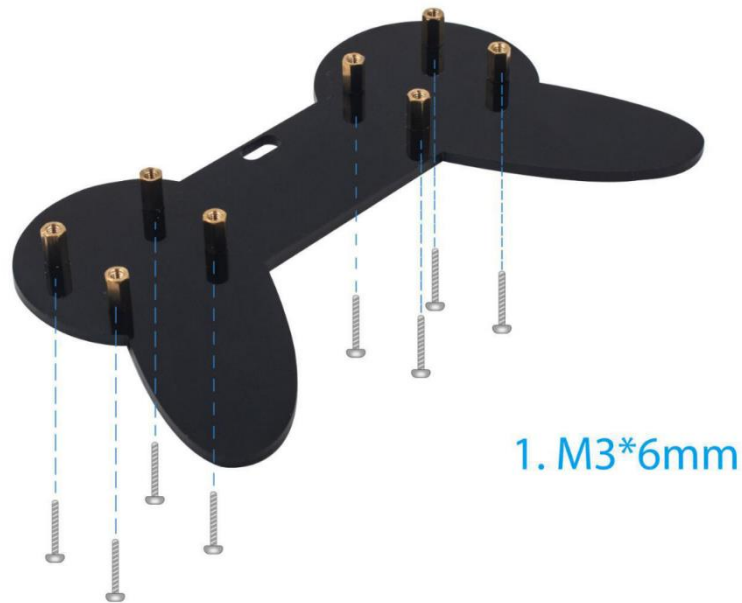


Result



Result

Step 27



Step 28





Result



5. Servo

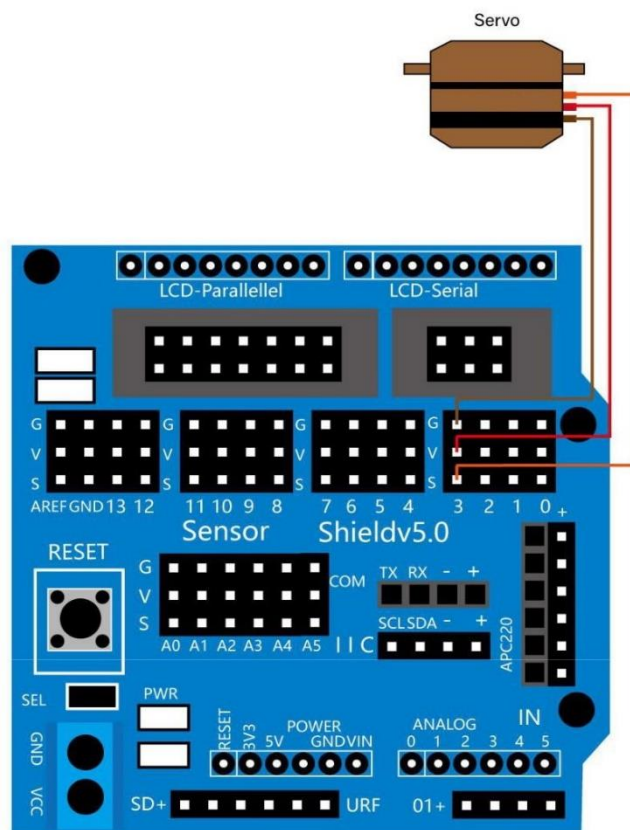
In diesem Kapitel zeigen wir anhand der Servo.h Bibliothek wie man einen Servomotor mit einem Arduino kompatiblen Board ansteuert. Servomotoren sind sehr vielseitig einsetzbar, sie sind beliebt in RC-Modellen, Robotern, Automaten und bringen Bewegung in sämtlich DIY-Elektronik Projekte.

Funktionsweise

Ein Servomotor (kurz Servo) ist ein Elektromotor, der sehr genau angesteuert werden kann. Neben der exakten Winkelposition kann auch die Drehgeschwindigkeit und Beschleunigung gesteuert werden. Der SG90 Micro Servo besteht aus einem kleinen DC-Motor, einem Getriebe und einer Steuerplatine mit Potentiometer.

