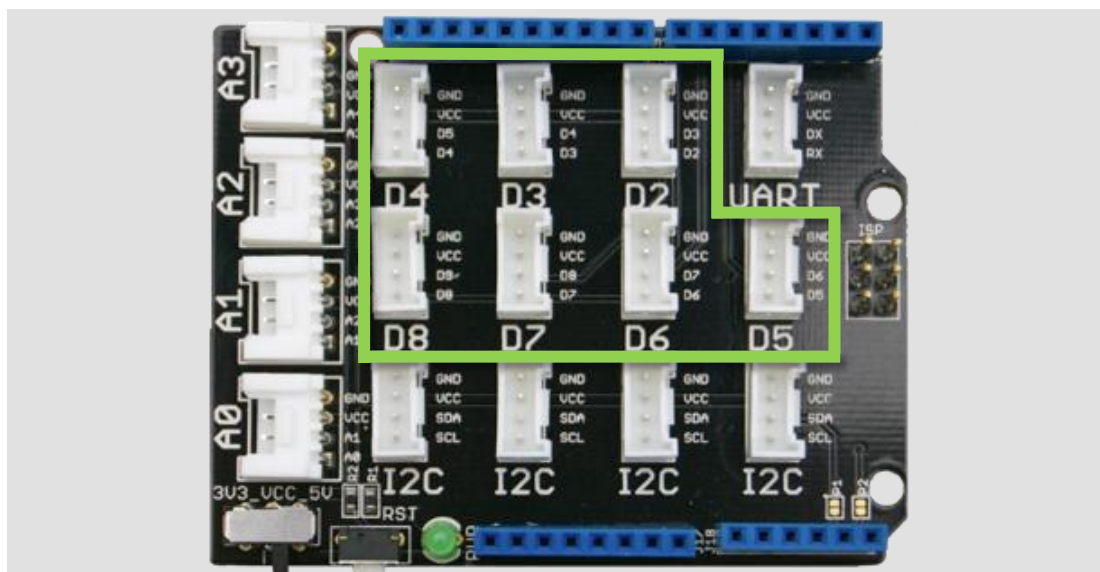




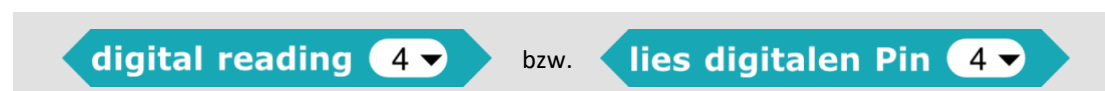
Bei den folgenden Modulen handelt es sich um Taster und Sensoren, deren Signale genau zwei diskrete Werte enthalten können: *0* oder *1*, bzw. *falsch* oder *wahr*:



Taster und digitale Sensoren werden an die digitalen Eingänge (*Digital Input/Output* D2, D3, ..., D8 – im Bild grün umrandet) angeschlossen:



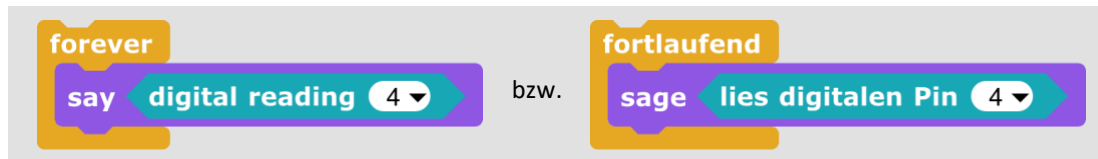
Zum Auslesen der Sensorwerte, die immer entweder *true* (*wahr*) oder *false* (*falsch*) sind, wird in Snap4Arduino der folgende Block aus der „Arduino“-Kategorie genutzt:



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:



Aufgaben:

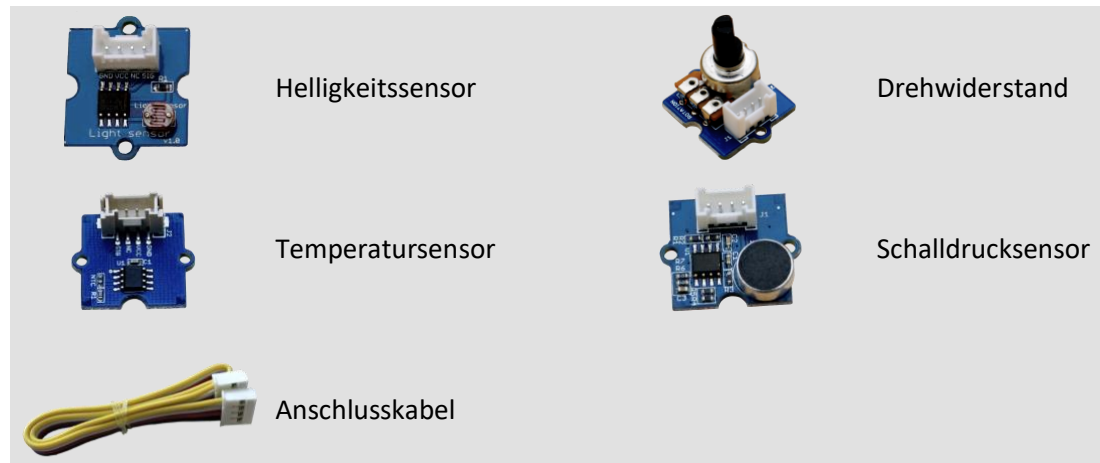
- 1) Schließe einen Taster an den Pin D2 vom Arduino an und lasse dir den aktuellen Wert auf der Bühne anzeigen. Achte auf die korrekte Einstellung im Sensor-Block! Wann wird über diesen Sensor der Status *true* und wann *false* ausgelesen?
- 2) Probiere auch den Berührungssensor und ermittle *true*- und *false*-Bedingungen.
- 3) Für welche beispielhaften Anwendungszwecke können diese Module dienen?

Für die folgenden Aufgaben werden weitere digitale Sensoren benötigt:

- 4) 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?
- 5) Probiere weitere digitale Sensoren und ermittle *true*- und *false*-Bedingungen. Schreibe sie dir übersichtlich mit einer kurzen Erklärung auf.
- 6) Für welche beispielhaften Anwendungszwecke können die Sensoren dienen?



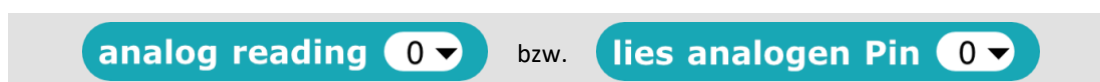
Bei den folgenden Modulen handelt es sich um analoge Sensoren, die theoretisch jeden beliebigen Wert zwischen einem Minimum und einem Maximum liefern können. Da sie digital ausgelesen werden, setzt ein Analog-Digital-Wandler die analogen Eingangssignale in einen digitalen Datenstrom mit einer begrenzten (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* – im Bild grün umrandet) angeschlossen:



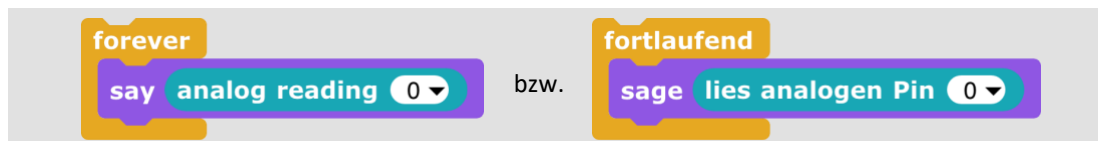
Zum Auslesen der Werte, die immer zwischen 0 und 1023 liegen, wird in Snap4Arduino der folgende Block aus der „Arduino“-Kategorie genutzt:



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:

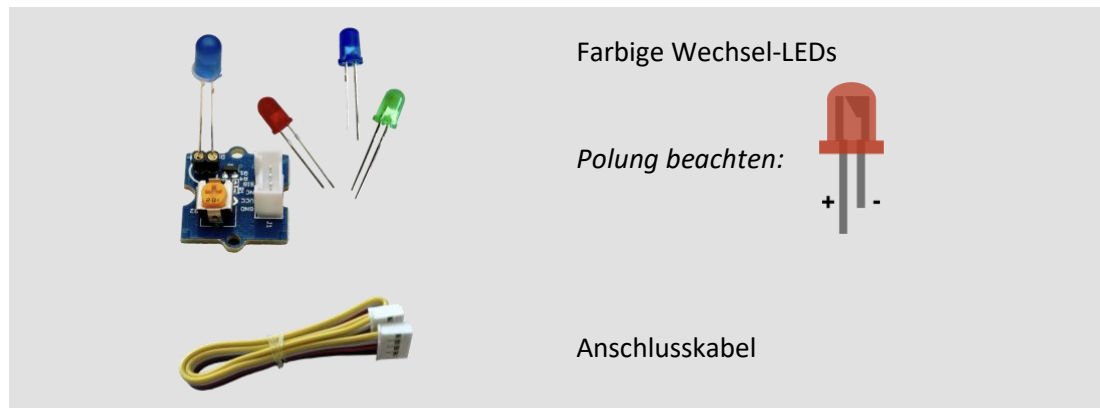


Aufgaben:

- 1) Schließe einen Drehwiderstand an den Pin A0 vom Arduino an und lasse dir den aktuellen Wert auf der Bühne 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) Nutze einen Helligkeitssensor und ermittle die Sensorwerte:
 - a) bei sehr dunkler Umgebung
 - b) bei sehr heller Umgebung
 - c) bei Tageslicht
 - d) bei eingeschaltetem Licht im Klassenzimmer
- 3) Probiere weitere analoge Sensoren und ermittle deren Minimal- und Maximalwerte, sowie aktuelle Werte bei Raumbedingungen.
- 4) Für welche beispielhaften Anwendungszwecke können die verschiedenen Sensoren dienen?



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



LED-Module werden an die digitalen Ausgänge (*Digital Input/Output D2, D3, ..., D8* – im Bild grün umrandet) angeschlossen:



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



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:

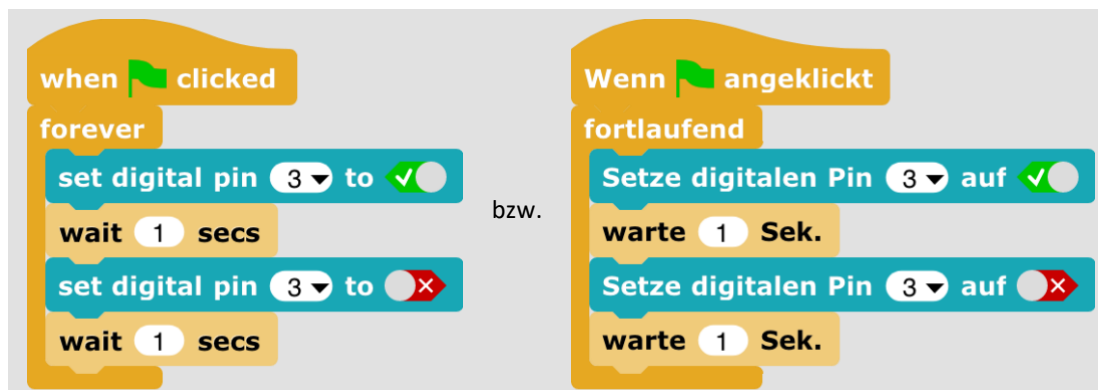




Hier wird der zu sendende Helligkeitswert (zwischen 0 und 255) und der Pin, an den die LED angeschlossen ist, eingetragen. Beachte: eine solche pseudoanaloge Ansteuerung funktioniert nur an speziellen PWM-Pins (D3, D5, D6).

Aufgaben:

- 1) Schließe eine LED an den Pin D2 vom Arduino an.
 - a) Lasse die LED leuchten und schalte sie anschließend wieder aus.
 - b) Was bewirkt das folgende Programm?



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



Bei den Servomotoren handelt es sich um Aktoren:



Servomotoren werden an die digitalen Ausgänge (*Digital Input/Output D2, D3, ..., D8* – im Bild grün umrandet) angeschlossen:

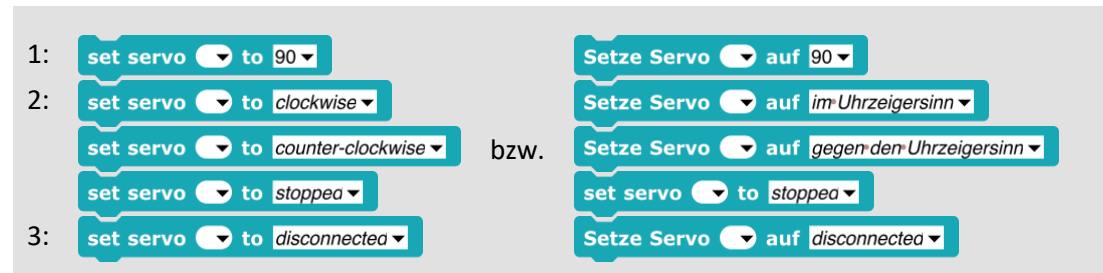


! Hinweis:

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:



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.

