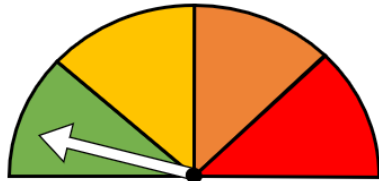


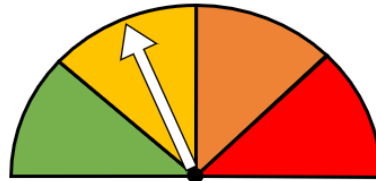


# Arduino Uno

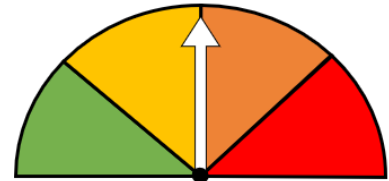
## Station 6b | Farbsensor und Servomotor



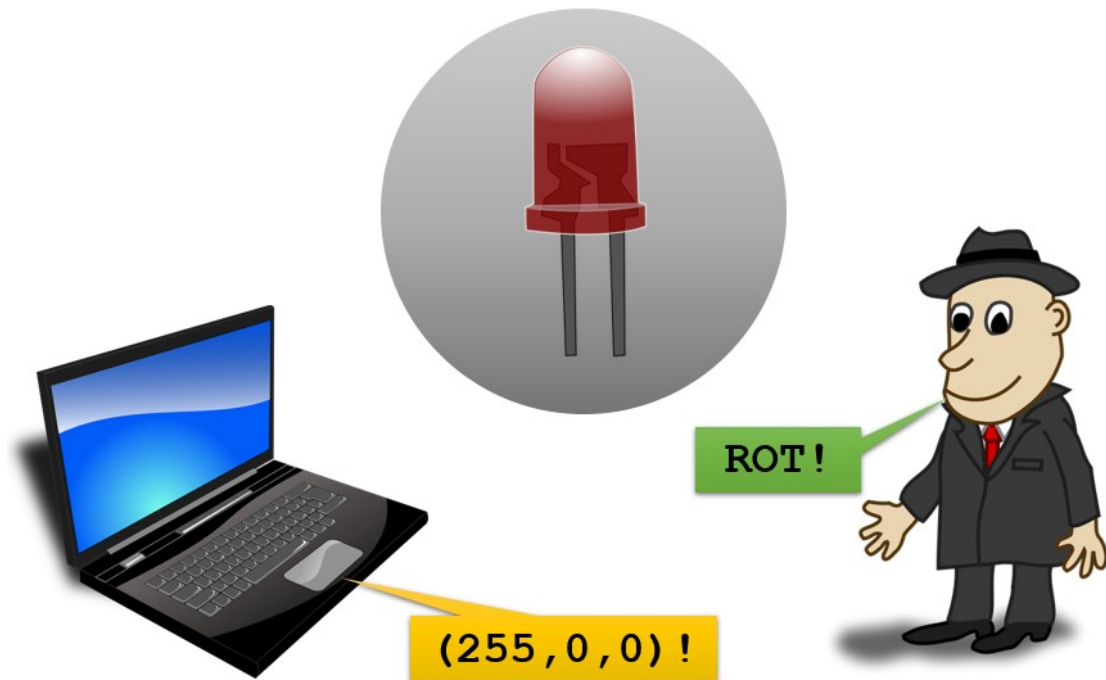
algorithmisches Denken



Programmieraufwand



Komplexität der Schaltung



## ZIEL DER STATION

Die Darstellung der eingescannten Farben soll nun nicht mehr über den seriellen Monitor ablaufen, sondern haptisch mit Hilfe eines **Servomotors** und einer dazugehörigen Schablone. Ziel soll es sein, dass der Motor einen Pfeil ansteuert und auf das richtige Segment auf der Schablone zeigt. Dazu lernst du nachfolgend den Servomotor kennen und wirst dich mit der Ansteuerung von diesem beschäftigen.