Die Steuerplatine setzt die Signale in genaue Befehle um und über das Potentiometer wird die Position überprüft. Die üblichen Hobbyservos besitzen drei Pins zur Ansteuerung: GND, VCC und PWM. Gelegentlich findet man einen vierten Pin vor, dieser dient als Feedback-Pin, um die Position des Potentiometers über einen Microcontroller abzurufen.

Anschlussplan

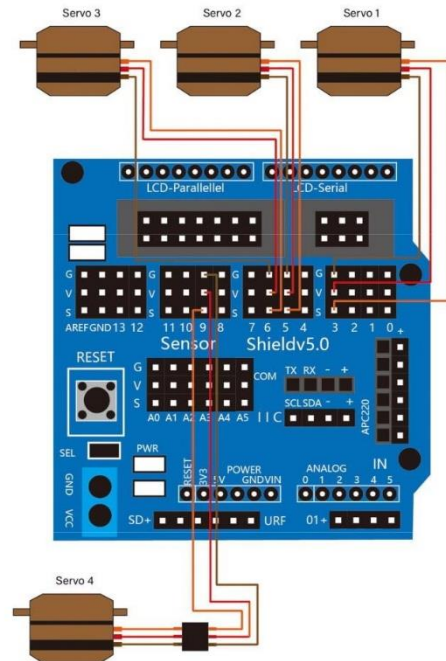
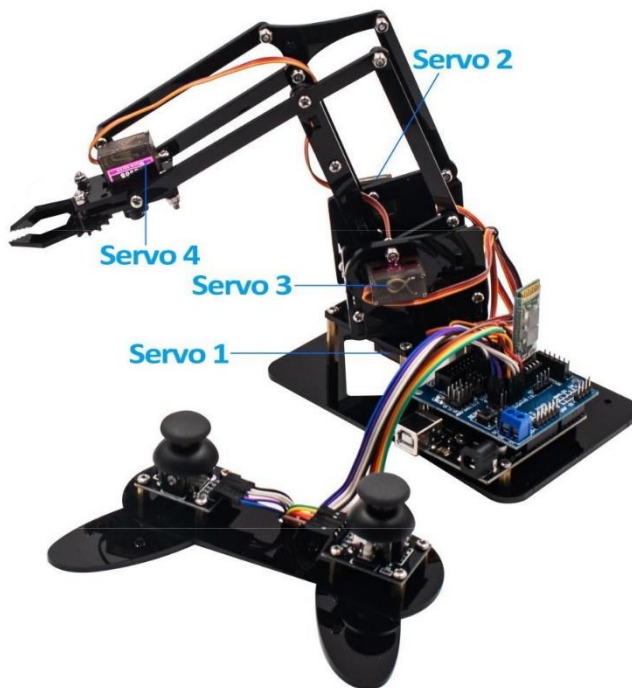


Servomotor	Arduino Sensor Shield
Gelb (PWM-Signalleitung)	Pin 3
Rot (VCC)	5V
Braun (GND)	GND

Code

Laden Sie den Code für Lektion 5 Servo auf den Arduino. Ohne die Servo.h Bibliothek funktioniert der Code nicht, wie Sie Bibliotheken installieren, können Sie in Lektion 2 nachlesen. Wenn Sie Probleme beim Hochladen haben, schlagen Sie in Lektion 3 nach.

Wenn die Steuerung mit einem Servo funktioniert, können Sie dazu über gehen die restlichen Servos anzuschließen.



Laden Sie den Code für Lektion 5 mehrere Servos auf den Arduino. Nach dem Einschalten, können Sie den Reset-Knopf drücken damit der Roboterarm eine kreisende Bewegung ausführt. Der Greifarm wird geschlossen und nach einer Sekunde wieder geöffnet. Der Arm dreht sich nach rechts und dann nach links. Anschließend wird der Arm ausgestreckt und nach einer Sekunde wieder eingezogen. Der Greifarm wird angehoben und nach einer Sekunde gesenkt.