Aufgaben:

- 1) Schließe einen Standard-Servomotor an den Pin D2 vom Arduino an und stelle den Winkel auf 90°.
- 2) Lasse den Servo „winken“.

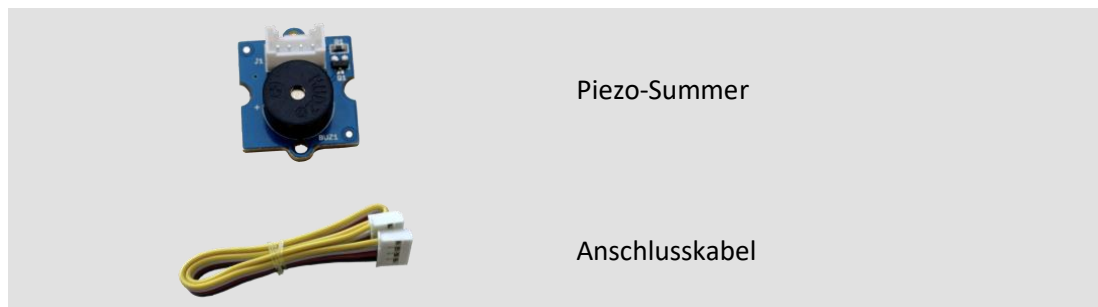
Für die folgenden Aufgaben wird ein CR-Servomotor benötigt:

- 3) Schließe einen CR-Servomotor an Pin D5 vom Arduino an und ermittle den korrekten Stoppwert.
- 4) Wie lässt sich die Drehgeschwindigkeit regulieren?

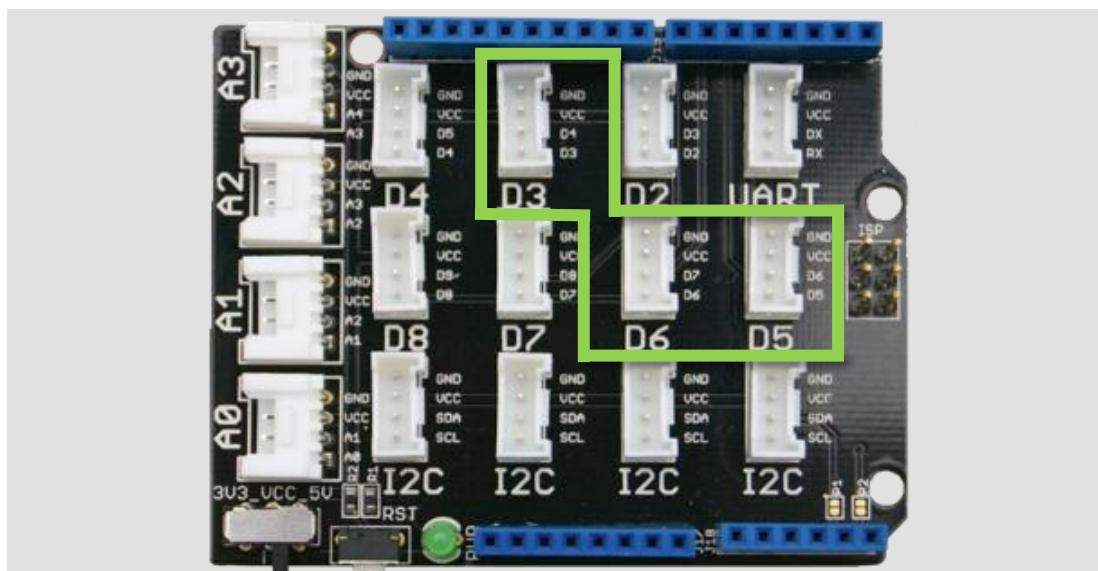
* In den meisten Starter-Kits sind ausschließlich Standard-Servomotoren enthalten, CR-Servomotoren müssen zusätzlich erworben werden.



