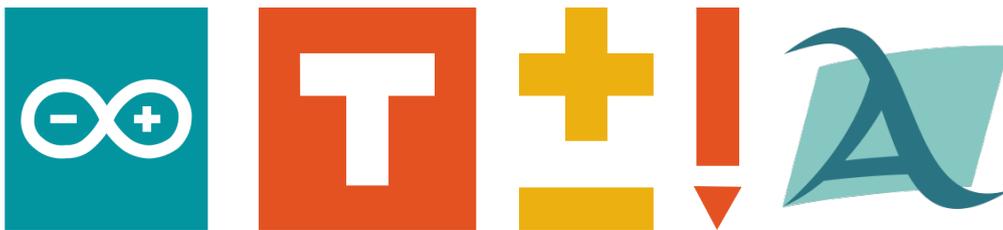




ARDUINO TINKERKIT UND SNAP4ARDUINO



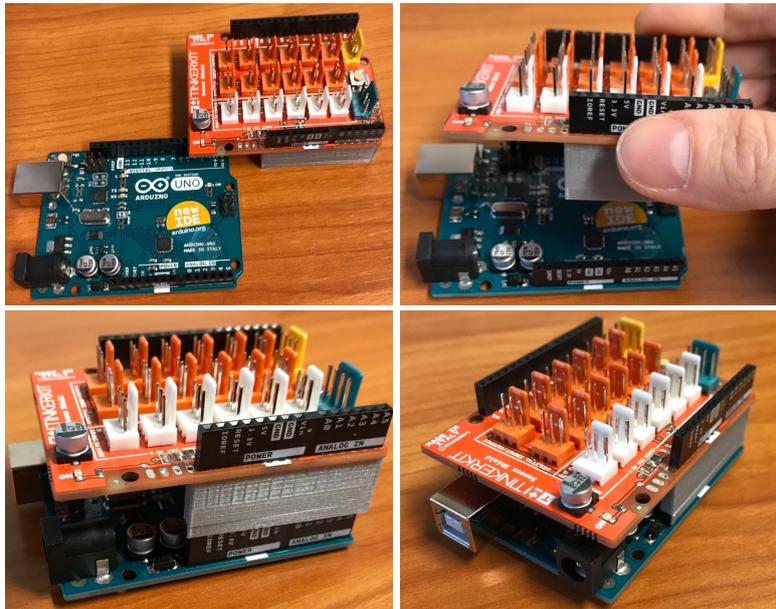
Anleitung



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.....	6
Taster und Digitale Sensoren (Binärwert)	6
Analoge Sensoren (wertdiskret).....	7
Sensoren mit mehreren Anschlüssen (wertdiskret)	9
<i>Analoge Sensoren mit mehreren Anschlüssen.....</i>	<i>9</i>
<i>Digitale Sensoren mit mehreren Anschlüssen</i>	<i>10</i>
Aktoren	11
LEDs	11
Piezo-Summer	12
Servomotoren	13
Displays	14

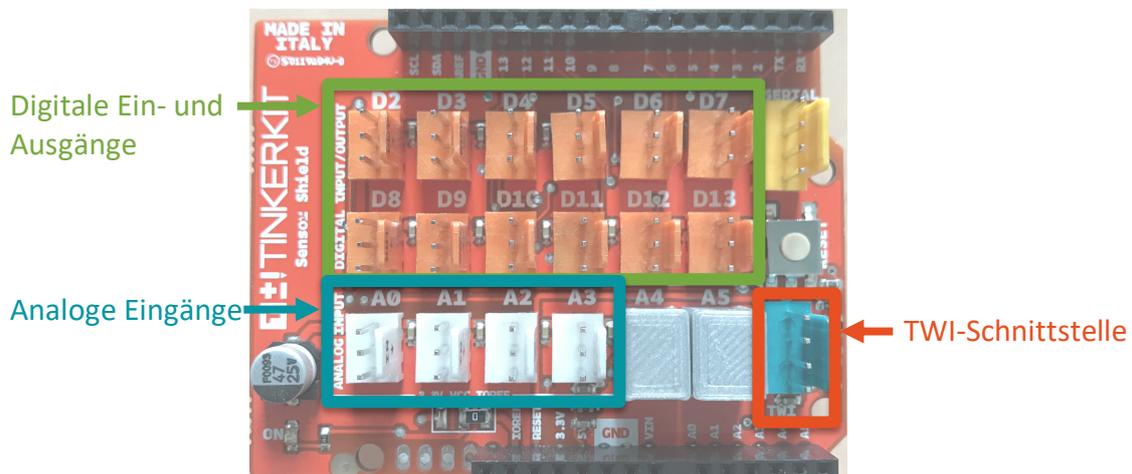


Um die Arbeit mit den elektronischen Bauteilen zu erleichtern, wird auf den Arduino Uno (oder ein kompatibles Board) das TinkerKit-Shield aufgesteckt:



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

- Digitale Ein- und Ausgänge (*Digital Input/Output D2, D3, ..., D13* – orange Stecker), davon PWM¹: D3, D5, D6, D9, D10, D11
- Analoge Eingänge (*Analog Input A0, A1, A2, A3* – weiße Stecker)
- TWI-Schnittstelle (*Two-Wire-Interface, TWI* – blauer Stecker)

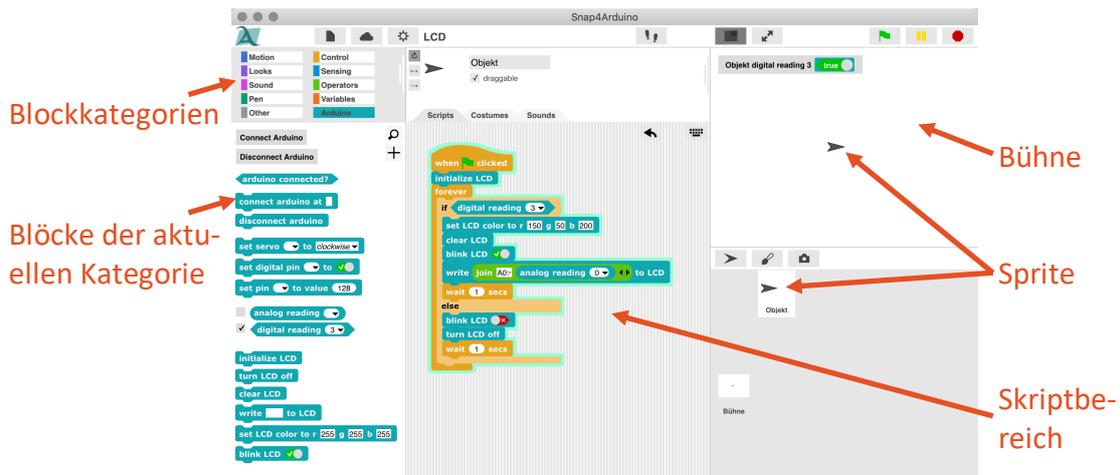


Das zusammengesteckte Board mit Shield wird dann per USB-Kabel mit dem Computer verbunden.

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

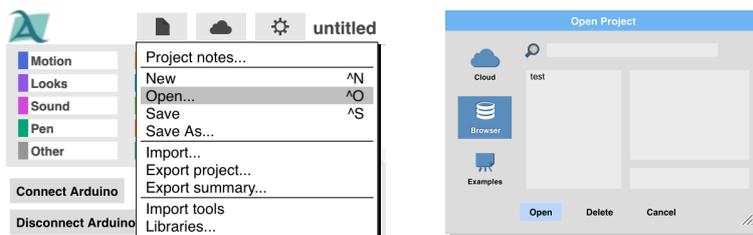


Die Snap4Arduino-Programmierumgebung besteht aus verschiedenen Bereichen:

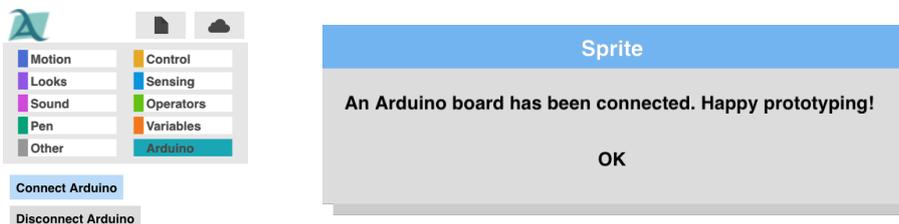


Projekte erstellen, öffnen und laden

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.