6. Analoges Joystick Modul

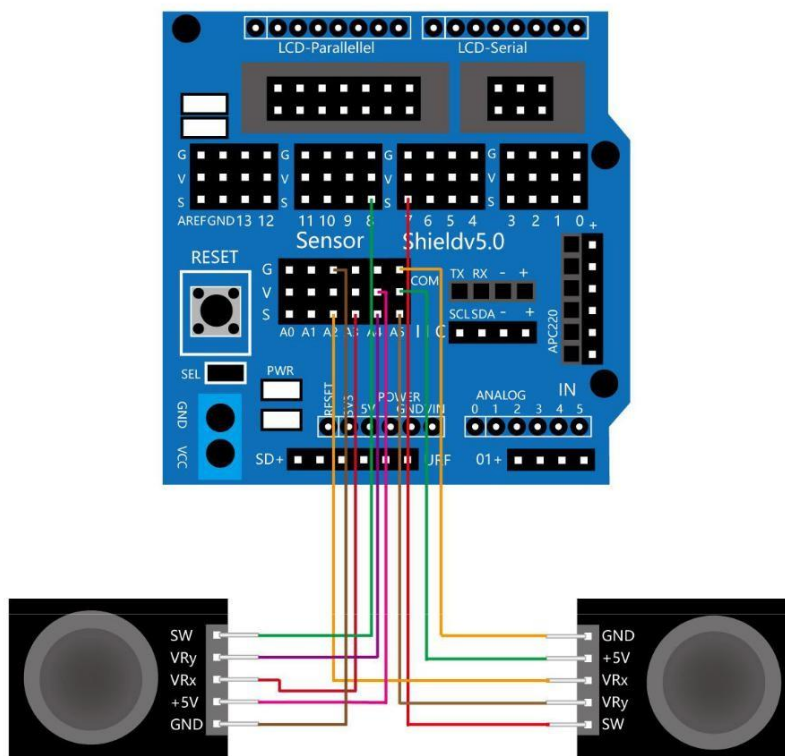
Mit Joysticks bringen Sie Ihre Projekte unter analoge Kontrolle! Dieses Joystick Modul eignet sich perfekt für eine 2-Achsen-Steuerung wie Sie häufig in Roboterarmen verwendet wird. Das Modul macht sich ebenfalls super als Eingabemethode für Spiele und Menüs.

Dieses Joystick Modul besitzt 5 Pins: GND, 5V, VRx, VRy und SW. Die Beschriftung kann je nach Charge variieren, die Belegung und Funktion bleibt aber die gleiche. Der Joystick arbeitet analog und liefert genauere Werte als digitale Joysticks. Er verfügt über einen integrierten Taster, der durch Drücken ausgelöst werden kann. Dadurch lassen sich z.B. Umschaltfunktionen (Toggle) oder Auswahlfunktionen (Select) programmieren.

Funktionsweise

Am Arduino werden die Daten der X- und Y-Pins über analoge Eingänge gelesen. Das Joystick Modul besitzt jeweils ein Potentiometer für die X- und Y-Achse. Je nach Position des Joysticks wird eine andere Ausgangsspannung an unseren Arduino gegeben. Das Arduino Board wandelt die analoge Spannung zwischen 0 bis 5V in Werte von 0 bis 1023 um. Der Taster liefert permanent 2V, bei Betätigung wird er mit GND verbunden. Im seriellen Monitor lesen wir 1 im Ruhezustand und 0 beim Auslösen.