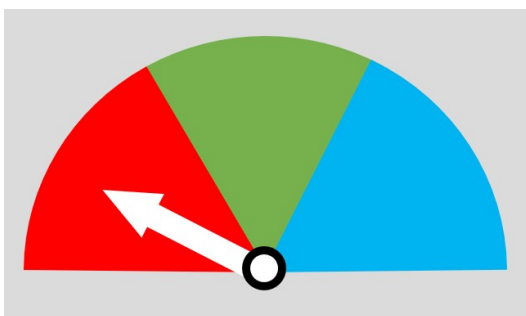


Abbildung 2: Schablone



Abbildung 1: Servomotor

## DER SERVOMOTOR

Der Servomotor besteht im Inneren aus einem Elektromotor, einem Getriebe und einem Potentiometer zur Bestimmung der Position. Angeschlossen wird dieser mit Hilfe von drei Kabel. Dabei handelt es sich um zwei Kabel zur Stromversorgung (+5V und GND), sowie einer Datenleitung. Der Aufsatz ist wechselbar und kann Positionen zwischen 0° und 180° ansteuern.

## STEUERUNG DES SERVOMOTORS

Als Grundlage zur Steuerung nutzen wir eine Programmbibliothek.

### **i** WAS IST EINE PROGRAMMBIBLIOTHEK?

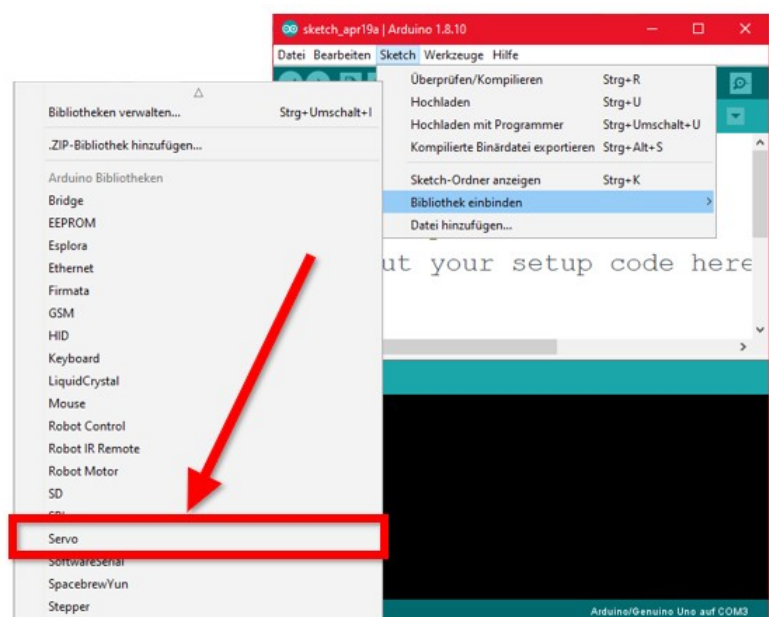
Bibliotheken sind quasi nichts anderes als eine Sammlung von Funktionen. Diese sind in der Bibliothek implementiert und können nach der Einbindung im Programmcode einfach aufgerufen werden.

Die Bibliothek die wir hier verwenden wollen heißt „Servo“ und wird in der Arduino Programmierumgebung bereits mitgeliefert. Bevor wir uns mit der Einbindung der Bibliothek beschäftigen, schauen wir uns erstmal an, welche Befehle die Programm-Bibliothek mitbringt.

Befehl	Bedeutung
<code>servo.attach();</code>	Dieser Befehl legt die Datenleitung fest, dass heißt in den Klammern muss als Wert der digitale Pin hineingeschrieben werden.  <b>Beispiel:</b> <code>servo.attach(5);</code>
<code>servo.write();</code>	Damit wird der bewegliche Aufsatz an eine Position bewegt. Der Wert kann sich im Wertebereich von 0° bis 180° bewegen. Die Ansteuerung benötigt allerdings etwas Zeit, daher wird nach dem Befehl das Festlegen einer Pause empfohlen.  <b>Beispiel:</b> <code>servo.write(60);</code>
<code>servo.read();</code>	Gibt die zuletzt mit <code>write()</code> -Befehl festgelegte Position zurück.  <b>Beispiel:</b> <code>position = servo.read();</code>
... und weitere ...	

Jetzt müssen wir uns nur noch anschauen, wie wir die Bibliothek nutzen können. Dafür navigierst du dich über „Sketch“ zu „Bibliothek einbinden“ und wählst dann in der angezeigten Liste die Bibliothek „Servo“ aus.

Die rechtsstehende Grafik zeigt dir nochmal wo genau du die benötigte Bibliothek findest.



Mit Anklicken der Bibliothek wird im Programmcode automatisch eine neue Zeile eingefügt.