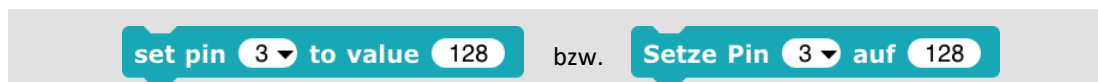
Bei den Piezo-Summern handelt es sich um Aktoren, die mittels Pulsweitenmodulation (PWM) angesteuert werden:



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



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

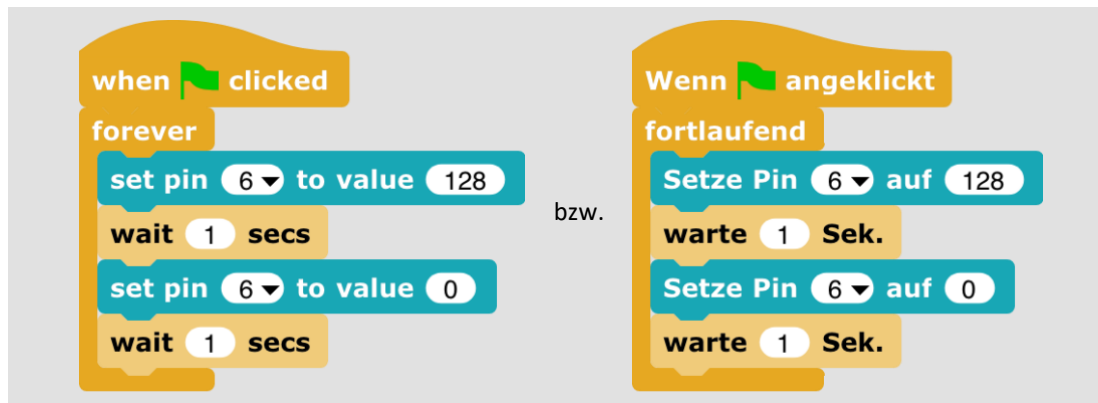


Hier wird der zu sendende Wert (aus: 0 oder 255; an: 1-254) und der Pin, an den der Summer angeschlossen ist, eingetragen. Je nach Wert variiert der ausgegebene Ton leicht in Klang und Lautstärke.



Aufgaben:

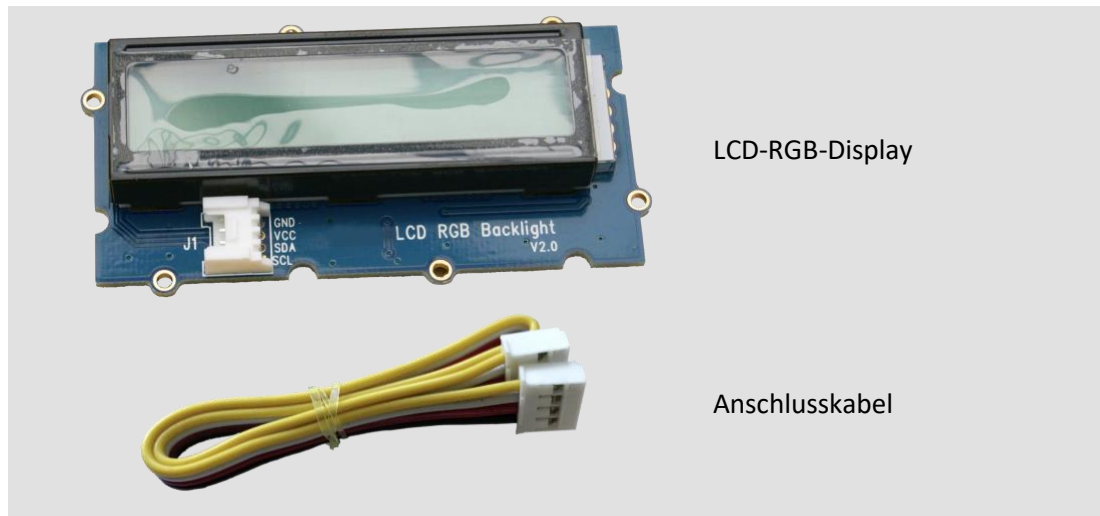
- 1) Schließe einen Summer an Pin D3 des Arduino an.
 - a) Lasse den Summer einen Ton erzeugen und schalte ihn anschließend wieder ab.
 - b) Was bewirkt das folgende Programm?