Anschlussplan



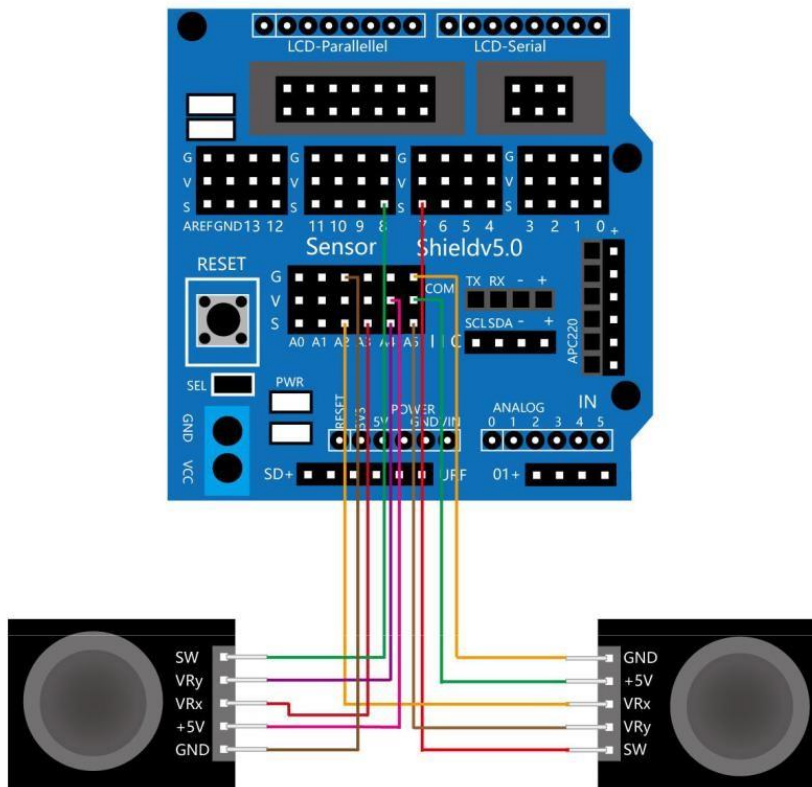
Joystick Test

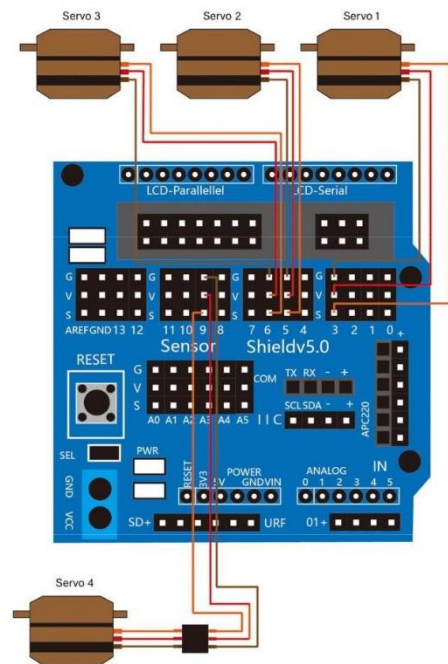
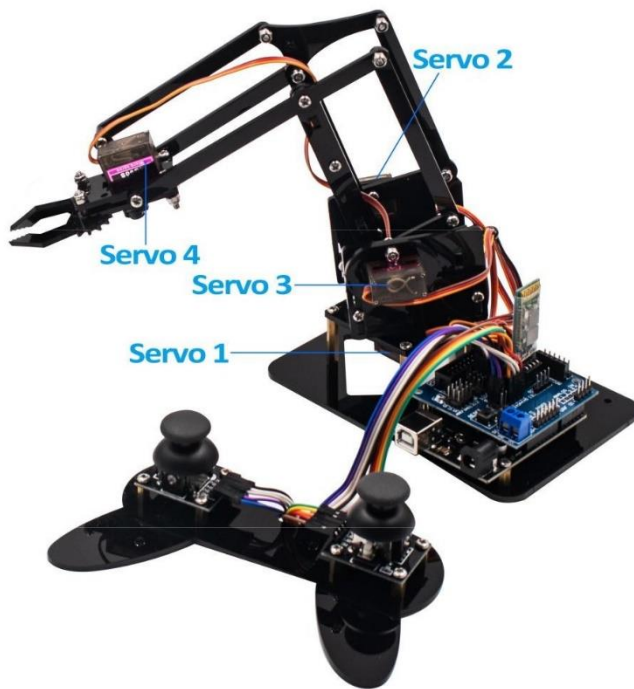
Nach dem Verkabeln können Sie den Sketch aus Lektion 6 Analog_Joystick_Module hochladen. Öffnen Sie den seriellen Monitor und lesen Sie die aktuelle Position des Joysticks aus.

