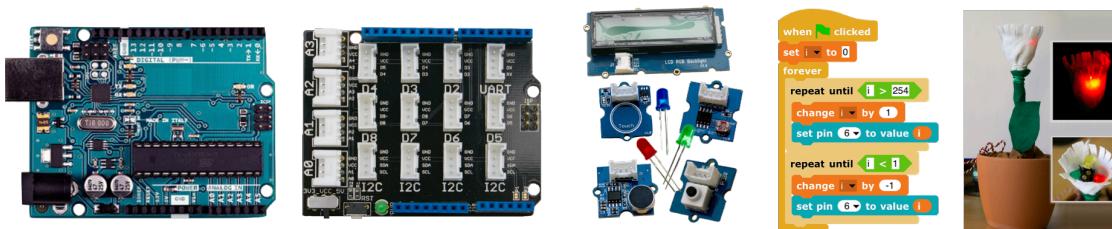




ARDUINO, GROVE UND SNAP4ARDUINO



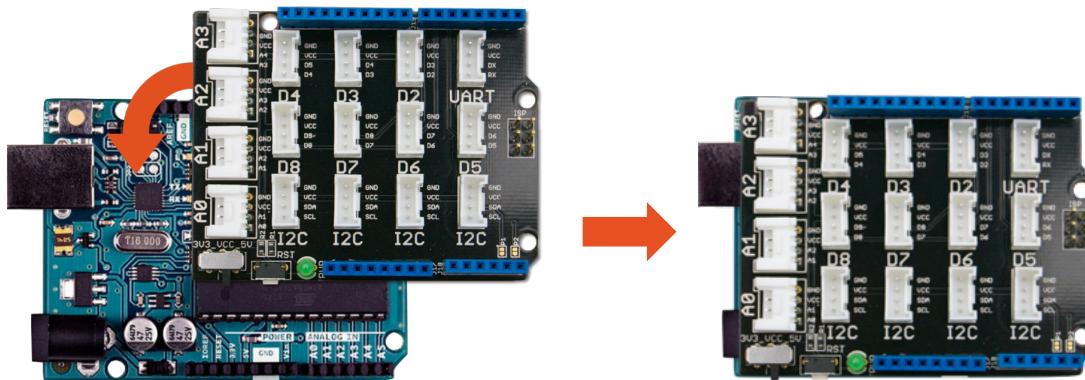
Anleitung



Ansteuerung des Arduino Uno	3
Die Programmierumgebung Snap4Arduino	4
Projekte erstellen, öffnen und laden.....	4
Programme zur Arduino-Ansteuerung erstellen	5
Die Arduino-Blöcke	5
Sensoren.....	5
Taster und digitale Sensoren (Binärwert).....	6
Analoge Sensoren (wertdiskret)	7
Aktoren.....	7
LEDs	8
Piezo-Summer	9
Servomotoren.....	10
Displays	11

