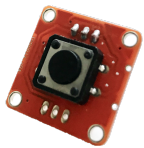
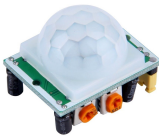
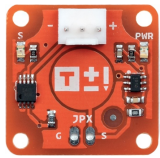


Station 1: Taster und Digitale Sensoren (Binärwert)

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

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:

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 ▼

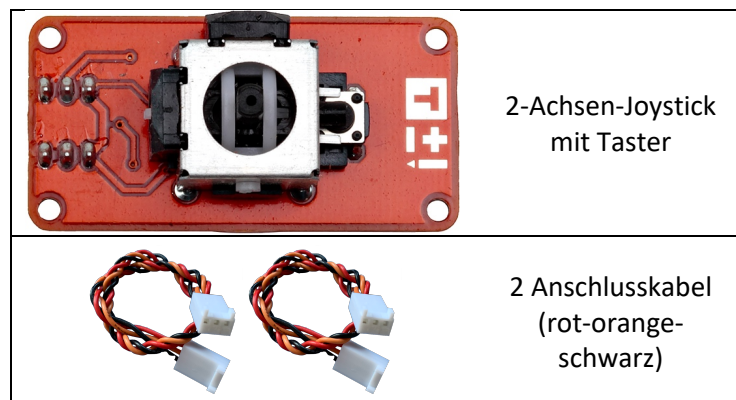
Aufgaben:

- 1) Schließe einen Kippschalter an den Pin D2 vom Arduino an und lasse dir den aktuellen Wert auf der Bühne anzeigen. Achte auf die korrekte Einstellung im Sensor-Block!
 - a. Wann wird über diesen Sensor der Status *true* und wann *false* ausgelesen?
 - b. Was ist der Unterschied zum Taster?
- 2) Verwende zwei weitere digitale Sensoren und ermittle *true*- und *false*-Bedingungen. Schreibe sie dir übersichtlich mit einer kurzen Erklärung auf.
- 3) Für welche beispielhaften Anwendungszwecke können die Sensoren dienen?

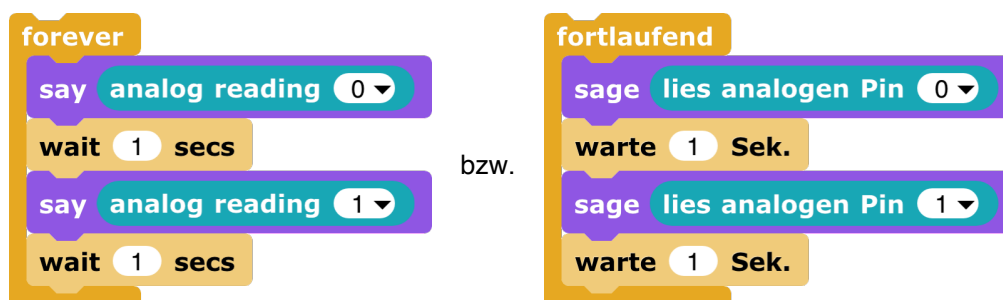
Zusatzaufgabe: Digitale Sensoren mit mehreren Anschlüssen

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

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, 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:



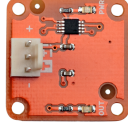
In die blauen Blöcke werden wieder die Nummern der Pins eingetragen, an die der Sensor angeschlossen ist.

Aufgaben:

- 1) Schließe den Joystick an die Pins A2 und A3 vom Arduino an und lasse dir die aktuellen Werte auf der Bühne anzeigen. Achte auf die korrekte Einstellung im Sensor-Block!
 - a. Welche Werte werden bei Betätigung des Hebels in die verschiedenen Richtungen ausgelesen?
 - b. Was bewirkt Druck von oben auf den Hebel?
 - c. Bewege mit dem Joystick den Turtle (das Dreieck) über die Bühne.