7. Roboterarm mit zwei Joysticks steuern

In diesem Kapitel steuern wir die vier Servos des Roboterarms mit den beiden Joysticks. Mit den beiden Joysticks werden die vier verschiedenen Achsen gesteuert.

Anschlussplan





Anweisungen

Linker Joystick	Arm	Rechter Joystick	Arm
X1 < 50	Linken Joystick nach rechts bewegen, um den Greifer zu öffnen	X2 < 50	Rechten Joystick nach links bewegen, um den Arm nach links zu rotieren
X1 > 1000	Linken Joystick nach Links bewegen, um den Greifer zu schließen	X2 > 1000	Rechten Joystick nach rechts bewegen, um den Arm nach rechts zu rotieren
Y1 < 50	Linken Joystick nach unten bewegen, um den oberen Arm zu senken	Y2 > 1000	Rechten Joystick nach unten bewegen, um den Arm einzufahren
Y1 > 1000	Linken Joystick nach oben bewegen, um den oberen Arm zu heben	Y2 < 50	Rechten Joystick nach oben bewegen, um den Arm auszufahren

8. Roboterarm mit Bluetooth steuern

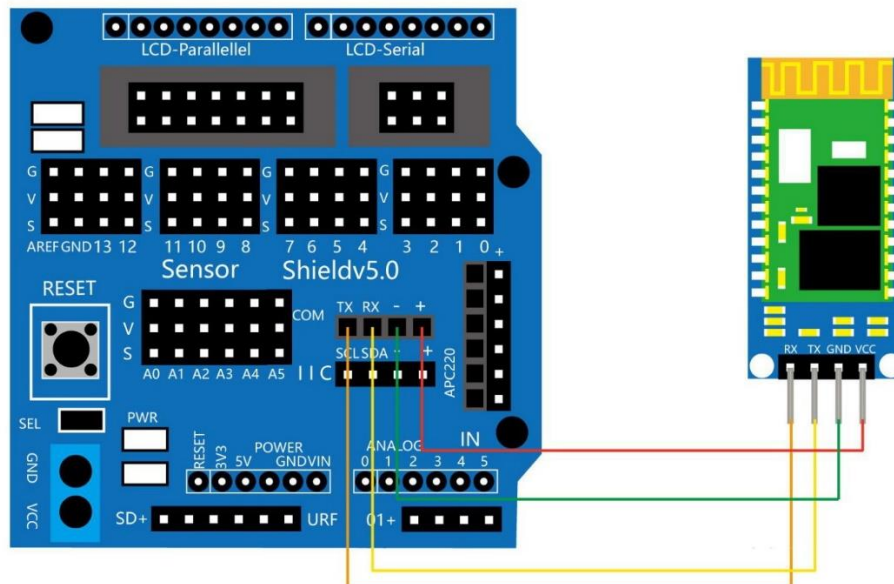
Dieses Kapitel ist in zwei Teilen unterteilt, zuerst erlernen Sie die Kommunikation mit dem Bluetooth Modul und dann lernen wie Sie mit der App den Roboterarm steuern können.

1. Bluetooth-Test

Das HC06 ist ein Bluetooth-Modul mit seriellem Anschluss und Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation mit komplettem 2.4GHz Funk-Transceiver und Basisband. Es verwendet ein CSR Blue core 04-External single chip Bluetooth System mit CMOS Technologie und AFH (Adaptive Frequency Hopping Feature).