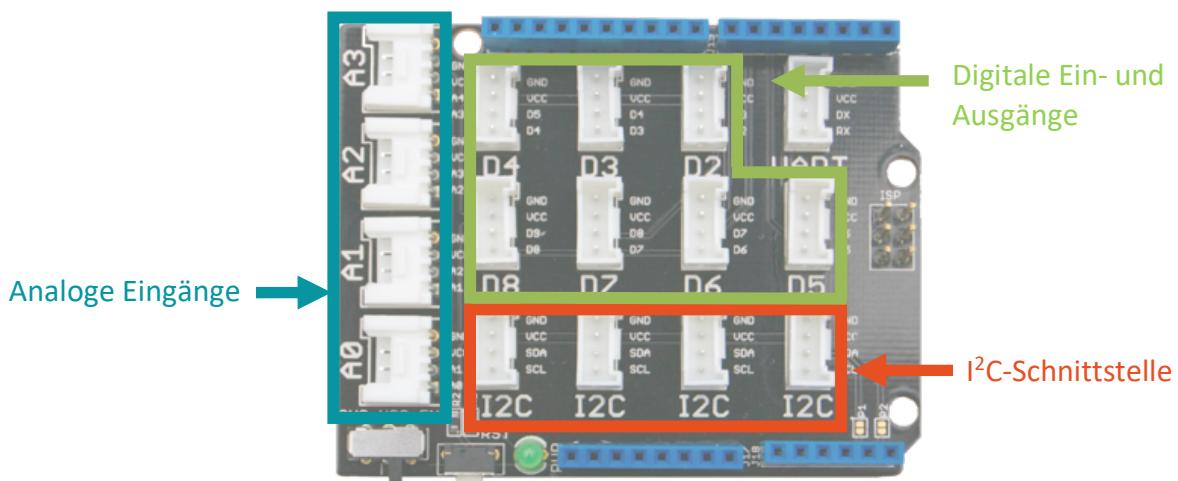


Um die Arbeit mit den elektronischen Bauteilen zu erleichtern, wird auf den Arduino Uno (oder ein kompatibles Board) das Grove-Shield aufgesteckt:



Das Grove-Shield enthält verschiedene Bereiche, an die Sensoren, Aktoren und Displays angeschlossen werden können:

- Digitale Ein- und Ausgänge (*Digital Input/Output D2, D3, ..., D8*), davon PWM¹: D3, D5, D6
- Analoge Eingänge (*Analog Input A0, A1, A2, A3*)
- I²C-Schnittstelle (*Inter-Integrated Circuit, I2C*)

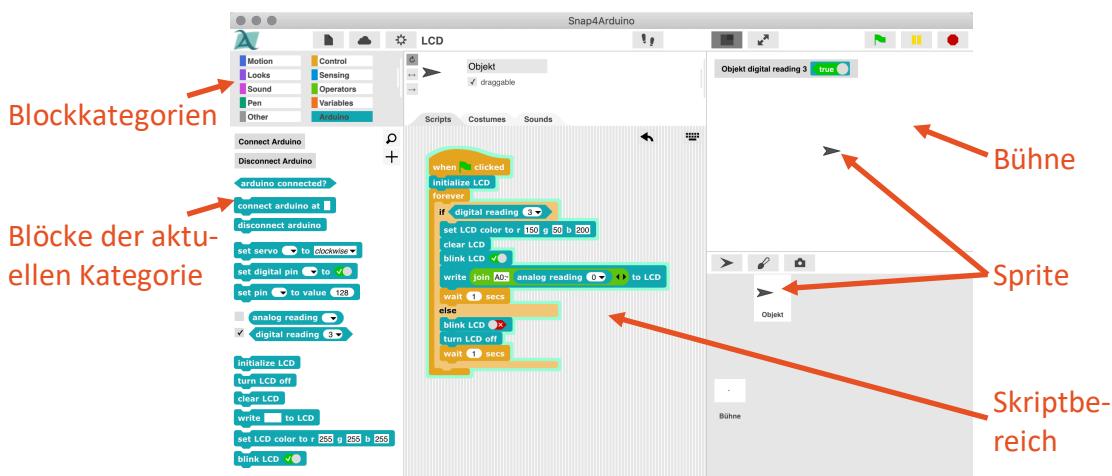


Das zusammengesteckte Board mit Shield wird dann per USB-Kabel mit dem Computer verbunden.

¹ PWM: Pulsweitenmodulation; Verfahren zur Erzeugung pseudoanaloger Signale durch Darstellung relativ hoher Menge diskreter Werte im jeweiligen Wertebereich (hier: 256)

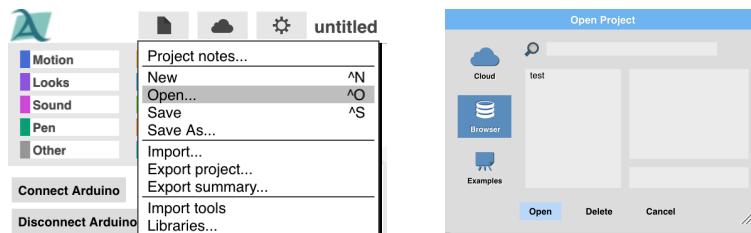


Die Snap4Arduino-Programmierumgebung besteht aus verschiedenen Bereichen:

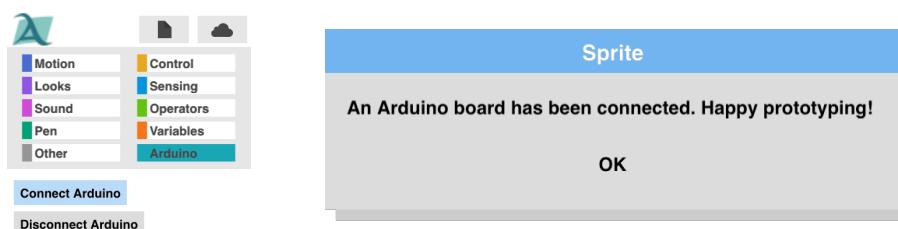


Projekte erstellen, öffnen und laden

Um ein neues Arduino-Projekt in Snap4Arduino zu erstellen, wird das Programm geöffnet. Die Sprache lässt sich über das „Zahnrad“-Symbol einstellen. Snap4Arduino erzeugt automatisch ein neues Projekt. Ein vorhandenes Projekt öffnet man über das Datei-Symbol und die Auswahl „Open...“ („Öffnen...“):



Projekte von einer externen Quelle können geladen werden, indem man im selben Menü auf „Import...“ („Importieren...“) klickt und das entsprechende Projekt auswählt. Bevor der Arduino programmiert werden kann, muss er noch mit dem aktuellen Sprite verbunden werden². Hierzu klickt man in der Übersicht der Blockkategorien auf „Arduino“ und anschließend auf „Connect Arduino“ („Mit Arduino verbinden“). Nach kurzer Zeit erscheint eine Meldung, die die erfolgreiche Verbindung bestätigt:



Falls das Projekt neu angelegt wurde, sollte es anschließend mit eigenem Namen gesichert („Save as...“ bzw. „Sichern als...“ im Dateimenü) und zwischendurch immer wieder gespeichert werden („Save“ bzw. „Sichern“).

² Jeder Sprite hat einen eigenen Skriptbereich und kann mit einem eigenen Arduino verbunden werden. So können auch Projekte erstellt werden, die mehrere interaktive Objekte einbeziehen.



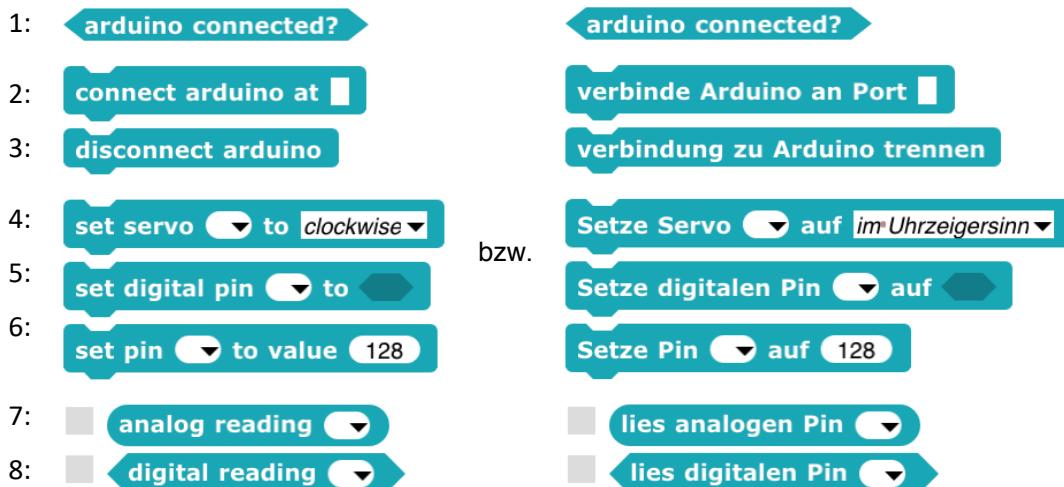
Programme zur Arduino-Ansteuerung erstellen

