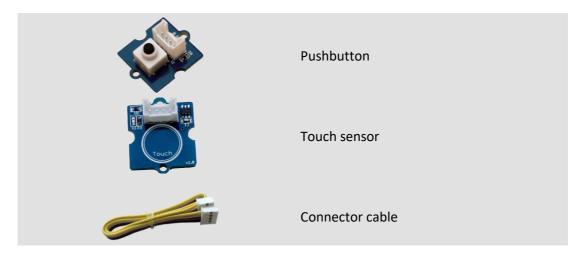
The following modules are pushbuttons and sensors whose signals can contain exactly two discrete values: 0 or 1, or false or true respectively:



Pushbuttons and digital sensors are connected to the digital inputs (digital input/output D2, D3, ..., D8—bordered green in the picture):



To read out the sensor values, which are always either *true* or *false*, Snap4Arduino uses the following block from the "Arduino" category:

digital reading 4 -

The number of the pin to which the sensor is connected is entered here.







Workstation 1 Pushbuttons and digital sensors (binary value)

To permanently display the values on stage, the "say" block from the "Looks" category can be used and enclosed with the "forever" block from the "Control" category:



Tasks:

- 1) Connect a pushbutton to pin D2 of the Arduino and display the current value on stage. Pay attention to the correct setting in the sensor block! When is this sensor's value *true* and when *false*? Also try the touch sensor and determine *true* and *false* conditions.
- 2) For which exemplary purposes can these modules be used?

Additional digital sensors are required for the following tasks:

- 3) Connect a toggle switch to pin D2 of the Arduino and display the current value on stage. Pay attention to the correct setting in the sensor block!
 - a) When is this sensor's value true and when false?
 - b) What is the difference to the button?
- 4) Try more digital sensors and determine their *true* and *false* conditions. Write them down clearly with a short explanation.
- 5) For which exemplary applications can the sensors be used?

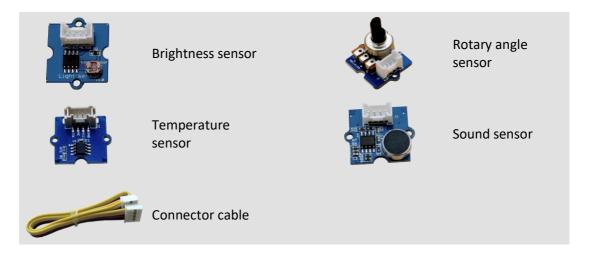




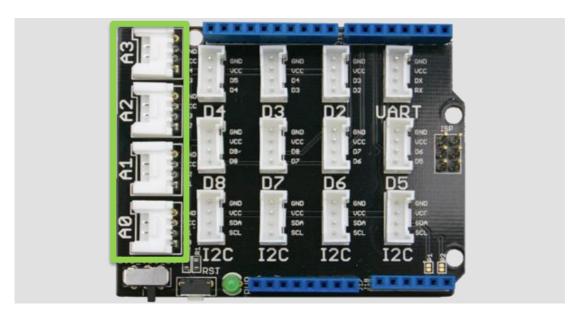




The following modules are analog sensors that can theoretically deliver any value between a minimum and a maximum. Since they are read out digitally, an analog-to-digital converter converts the analog input signals into a digital data stream with a limited (but often relatively high) number of discrete values, which can then be further processed or stored (here: 0-1023).



Analog sensors are connected to the analog inputs (analog input A0, A1, A2, A3—bordered green in the picture):



The following block from the "Arduino" category is used in Snap4Arduino to read the values, which are always between 0 and 1023:



The number of the pin to which the sensor is connected is entered here.









To permanently display the values on stage, the "say" block from the "Looks" category can be used and enclosed with the "forever" block from the "Control" category:



Tasks:

- 1) Connect a rotary angle sensor to pin A0 of the Arduino and display the current value on stage. Pay attention to the correct setting in the sensor block!
 - a) What is the smallest value that can be read via this sensor?
 - b) What is the highest value that can be read via this sensor?

Write them down in a table together with the values from tasks 2 and 3. Write a short description in the first column.

- 2) Use a brightness sensor and determine the sensor values:
 - a) in a very dark environment
 - b) in a very bright environment
 - c) in daylight
 - d) with the classroom lighting switched on
- 3) Try other analog sensors and determine their minimum and maximum values, as well as current values under room conditions.
- 4) For which exemplary purposes can the different sensors be used?