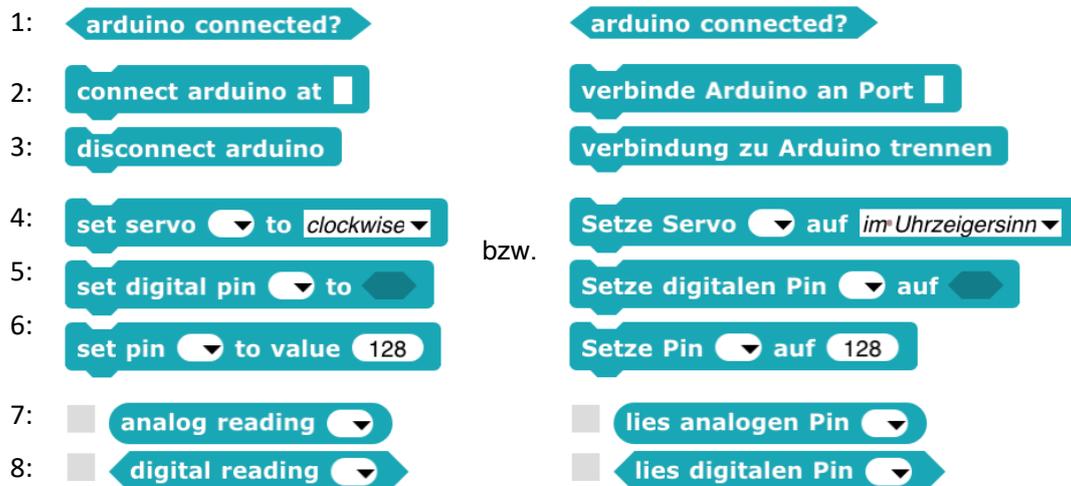
Programme zur Arduino-Ansteuerung erstellen

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 clicked` bzw. `Wenn 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.

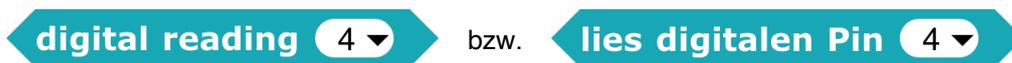
Taster und Digitale Sensoren (Binärwert)

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

	Taster		Anschlusskabel (rot-orange-schwarz)
	Neigungssensor		Kippschalter
	Berührungssensor		Anschlusskabel (schwarz-rot-gelb, ohne Markierung)
	Bewegungssensor (PIR)		

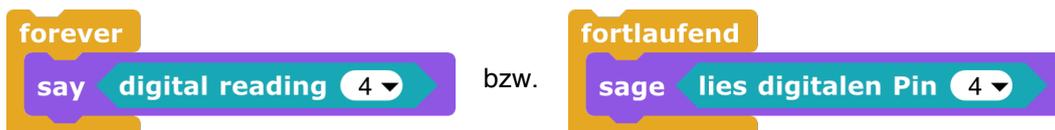
Digitale Sensoren werden an die digitalen Ein-/Ausgänge angeschlossen (*Digital Input/Output D2, D3, ..., D13* – orange Stecker).

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



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

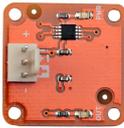
Um die Werte auf der Bühne dauerhaft anzeigen zu lassen, kann der „say“-Block aus der „Looks“-Kategorie verwendet und mit dem „forever“-Block aus der „Control“-Kategorie umschlossen werden:





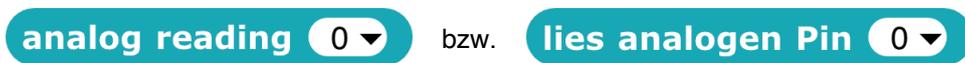
Analoge Sensoren (wertdiskret)

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).

	Helligkeitssensor		Magnetfeldsensor
	Temperatursensor		Anschlusskabel (rot-orange-schwarz)
	Schiebewiderstand		Schalldrucksensor
	Drehwiderstand		Anschlusskabel (schwarz-rot-gelb, ohne Markierung)

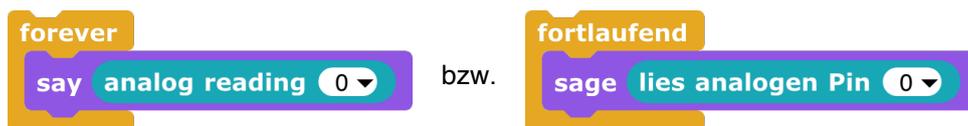
Analoge Sensoren werden an die analogen Eingänge angeschlossen (*Analog Input* A0, A1, ..., A5 – weiße Stecker).