Snap4Arduino kommuniziert mit dem Arduino Uno über die Blöcke in der Kategorie „Arduino“. Zur Erstellung eines Programms werden außerdem Blöcke aus den Kategorien „Control“ („Steuerung“), „Operators“ („Operatoren“) und „Variables“ („Variablen“) benötigt. Die anderen Kategorien dienen ausschließlich dazu, Sprites auf der Bühne zu manipulieren. Ein Snap4Arduino-Programm beginnt typischerweise mit dem Block **when green flag clicked** bzw. **Wenn grüne Flagge angeklickt**. Daran anschließend folgt die Erstellung des eigentlichen Programms, indem die benötigten Blöcke per Drag & Drop aus den jeweiligen Kategorien auf die Bühne gezogen und zusammengeklickt werden, z. B.:



Die Arduino-Blöcke

Es gibt verschiedene Arduino-Blöcke. Diese werden genutzt, die Verbindung mit dem Board zu prüfen, herzustellen oder zu trennen und um an den einzelnen Pins anliegende Signale der angeschlossenen Sensoren auszulesen und Signale über die Pins nach außen an die Aktoren und Displays zu geben:



- 1: prüfen, ob Arduino mit dem aktuellen Sprite verbunden ist
- 2: Arduino mit dem aktuellen Sprite verbinden, dazugehörigen USB-Port angeben
- 3: Verbindung von Arduino und Sprite trennen
- 4: Servomotoren ansteuern
- 5: Digitale Ausgänge ansteuern
- 6: PWM-Ausgänge ansteuern
- 7: analoge Sensoren auslesen (Häkchen setzen für Anzeige auf der Bühne)
- 8: digitale Sensoren auslesen (Häkchen setzen für Anzeige auf der Bühne)

Nähere Erläuterungen zur Verwendung der verschiedenen Blöcke finden sich auf den folgenden Seiten bei den jeweiligen Komponenten. Zusätzliche Blockbibliotheken können über das Dateimenü genauso importiert werden, wie Projekte.



Sensoren sind technische Bauteile, die Eigenschaften der Umgebung erfassen (z. B. Temperatur, Helligkeit, Druck) und (meist) elektrische Signale abgeben, aus denen der Wert der zugehörigen Messgröße abgeleitet werden kann.

Taster und digitale Sensoren (Binärwert)

Bei den folgenden Modulen handelt es sich um Taster und Sensoren, deren Signale genau zwei diskrete Werte enthalten können: *0* oder *1*, bzw. *falsch* oder *wahr*:



Taster und digitale Sensoren werden an die digitalen Ein-/Ausgänge angeschlossen (*Digital Input/Output D2, D3, ..., D8*).

Zum Auslesen der Sensorwerte, die immer entweder *true (wahr)* oder *false (falsch)* sind, wird in Snap4Arduino der folgende Block aus der „Arduino“-Kategorie genutzt:

digital reading 4 ▾

bzw.

lies digitalen Pin 4 ▾

Hier wird die Nummer des Pins eingetragen, an den der Sensor angeschlossen ist.

Um die Werte auf der Bühne dauerhaft anzeigen zu lassen, kann der „say“-Block aus der „Looks“-Kategorie verwendet und mit dem „forever“-Block aus der „Control“-Kategorie umschlossen werden:

forever

say **digital reading** 4 ▾

bzw.