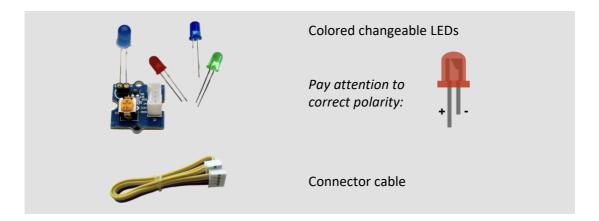




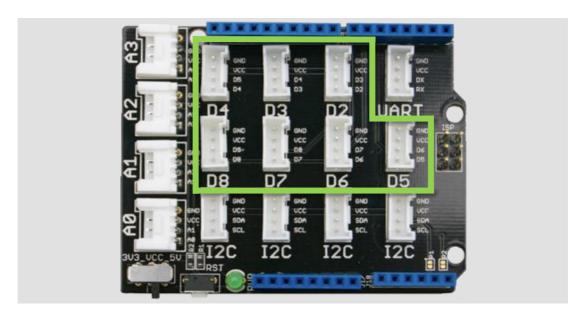


LED-Modules are actuators that can be controlled both with binary values and with pulse-width modulation (PWM):

LEDs



LED modules are connected to the digital outputs (digital input/output D2, D3, ..., D8—bordered green in the picture):



To turn LEDs on or off, the following block of the "Arduino" category is used in Snap4Arduino:



The pin to which the LED is connected and the status to be sent (*on* or *off*) are entered here.

To control the brightness of an LED, in Snap4Arduino the following block from the "Arduino" category is used:











The brightness value to be transmitted (between 0 and 255) and the pin to which the LED is connected are entered here.

LEDs

Note: such a pseudo analog control only works on special PWM pins (D3, D5, D6).

Tasks:

- 1) Connect an LED to pin D2 of your Arduino.
 - a) Let the LED light up and then switch it off again.
 - b) What does the following program do?

```
when clicked

forever

set digital pin 3 to 

wait 1 secs

set digital pin 3 to 

wait 1 secs
```

- 2) Write a program that alternately turns two LEDs on and off.
- 3) Write a program that makes an LED light up brighter every second.
- 4) For which exemplary applications can the LEDs be used?





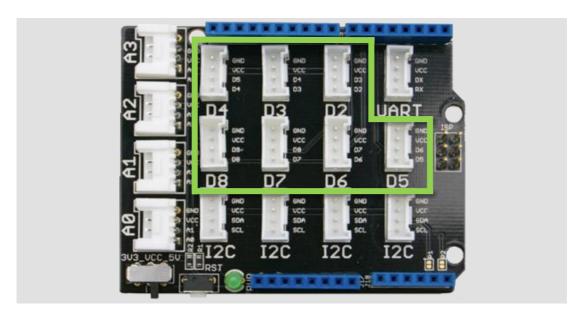




Servo motors are actuators:



Servo motors are connected to the digital outputs (digital input/output D2, D3, ..., D8—bordered green in the picture):



! Hint:

If the motors do not work properly, it is usually helpful to connect an external power source.









The following blocks from the "Arduino" category are used in Snap4Arduino to control the servo motors:

```
1: set servo → to 90 →

2: set servo → to clockwise →

set servo → to counter-clockwise →

set servo → to stopped →

3: set servo → to disconnected →
```

The pin to which the servo motor is connected and the value to be transmitted (see below) are entered here.

The following options are available for controlling the servo motors:

- 1: Servo motors (standard): angle between approx. 0° and 180°
- 2: Servo motors (continuous rotation (CR)—permanently rotating):
 - rotate clockwise (value approx. between 1000 and 1475)
 - rotate counterclockwise (value approx. between 1475 and 2000)
 - stop: value approx. 1475
- 3: disconnect the the servo motor

The exact values must be determined experimentally.

Tasks:

- 1) Connect a standard servo motor to pin D2 of the Arduino and set the angle to 90°.
- 2) Let the servo "wave".

A CR servo motor is required for the following tasks:

- 3) Connect a CR servo motor to pin D5 of the Arduino and determine the correct stop value.
- 4) How can the rotation speed be regulated?