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



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

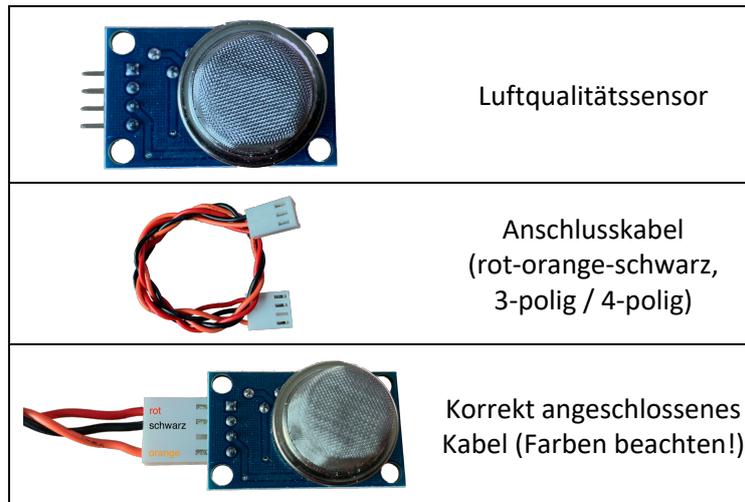
Um die Werte auf der Bühne dauerhaft anzeigen zu lassen, kann der „say“-Block aus der „Looks“-Kategorie verwendet und mit dem „forever“-Block aus der „Control“-Kategorie umschlossen werden:





Luftqualitätssensor

Bei dem folgenden Modul handelt es sich um einen Luftqualitätssensor (MQ135).



Auch bei den Luftqualitätssensoren handelt es sich um analoge Sensoren mit den zuvor beschriebenen Eigenschaften. Sie können daher auf die gleiche Weise ausgelesen werden, wie alle anderen analogen Sensoren (→ *Analoge Sensoren (wertdiskret)*). Zusätzlich kann der aktuelle CO₂-Gehalt der Luft mit den folgenden Blöcken bestimmt werden:

- | | | | |
|----|--|------|--|
| 1: | CO2 value from sensor on pin 0 | bzw. | CO2-Wert von Sensor an Pin 0 |
| 2: | Calibrated R0 for 412.55 ppm from pin 0 | bzw. | Kalibrierter R0 für 412.55 ppm an Pin 0 |
| 3: | CO2 value on pin 0 with R0 = 74 | bzw. | CO2-Wert an Pin 0 mit R0 = 74 |

Hier werden der Pin, an den der Sensor angeschlossen ist und ggf. weitere Werte für die Kalibrierung des Sensors eingetragen (*siehe unten*).

- 1: gibt aktuellen CO₂-Wert aus (Kaltwiderstand R₀ ist voreingestellt auf 74)
- 2: gibt den kalibrierten Kaltwiderstandswert R₀ aus (voreingestellter Wert 412,55 ppm zur Kalibrierung an frischer Luft, siehe z. B. <https://www.co2.earth>)
- 3: gibt aktuellen CO₂-Wert aus (Kaltwiderstand R₀ ist einstellbar)

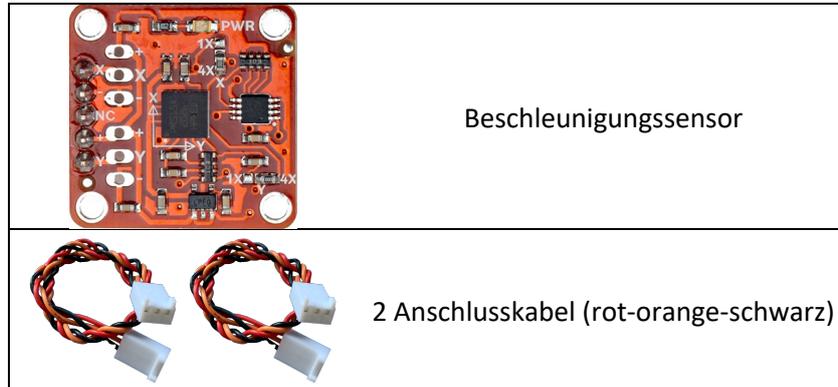


Sensoren mit mehreren Anschlüssen (wertdiskret)

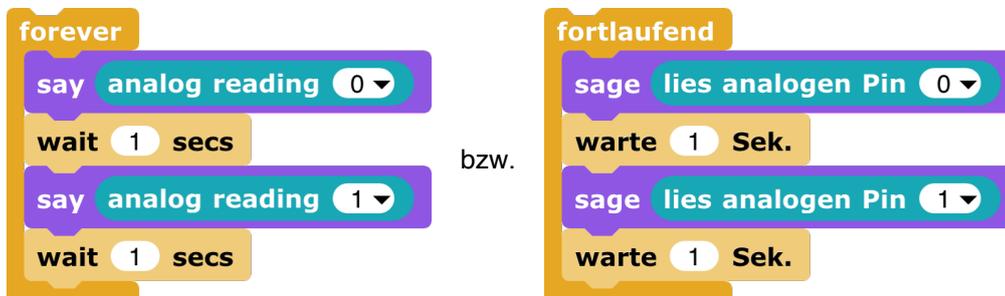
Sensoren mit mehreren Anschlüssen werden mit mehreren Kabeln mit dem Arduino verbunden und liefern entsprechend auch mehrere Werte.