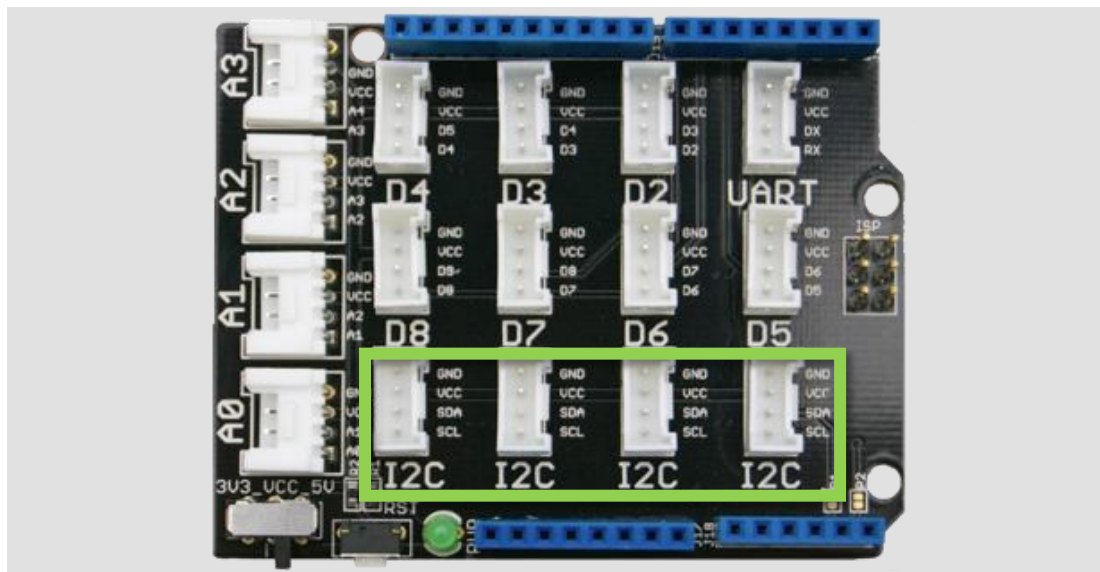
- 2) Erstelle ein Programm, das eine Polizeisirene simuliert.
- 3) Sende eine Morse-Nachricht an einen deiner Mitschüler.
- 4) Für welche weiteren Anwendungszwecke können Summer dienen?



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

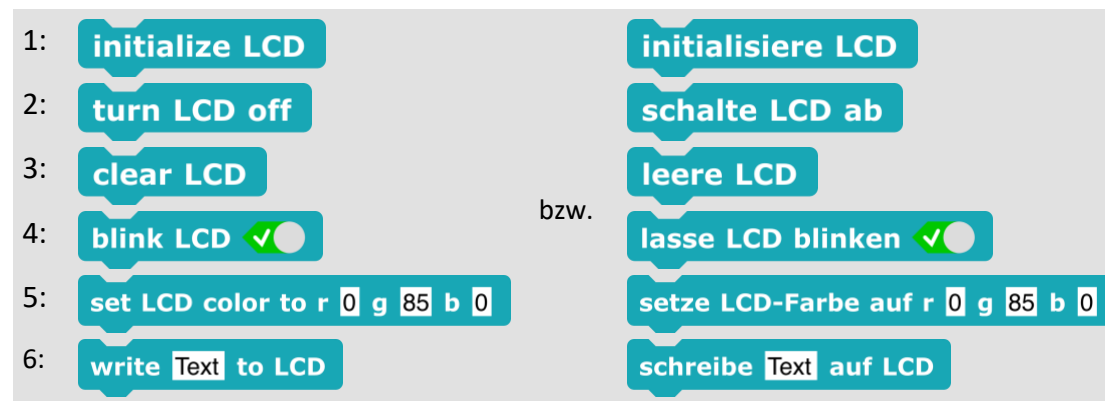


Das LCD-RGB-Display wird an einen I²C-Eingang angeschlossen (*Inter-Integrated Circuit, I2C* – im Bild grün umrandet).





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



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: Eingebener Text wird auf dem Display ausgegeben

Aufgaben:

- 1) Schließe das Display an den I²C -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?