Station 2a: 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:

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 ▼

Station 2a: Analoge Sensoren (wertdiskret)

Aufgaben:

- 1) SchlieÙe einen Schiebewiderstand an den Pin A0 vom Arduino an und lasse dir den aktuellen Wert auf der BÙhne anzeigen. Achte auf die korrekte Einstellung im Sensor-Block!
 - a. Welches ist der kleinste Wert, der über diesem Sensor ausgelesen werden kann?
 - b. Welches ist der höchste Wert, der über diesem Sensor ausgelesen werden kann?

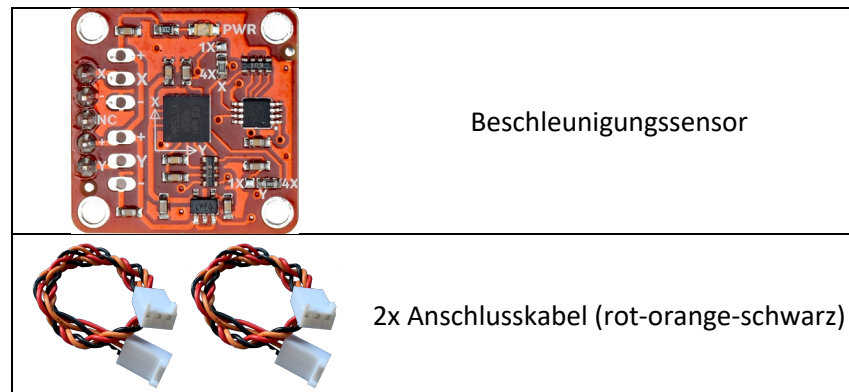
Schreibe sie dir zusammen mit den Werten aus den Aufgaben 2 und 3 in einer Tabelle auf. Schreibe in die erste Spalte eine passende kurze Beschreibung.

- 2) Probiere weitere analoge Sensoren und ermittle deren Minimal- und Maximalwerte, sowie aktuelle Werte bei Raumbedingungen.
- 3) Für welche beispielhaften Anwendungszwecke können die verschiedenen Sensoren dienen?

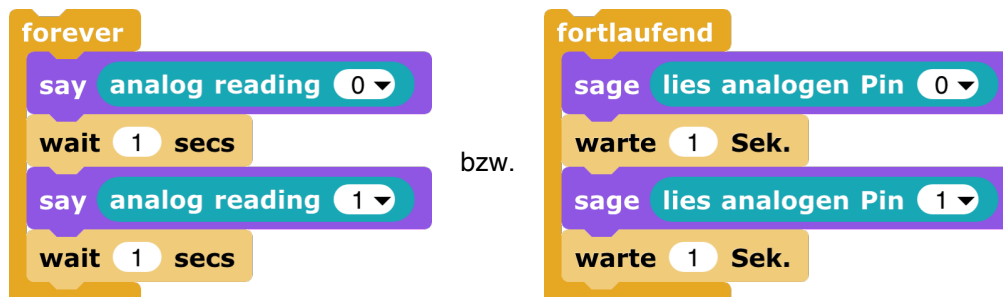
Station 2b: Analoge Sensoren (wertdiskret) mit mehreren Anschlüssen

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

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:

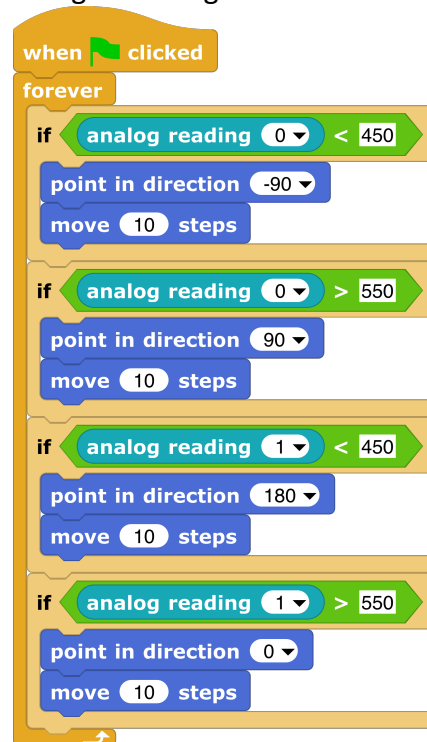


Auch hier werden wieder die Nummern der Pins eingetragen, an die der Sensor angeschlossen ist.