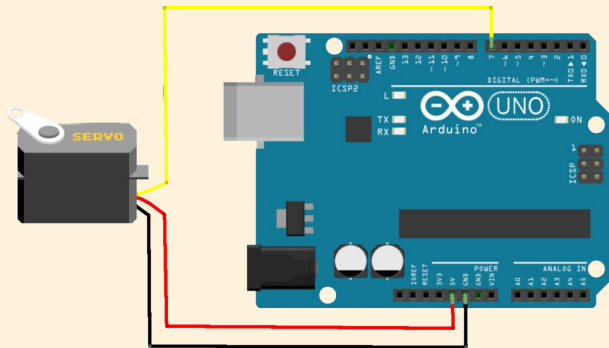
```
#include <Servo.h>
```

Diese zeigt dir an, dass dein Programm nun diese Bibliothek eingebunden hat und du diese Befehle nutzen kannst.



## AUFGABE 1 - SERVOMOTOR VERBINDEN

Verbinde nun den Servomotor, zusätzlich zum Farbsensor, mit dem Arduino. Zur Orientierung kannst du die rechtsstehende Grafik nutzen. Füge weiterhin deinem Programmcode die Bibliothek für den Servomotor hinzu.



fritzing

## STEUERN DES MOTORS

Nun möchten wir die Positionen ansteuern. Insgesamt haben wir drei Bereiche, welche eine Halbkugel abdecken, sprich 180°. Pro Segment stehen uns als 60° zur Verfügung und den Zeiger möchten wir immer mittig positionieren. Somit ergeben sich folgende Gradzahlen:

rot - 30°

grün - 90°

blau - 150°

Die Bibliothek haben wir ja bereits erfolgreich eingebunden. Im nächsten Schritt müssen wir uns ein sogenanntes **Objekt** instanzieren. Dieses ist unser Servomotor und der dazugehörige Code sieht wie folgt aus:

```
Servo servo;
```

In Worten bedeutet dies nichts anderes, als das wir ein Servo-Objekt mit dem Namen „servo“ erzeugen. Dann müssen wir weiterhin die Verbindung mit Motor herstellen, dies tun wir innerhalb der `setup()`-Methode mit dem Befehl:

```
servo.attach(PIN);
```

Die Punktschreibweise heißt, dass auf das Objekt „servo“ die `attach()`-Funktion angewandt wird. Nun können wir innerhalb der `loop()`-Methode mit dem Befehl:

```
servo.write(Gradzahl);
```

die jeweiligen Positionen anfahren.

## FARBEN MIT DEM ZEIGER ANZEIGEN

Nun wollen wir die Anzeige des gescannten Farbwertes realisieren. Dazu kannst du, innerhalb der jeweiligen Verzweigungen, die `write()`-Funktion auf unser Servo-Objekt anwenden. Ich empfehle dir hierbei die Anzeige im seriellen Monitor nicht zu löschen, damit du die Möglichkeit hast die Ausgabe zu kontrollieren.

```
if (rot < blau && rot < gelb){
    Serial.println("Farbe rot erkannt!");
    servo.write(30);
    delay(1000);
} else if (Bedingung für grüne Farbe){
    ...
} else if (Bedingung für blaue Farbe){
    ...
} else {
    Serial.println("Keine Farbe erkannt! :( ");
}
```



## AUFGABE 2 – DARSTELLUNG MIT DER RGB-LED

Ergänze nun jeweils die Fallunterscheidungen mit dem korrekten Befehlen zur Steuerung des Servomotors, für die jeweilige Farbe.

Nutze weiterhin die zur Verfügung gestellte Schablone und positioniere den Servomotor so, dass links unten, am unteren Rand des rechten Segments, die Position  $0^\circ$  eingestellt ist. Teste deinen Aufbau im Anschluss mit den zur Verfügung gestellten farbigen Materialien.



-----  
Grafik auf dem Deckblatt: Rote LED-Lampe aus, Openclipart, CC0, <https://creativecommons.org/publicdomain/zero/1.0/>,  
<https://publicdomainvectors.org/de/kostenlose-vektorgrafiken/Rote-LED-Lampe-aus/61716.html>

und

Kaufmann-comic-Figur-Vektor-Bild, CC0, <https://creativecommons.org/publicdomain/zero/1.0/>,  
<https://publicdomainvectors.org/de/kostenlose-vektorgrafiken/Kaufmann-comic-Figur-Vektor-Bild/6840.html>

Screenshots: fritzing electronics made by easy und Arduino IDE 1.8.12 (windows)

Alle weiteren Grafiken: Patrick Binkert, EduInf@TUD