Anschluss

HC-06 Bluetooth Modul	Arduino Sensor Shield
VCC	5V
GND	GND
TXD	RX
RXD	TX



Code

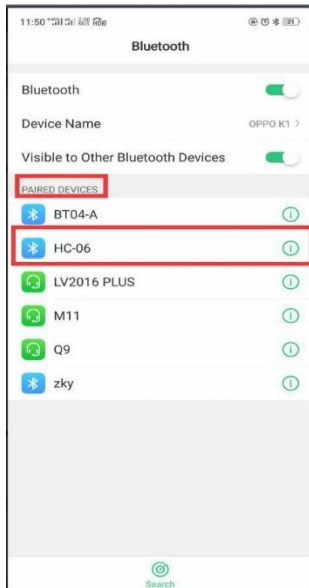
Öffnen Sie das Programm aus Lektion 8 Bluetooth Test und laden Sie es auf Ihren Arduino.

ACHTUNG: Das Bluetooth-Modul muss vor jedem Upload abgezogen werden, sonst schlägt das Hochladen des Programms fehl. Erst nachdem das Programm erfolgreich hochgeladen wurde, können Sie das Bluetooth-Modul wieder anschließen!

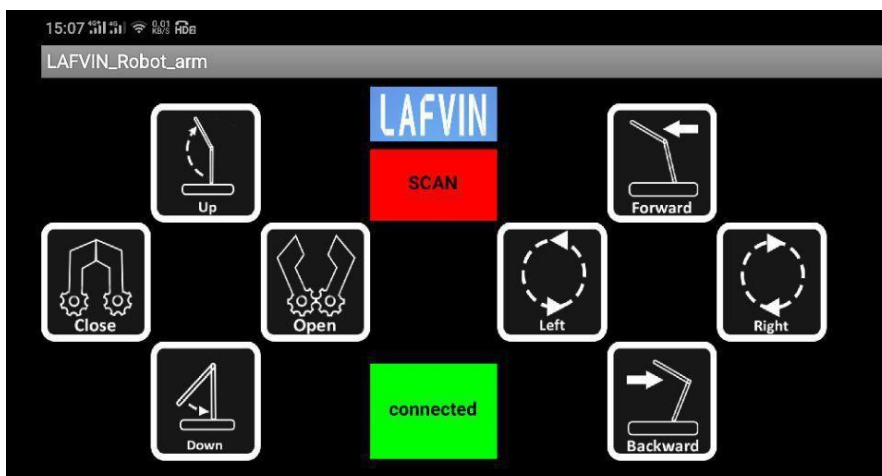
Einrichtung der App

Zuerst laden Sie die APK Datei aus Lektion 8 auf Ihr Smartphone und installieren die App.

Stellen Sie dann sicher, dass das Bluetooth-Modul angeschlossen ist. Koppeln Sie Ihr Telefon mit dem HC-06. Gehen Sie dazu auf Einstellungen->Bluetooth->Gerät scannen->HC-06 auswählen und koppeln Sie es. Das Passwort ist „1234“. Wenn Ihr Telefon mit dem Bluetooth-Modul verbunden ist, sehen Sie ein brauchbares Gerät mit dem Namen HC-06 unter PAIRED DEVICES (wie unten gezeigt).



Wenn das HC-06 nicht auftaucht, wiederholen Sie die Anweisungen. Sobald es geklappt hat, können Sie die App öffnen.



Klicken Sie auf SCAN und wählen Sie den HC-06 aus.