Station 2b: Analoge Sensoren (wertdiskret) mit mehreren Anschlüssen

Aufgaben:

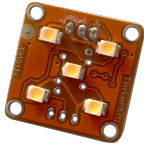
- 1) Schließe den Beschleunigungssensor an die Pins A0 und A1 vom Arduino an und lasse dir die aktuellen Werte auf der Bühne anzeigen. Achte auf die korrekte Einstellung im Sensor-Block!
 - a. Welche Werte werden bei Beschleunigung in die verschiedenen Richtungen ausgelesen?
 - b. Was bewirkt das folgende Programm?



- 2) Für welche beispielhaften Anwendungszwecke kann dieser Sensor dienen?

Station 3: LEDs

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

	Farbige LEDs in verschiedenen Größen
	Hi-Power-LEDs
	Anschlusskabel (rot-orange-schwarz)

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 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:

set pin 9 to value 128 bzw. Setze Pin 9 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, D9, D10, D11).

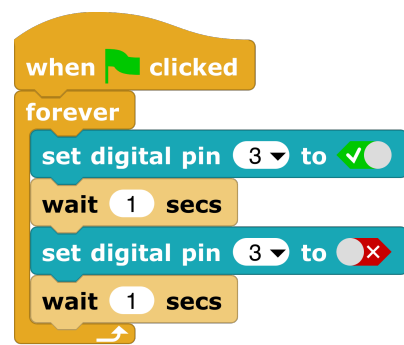
Station 3: LEDs

Aufgaben:

- 1) Schließe eine LED an den Pin D3 vom Arduino.
 - a. Lasse die LED leuchten und schalte sie anschließend wieder aus.
 - b. Was bewirkt das folgende Programm?



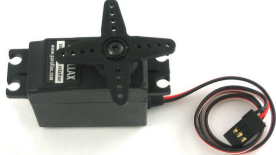
bzw.



- 2) Schreibe ein Programm, das zwei LEDs im Wechsel an- und wieder ausschaltet.
- 3) Schreibe ein Programm, das eine LED im Sekundentakt immer heller werden lässt.
- 4) Für welche beispielhaften Anwendungszwecke können die LEDs dienen?

Station 4: 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:  |  |
|  |  |
|  |  |
| 3:  |  |
- bzw.

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) – 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.

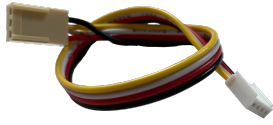
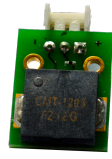
Station 4: Servomotoren

Aufgaben:

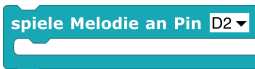
- 1) SchlieÙe einen Standard-Servomotor an den Pin D2 vom Arduino an und stelle den Winkel auf 90°.
- 2) Lasse den Servo „winken“.
- 3) SchlieÙe einen CR-Servo an Pin D5 vom Arduino an und ermittle den korrekten Stoppwert.
- 4) Wie lässt sich die Drehgeschwindigkeit regulieren?