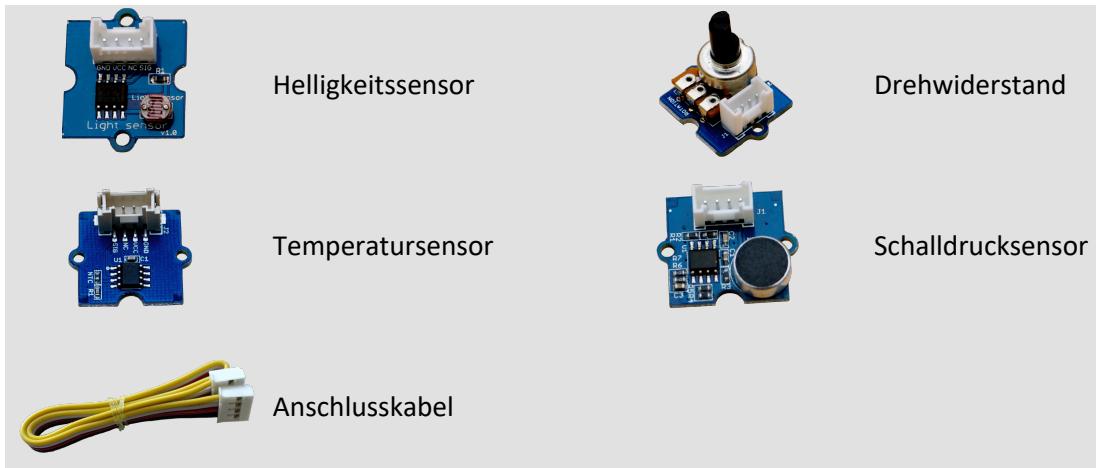
fortlaufend

sage **lies digitalen Pin** 4 ▾



Analoge Sensoren (wertdiskret)

Bei den folgenden Modulen handelt es sich um analoge Sensoren, die theoretisch jeden beliebigen Wert zwischen einem Minimum und einem Maximum liefern können. Da sie digital ausgelesen werden, setzt ein Analog-Digital-Wandler die analogen Eingangssignale in einen digitalen Datenstrom mit einer begrenzten (aber oft relativ hohen) Anzahl diskreter Werte um, der dann weiterverarbeitet oder gespeichert werden kann (hier: 0-1023).



Analoge Sensoren werden an die analogen Eingänge (*Analog Input A0, A1, A2, A3*) angeschlossen:

Zum Auslesen der Werte, die immer zwischen 0 und 1023 liegen, wird in Snap4Arduino der folgende Block aus der „Arduino“-Kategorie genutzt:

analog reading 0 ▾ bzw. **lies analogen Pin** 0 ▾

Hier wird die Nummer des Pins eingetragen, an den der Sensor angeschlossen ist.

Um die Werte auf der Bühne dauerhaft anzeigen zu lassen, kann der „say“-Block aus der „Looks“-Kategorie verwendet und mit dem „forever“-Block aus der „Control“-Kategorie umschlossen werden:

forever
say **analog reading** 0 ▾ bzw. **fortlaufend**
sage **lies analogen Pin** 0 ▾

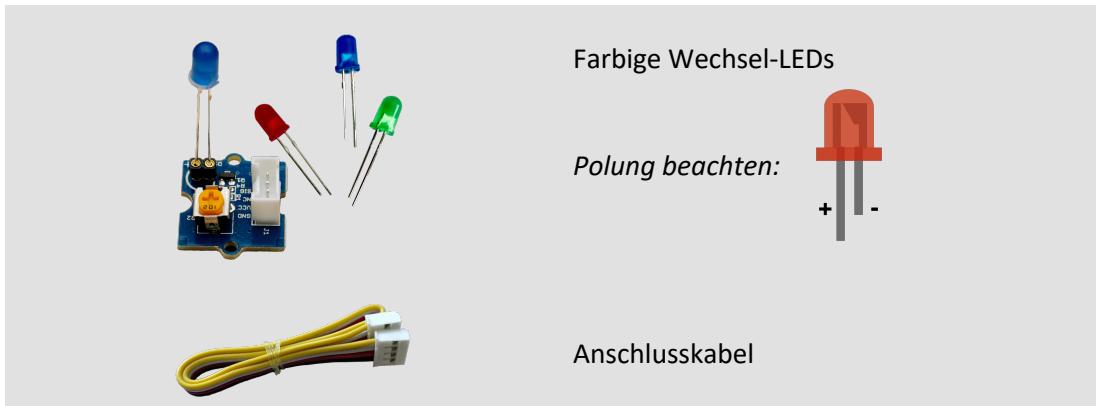
Aktoren sind technische Bauteile, die elektrische Signale in physikalische Größen umsetzen, wie beispielsweise Licht oder Bewegung. Sie sind also sozusagen das Gegenstück von Sensoren.