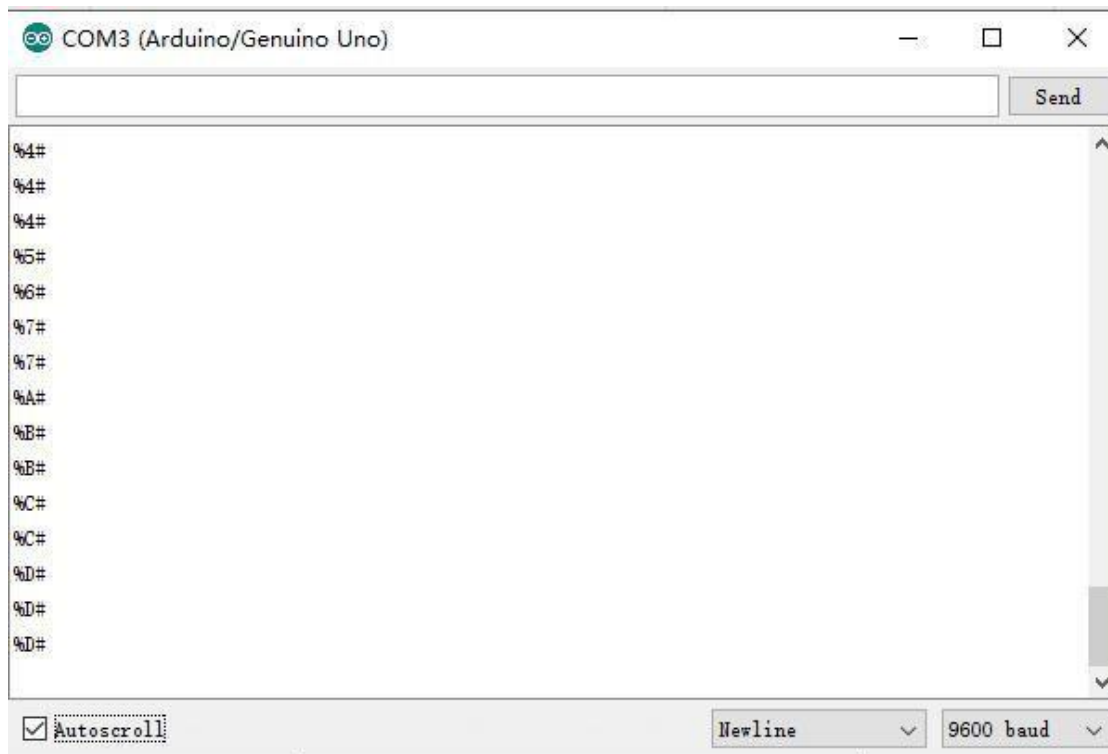










Nach der Auswahl drücken Sie auf den grünen Button und warten Sie 5 Sekunden. Wenn die grüne Schaltfläche „Connected“ anzeigt, war das Verbinden erfolgreich.

Hinweis: Wenn das Bluetooth-Modul nicht erfolgreich verbunden ist, blinkt die rote LED weiter. Wenn die Verbindung erfolgreich ist, leuchtet die rote LED.

Roboterarm mit Bluetooth steuern

Verbinden Sie das Arduino Board mit Ihrem Computer und öffnen Sie den seriellen Monitor. Stellen Sie die Baudrate auf 9600 und drücken Sie ein paar Knöpfe auf der App. Im seriellen Monitor müssten jetzt die entsprechenden Codes erscheinen.