Station 5: Piezo-Summer

Bei den Piezo-Summern handelt es sich um Aktoren:

	Piezo-Summer
	Anschlusskabel (schwarz-rot-weiß-gelb, 3-polig / 4-polig)
	Piezo-Summer
	Anschlusskabel (schwarz-rot-gelb, ohne Markierung)

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

- 1:   bzw.  
- 2:  

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:

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

Station 5: Piezo-Summer

Aufgaben:

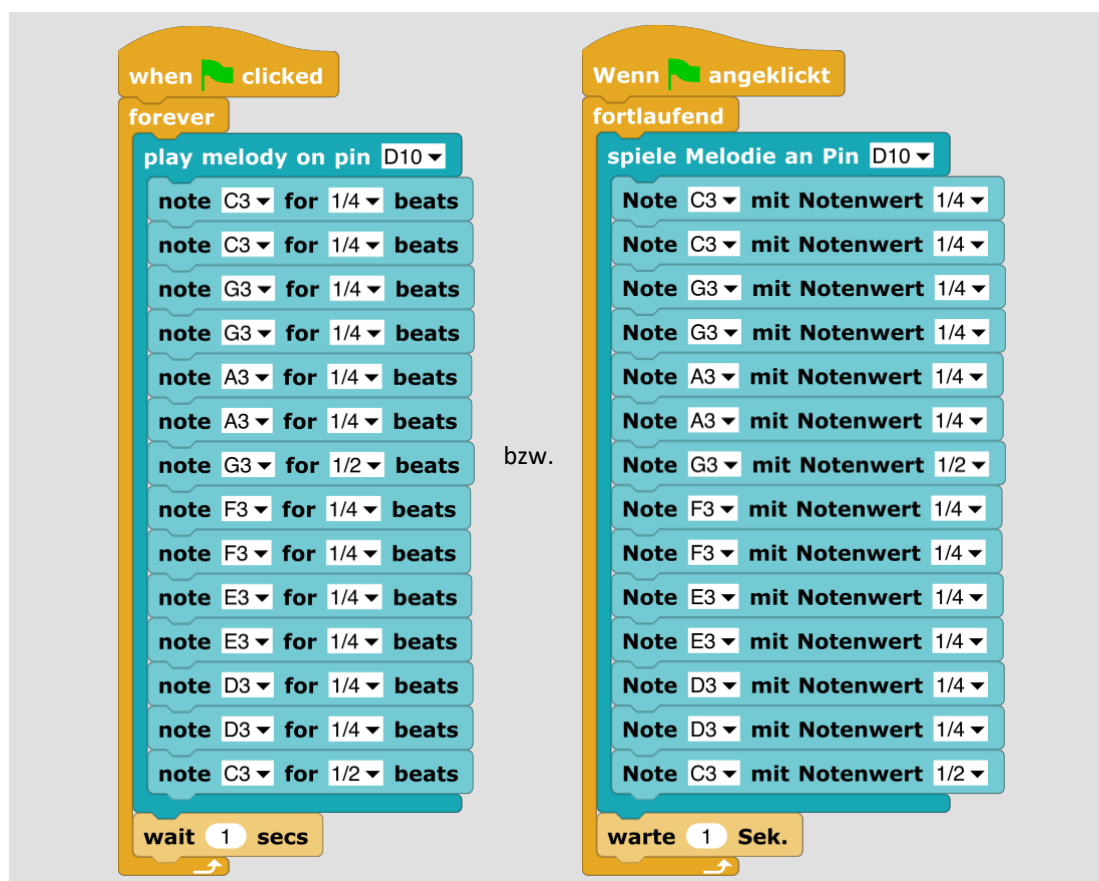
- 1) Schließe einen Summer an Pin D3 des Arduino an.
 - a) Spiele eine kurze Melodie mit fünf unterschiedlich langen Tönen.
 - b) Spiele den Anfang des Kinderliedes „Alle meine Entchen“:

Alle meine Entchen



Quelle: www.klarinettennoten.info

- 2) Was bewirkt das folgende Programm?



The image shows two equivalent Scratch-style code snippets side-by-side, separated by the text "bzw." (or).