Aktoren werden an die digitalen Ein-/Ausgänge angeschlossen (*Digital Input/Output D2, D3, ..., D8*).



LEDs

Bei den LED-Modulen handelt es sich um Aktoren, die sowohl binär als auch mittels Pulsweitenmodulation (PWM) angesteuert werden können:



Zum Ein- oder Ausschalten der LEDs wird in Snap4Arduino der folgende Block aus der „Arduino“-Kategorie genutzt:

set digital pin 3 ▾ to ✓

bzw.

Setze digitalen Pin 3 ▾ auf ✓

Hier werden der Pin, an den die LED angeschlossen ist, und der zu sendende Status (*an* oder *aus*) eingetragen.

Um die Helligkeit einer LED steuern zu können, wird in Snap4Arduino der folgende Block aus der „Arduino“-Kategorie genutzt:

set pin 3 ▾ to value 128

bzw.

Setze Pin 3 ▾ auf 128

Hier wird der zu sendende Helligkeitswert (zwischen 0 und 255) und der Pin, an den die LED angeschlossen ist, eingetragen. Beachte: eine solche pseudoanaloge Ansteuerung funktioniert nur an speziellen PWM-Pins (*D3, D5, D6*).



Piezo-Summer

Bei den Piezo-Summern handelt es sich um Aktoren, die mittels Pulsweltenmodulation (PWM) angesteuert werden:



Der Piezo-Summer wird an einen der PWM-Pins der digitalen Ein-/Ausgänge angeschlossen (*Digital Input/Output D3, D5, D6*).

Zum Ansteuern der Summer werden in Snap4Arduino die folgenden Blöcke aus der „Arduino“-Kategorie genutzt:

set pin 3 ▾ to value 128

bzw.

Setze Pin 3 ▾ auf 128

Hier wird der zu sendende Wert (aus: 0 oder 255; an: 1-254) und der Pin, an den der Summer angeschlossen ist, eingetragen. Je nach Wert variiert der ausgegebene Ton leicht in Klang und Lautstärke.



Servomotoren

Bei den Servomotoren handelt es sich um Aktoren:



Servomotor (Standard)

Hinweis: sollten die Motoren nicht richtig funktionieren, hilft es meist, eine externe Stromquelle anzuschließen.

Zum Ansteuern der Servomotoren werden in Snap4Arduino folgende Blöcke aus der „Arduino“-Kategorie genutzt:

- 1: `set servo ▾ to 90 ▾`
- 2: `set servo ▾ to clockwise ▾`
`set servo ▾ to counter-clockwise ▾`
`set servo ▾ to stopped ▾`
- 3: `set servo ▾ to disconnected ▾`

bzw.

- `Setze Servo ▾ auf 90 ▾`
- `Setze Servo ▾ auf im Uhrzeigersinn ▾`
`Setze Servo ▾ auf gegen den Uhrzeigersinn ▾`
`set servo ▾ to stopped ▾`
- `Setze Servo ▾ auf disconnected ▾`

Hier werden der Pin, an den der Servomotor angeschlossen ist, und der zu sendende Wert (*siehe unten*) eingetragen.