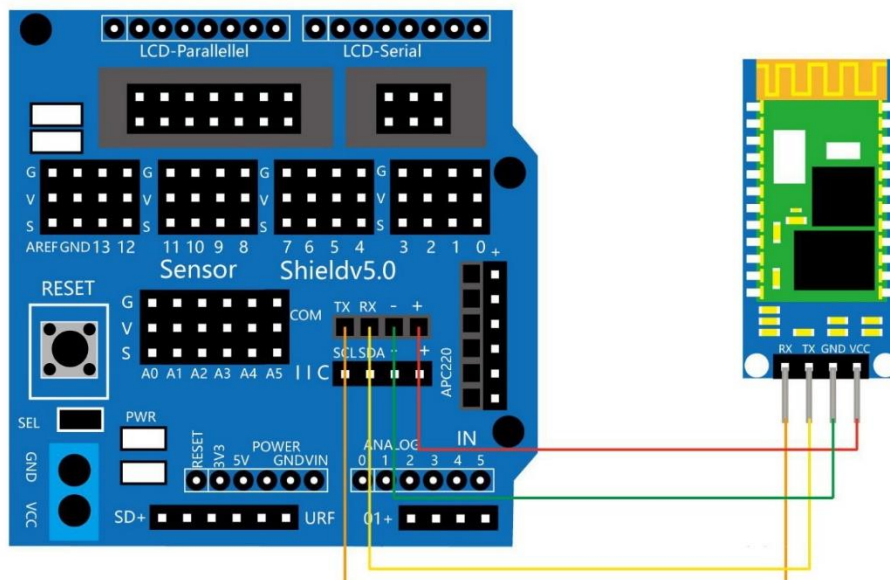


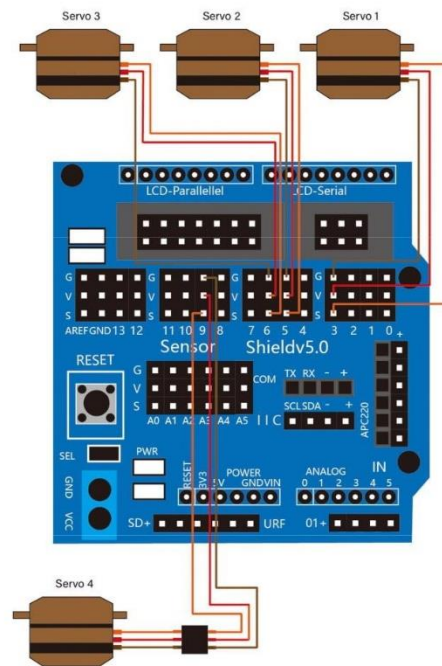
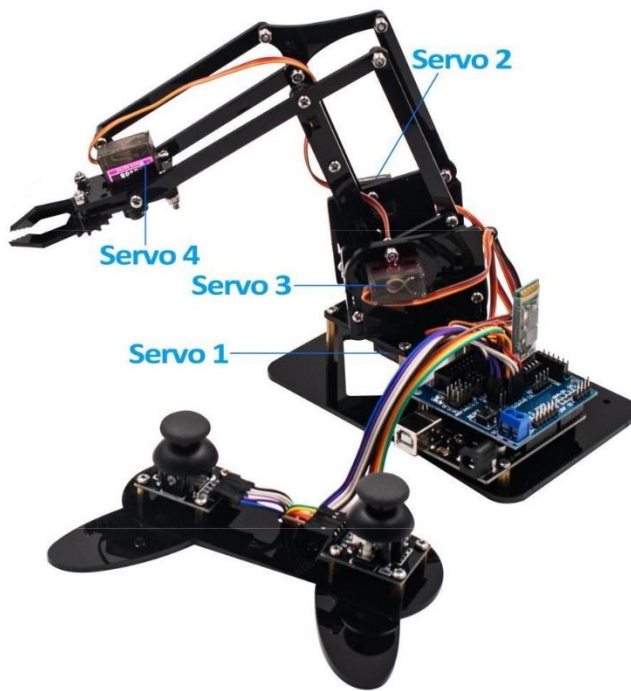
APP KEY	value
 Up	%4#
 Close	%5#
 Close	%6#
 Down	%7#
 Forward	%A#
 Left	%B#
 Right	%C#
 Backward	%D#

2. Steuerung über Bluetooth

Nachdem wir Ihnen gezeigt haben, wie man Befehle an das Bluetooth-Modul sendet, lernen Sie nun wie Sie den Roboterarm mit der App über Bluetooth steuern.

Anschlussplan













Öffnen Sie das Programm aus Lektion 8 Bluetooth Steuerung und laden Sie es auf Ihren Arduino.

ACHTUNG: Das Bluetooth-Modul muss vor jedem Upload abgezogen werden, sonst schlägt das Hochladen des Programms fehl. Erst nachdem das Programm erfolgreich hochgeladen wurde, können Sie das Bluetooth-Modul wieder anschließen!

Anschließend können Sie anfangen, den Roboterarm mit der App zu steuern.

Schaltfläche	Aktion
	Arm heben
	Greifer schließen
	Greifer öffnen
	Arm senken
	Arm ausstrecken
	Drehung links
	Drehung rechts
	Arm zurückziehen