Left snippet (English):

- Trigger: **when clicked**
- Loop: **forever**
- Block: **play melody on pin D10**
- Notes and durations:
 - note C3 for 1/4 beats
 - note C3 for 1/4 beats
 - note G3 for 1/4 beats
 - note G3 for 1/4 beats
 - note A3 for 1/4 beats
 - note A3 for 1/4 beats
 - note G3 for 1/2 beats
 - note F3 for 1/4 beats
 - note F3 for 1/4 beats
 - note E3 for 1/4 beats
 - note E3 for 1/4 beats
 - note D3 for 1/4 beats
 - note D3 for 1/4 beats
 - note C3 for 1/2 beats
- Wait block: **wait 1 secs**

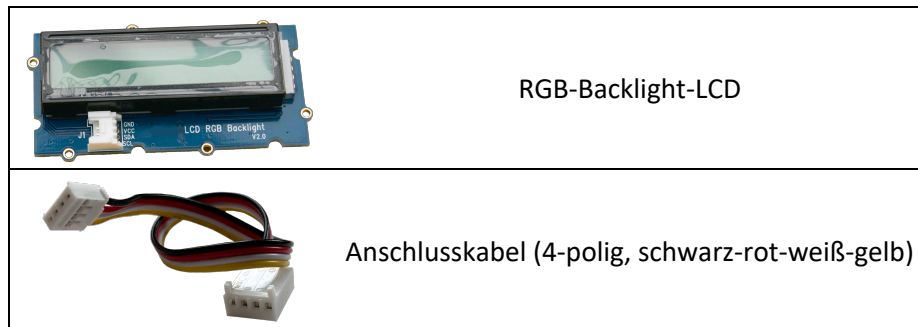
Right snippet (German):

- Trigger: **Wenn angeklickt**
- Loop: **fortlaufend**
- Block: **spiele Melodie an Pin D10**
- Notes and durations:
 - Note C3 mit Notenwert 1/4
 - Note C3 mit Notenwert 1/4
 - Note G3 mit Notenwert 1/4
 - Note G3 mit Notenwert 1/4
 - Note A3 mit Notenwert 1/4
 - Note A3 mit Notenwert 1/4
 - Note G3 mit Notenwert 1/2
 - Note F3 mit Notenwert 1/4
 - Note F3 mit Notenwert 1/4
 - Note E3 mit Notenwert 1/4
 - Note E3 mit Notenwert 1/4
 - Note D3 mit Notenwert 1/4
 - Note D3 mit Notenwert 1/4
 - Note C3 mit Notenwert 1/2
- Wait block: **warte 1 Sek.**

- 3) Erstelle ein Programm, das eine Polizeisirene simuliert.
- 4) Sende eine Morse-Nachricht an einen deiner Mitschüler.
- 5) Für welche weiteren Anwendungszwecke können Summer dienen?

Station 6: Displays

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 Skrittbereich ziehen). Damit werden in Snap4Arduino die folgenden Blöcke in die „Arduino“-Kategorie hinzugefügt und können nun genutzt werden:

- | | | | |
|----|--|------|--|
| 1: | initialize LCD | | initialisiere LCD |
| 2: | turn LCD off | | schalte LCD ab |
| 3: | clear LCD | | leere LCD |
| 4: | blink LCD ☒ | bzw. | 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 |

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: Eingebener Text wird auf dem Display ausgegeben

Station 6: Displays



Aufgaben:

- 1) SchlieÙe das Display an den TWI-Anschluss an und lasse dir den Text „Hallo Welt“ auf dem Display anzeigen.
- 2) Ändere die Farbe der Hintergrundbeleuchtung.
 - a. Welche Bedeutung haben die Werte r, g und b?
 - b. Welche Werte-Tripel erzeugen gelbes, violettes und türkises Licht?
- 3) Teste aus, wieviele Zeichen auf dem Display maximal angezeigt werden können. Was passiert, wenn du diese Anzahl überschreitest?
- 4) Für welchen beispielhaften Anwendungszweck kann das Display dienen?