Bei der Ansteuerung der Servomotoren gibt es folgende Möglichkeiten:

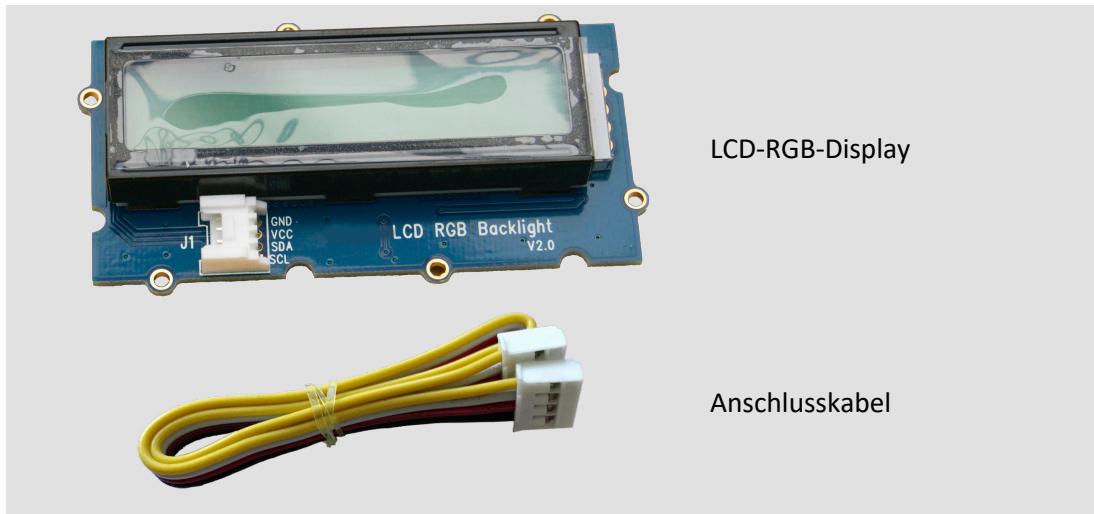
- 1: Servomotoren (Standard): Winkel zwischen ca. 0° und 180°
- 2: Servomotoren (Continuous Rotation (CR)³ – dauerhaft drehend):
 - drehen im Uhrzeigersinn (Wert ca. zwischen 1000 und 1475)
 - drehen gegen Uhrzeigersinn (Wert ca. zwischen 1475 und 2000)
 - anhalten: Wert ca. 1475
- 3: Verbindung zum Servomotor trennen

Die genauen Werte müssen experimentell ermittelt werden.

³ In den meisten Starter-Kits sind ausschließlich Standard-Servomotoren enthalten, CR-Servomotoren müssen zusätzlich erworben werden.



Bei dem folgenden Modul handelt es sich um ein LCD-RGB-Display:



Das LCD-RGB-Display wird an einen I^2C -Eingang angeschlossen (*Inter-Integrated Circuit, I²C*).

Zum Ansteuern des Displays wird die Block-Bibliothek „LCD-Blocks.xml“ benötigt. Diese kann über das Menü („Import...“ bzw. „Importieren...“) geladen werden, oder indem die Datei per Drag & Drop auf den Skriptbereich gezogen wird. Anschließend werden in Snap4Arduino die folgenden Blöcke aus der „Arduino“-Kategorie genutzt:

- | | | |
|----|-------------------------------|----------------------------------|
| 1: | initialize LCD | initialisiere LCD |
| 2: | turn LCD off | schalte LCD ab |
| 3: | clear LCD | leere LCD |
| 4: | blink LCD ✓ | lasse LCD blinken ✓ |
| 5: | set LCD color to r 0 g 85 b 0 | setze LCD-Farbe auf r 0 g 85 b 0 |
| 6: | write Text to LCD | schreibe Text auf LCD |

bzw.

Im Gegensatz zu den meisten sonstigen Modulen ist die Eingabe eines Pins bei Verwendung des Displays nicht nötig, da es automatisch erkannt wird.

Mit den Blöcken werden jeweils bestimmte Display-Funktionen bereitgestellt:

- 1: Display wird initialisiert (eingeschaltet und auf Ausgangszustand gesetzt)
 - 2: Display wird abgeschaltet
 - 3: Display wird geleert (Text wird gelöscht)
 - 4: Blink-Methode wird gestartet oder gestoppt
 - 5: Display-Hintergrundfarbe wird mit RGB-Werten eingestellt
 - 6: Eingegebener Text wird auf dem Display ausgegeben