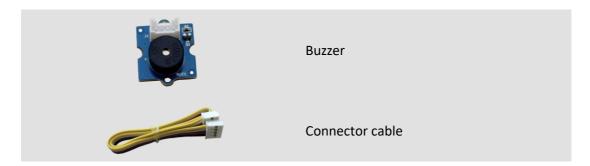




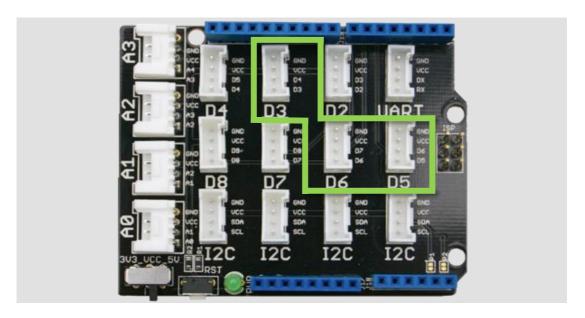




Buzzers are actuators which are controlled by pulse width modulation (PWM):



The buzzer is connected to one of the PWM pins of the digital outputs (*digital input/output D3, D5, D6*—bordered green in the picture):



The following blocks from the "Arduino" category are used in Snap4Arduino to control buzzers:



The value to be transmitted (off: 0 or 255; on: 1-254) and the pin to which the buzzer is connected are entered here. Depending on the value, the tone output varies slightly in sound and volume.







V2019.09.26



Tasks:

- 1) Connect a buzzer to pin D3 of the Arduino.
 - a) Let the buzzer produce a sound and then turn it off again.
 - b) What does the following program do?

```
when clicked

forever

set pin 6 → to value 128

wait 1 secs

set pin 6 → to value 0

wait 1 secs
```

- 2) Create a program that simulates a police siren.
- 3) Send a Morse message to one of your classmates.
- 4) For what other applications can buzzers be used?

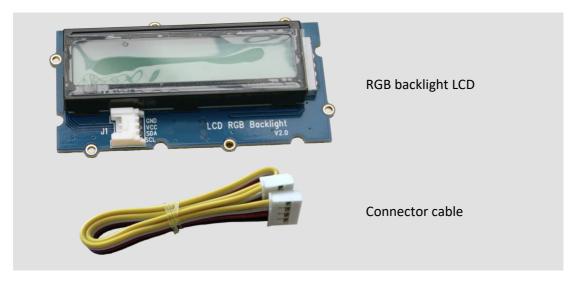




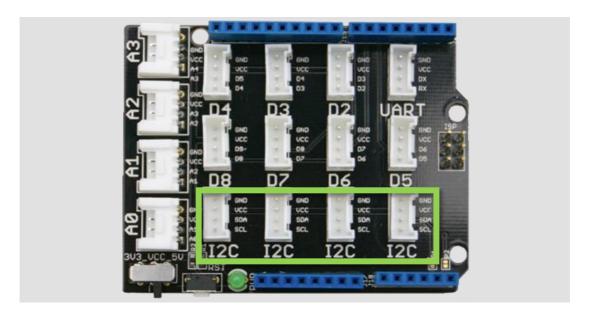




The following module is an RGB backlight LCD:



The RGB backlight LCD is connected to an I^2C -input (inter integrated circuit, I2C—bordered green in the picture).











The following blocks from the "Arduino" category are used in Snap4Arduino to control the display:

```
1: initialize LCD
2: turn LCD off
3: clear LCD
4: blink LCD ✓
5: set LCD color to r 0 g 85 b 0
6: write Text to LCD
```

Unlike most other modules, it is not necessary to enter a pin when using the display as it is automatically recognized.

The blocks provide certain display functions:

- 1: Display is initialized (switched on and set to initial state)
- 2: Display is switched off
- 3: Display is emptied (text is deleted)
- 4: Blink method is started or stopped
- 5: Display background color is set with RGB values
- 6: Entered text is shown on the display.

Tasks:

- 1) Connect the display to the I²C port and have the text "Hello World" appear on the display.
- 2) Change the backlight color.
 - a) What is the meaning of the values r, g and b?
 - b) Which value triples produce yellow, violet and turquoise light?
- 3) Test the maximum number of characters that can be shown on the display. What happens if you exceed this number?
- 4) For what exemplary applications can the display be used?