Analoge Sensoren mit mehreren Anschlüssen

Bei dem folgenden Modul handelt es sich um einen analogen Sensor mit zwei Anschlüssen:



Dieser Sensor wird an die analogen Eingänge angeschlossen (*Analog A0, A1, A2, A3* – weiße Stecker) und mit dem „analog reading“-Block aus der „Arduino“-Kategorie ausgelesen. 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, zusätzlich werden „wait“-Blöcke genutzt, um die Werte jeweils für eine Sekunde anzuzeigen:

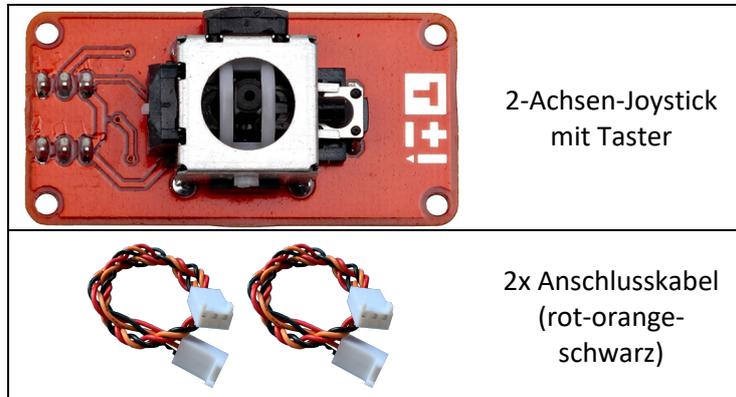


Auch hier werden wieder die Nummern der Pins eingetragen, an die der Sensor angeschlossen ist.



Digitale Sensoren mit mehreren Anschlüssen

Bei dem folgenden Modul handelt es sich um einen digitalen Sensor mit zwei Anschlüssen, der eine feste Anzahl diskreter Zustände annehmen kann, um die Stellung des Joysticks und den Tastendruck zu kennzeichnen. Die zugehörigen Signale befinden sich in jeweils klar definierten, voneinander unterschiedlichen und verhältnismäßig kleinen Wertebereichen.



Der Joystick wird, auch wenn dies der Intuition widerspricht, ebenso wie andere digitale Sensoren, die mehr als genau zwei diskrete Werte liefern, an die analogen Eingänge (*Analog Input A0, A1, A2, A3* – weiße Stecker) angeschlossen und wie dort beschrieben ausgelesen (→ *Sensoren mit mehreren Anschlüssen - Analoge Sensoren*).

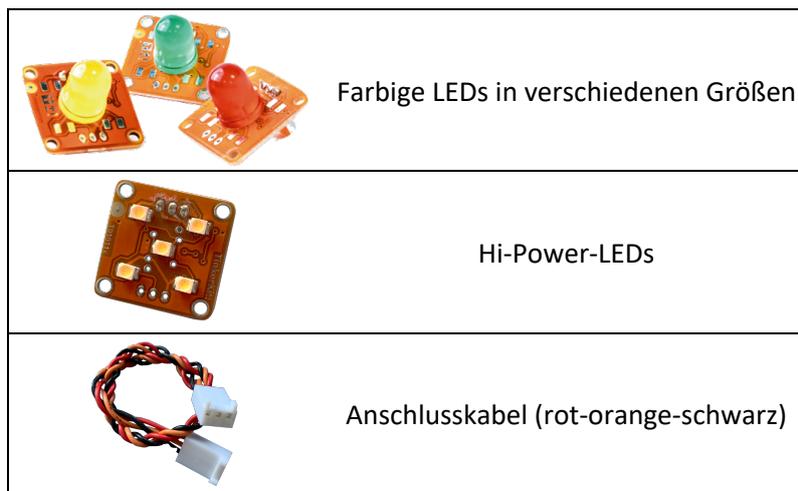


Aktoren sind technische Bauteile, die elektrische Signale in physikalische Größen umsetzen, wie beispielsweise Licht oder Bewegung. Sie sind also sozusagen das Gegenteil von Sensoren.

Aktoren werden an die digitalen Ein-/Ausgänge angeschlossen (*Digital Input/Output D2, D3, ..., D13* – orange Stecker).

LEDs

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



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



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

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

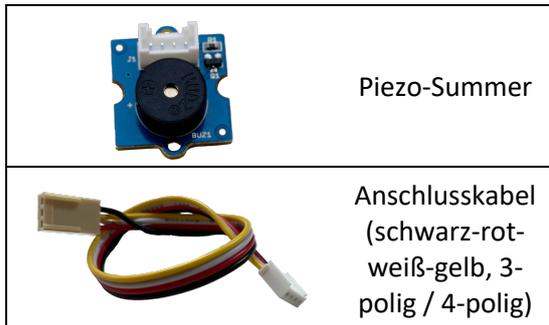


Hier wird der zu sendende Helligkeitswert (zwischen 0 und 255) und der Pin, an den die LED angeschlossen ist, eingetragen. Beachte: eine solche pseudoanaloge Ansteuerung funktioniert nur an speziellen PWM-Pins (*D3, D5, D6, D9, D10, D11*).

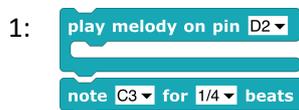


Piezo-Summer

Bei den Piezo-Summern handelt es sich um Aktoren:



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



bzw.



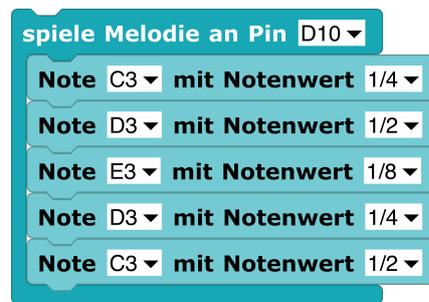
Hier werden der Pin, an den der Summer angeschlossen ist, und die zu sendenden Noten mit den entsprechenden Notenwerten eingetragen (*siehe unten*).

Bei der Ansteuerung der Summer gibt es folgende Möglichkeiten:

1: es wird eine Melodie erstellt und im Gesamten abgespielt, z. B.:



bzw.



2: es wird ein einzelner Ton abgespielt

Eine Melodie könnte auch über das Aneinanderfügen einzelner „play note“-Blöcke abgespielt werden, jedoch hat dies zwei Nachteile:

- a) In Jedem Block muss der Pin angegeben werden, auf dem der Ton ausgegeben werden soll.
- b) Jeder Ton wird einzeln übertragen, so dass unnötiger Datenverkehr erzeugt wird, was zu Verzögerungen in der Programmausführung führen kann.



Servomotoren

Bei den Servomotoren handelt es sich um Aktoren:

	Servomotoren (Standard)
	Servomotoren (CR – Continuous Rotation)
	Adapterkabel (schwarz-rot-gelb, mit roten Ausrufezeichen ! markiert)

Beim Anschluss der Servomotoren an die digitalen Ein-/Ausgänge (*Digital Input/Output D2, D3, ..., D13* – orange Stecker) ist auf Verwendung des korrekten Kabels (siehe oben) zu achten! **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:			
2:			
		bzw.	
			
3:			

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 RGB-Backlight-LCD (Liquid Chrystal Display):



Das RGB-Backlight-LCD wird an den *TWI*-Eingang angeschlossen (*Two-Wire-Interface*, *TWI* – blauer Stecker).

Zum Ansteuern des Displays wird die Bibliothek „LCD-blocks.xml“ eingebunden (einfach per Drag&Drop auf den Skritbereich ziehen). Damit werden in Snap4Arduino die folgenden Blöcke in die „Arduino“-Kategorie hinzugefügt und können nun genutzt werden:

- | | | | |
|----|--|------|--|
| 1: | initialize LCD | bzw. | initialisiere LCD |
| 2: | turn LCD off | | schalte LCD ab |
| 3: | clear LCD | | leere LCD |
| 4: | blink LCD <input checked="" type="checkbox"/> | | lasse LCD blinken <input checked="" type="checkbox"/> |
| 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 |

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: Die Blink-Methode wird gestartet oder gestoppt
- 5: Display-Hintergrundfarbe wird mit RGB-Werten eingestellt
- 6: Eingegebener Text wird auf dem Display ausgegeben