

Snap4Arduino



Snap4Arduino ist eine Abwandlung von *Snap!*, die die Kommunikation und das Arbeiten mit einem Arduino ermöglicht.

Der Download der Software ist unter <http://snap4arduino.org/> problemlos für verschiedene Betriebssysteme möglich.

Download

There are currently versions for GNU/Linux, MacOSX, Microsoft Windows and Chromebook available.

The project and all its components (including Snap!) are registered under public free software licenses (AGPLV3 and MIT), so you can download the sources and pretty much do whatever you want with them!

Please download the version that matches your operating system. Current version: 1.2.2, released on 22/12/2016



Nach Installation ist es unter **Einstellungen** → **Sprache** möglich, die Sprache auf Deutsch zu ändern.



Vorbereitung des Arduino

Um mit *Snap4Arduino* einen Arduino programmieren zu können, muss auf diesem die *Standard Firmata* installiert sein (siehe Anleitung: *Wie lernt mein Arduino JAVA?*).

Des Weiteren muss dieser per USB-Kabel mit dem PC verbunden sein. Über den Baustein **Mit Arduino verbinden** in der Kategorie **Arduino** kann der Benutzer manuell eine Verbindung zum Arduino herstellen.

Das Trennen der Verbindung ist über den Block **Verbindung zu Arduino trennen** manuell möglich.



Tipp: Ab und zu trennt Snap4Arduino eigenständig die Verbindung zum Arduino und gibt eine entsprechende Meldung aus. Vor Test des Codes muss dann die Verbindung wieder erneuert werden.

Snap4Arduino kommuniziert über die Blöcke der Kategorie „**Arduino**“ mit dem Arduino. Zur Erstellung eines Programms werden außerdem hauptsächlich Blöcke aus den Kategorien „**Steuerung**“, „**Operatoren**“ und „**Variablen**“ benötigt. Die Blöcke der anderen Kategorien dienen hauptsächlich dazu, eine Figur auf der Bühne zu steuern.

Die Arduino-Blöcke

Es gibt sieben verschiedene **Arduino-Blöcke**:

verbinde Arduino an Port

Dieser Baustein ermöglicht die Verbindung zu einem Arduino. Die Angabe des entsprechenden Ports erfolgt per Hand.

Diese Art der Verbindung funktioniert erfahrungsgemäß leider nicht wie erwartet. Die manuelle Verbindung über **Mit Arduino verbinden** ist dem vorzuziehen.

verbindung zu Arduino trennen

Dieser Baustein sollte die Verbindung zum Arduino trennen. Leider funktioniert er bislang nicht wie erwartet. Eine manuelle Trennung über **Verbindung zu Arduino trennen** ist dem vorzuziehen.

Setze Servo auf

Dieser Block dient dazu, einem angeschlossenen **Servo** einen Wert zuzuweisen. Dazu muss der Servo am Arduino angeschlossen und die entsprechende Nummer des Pins eingetragen werden. Im rechten Auswahlménü wird dann der Ausgabewert gewählt:

Wir stellen über **Winkel (0-180)** einen konkreten Winkel für den Servo eingestellt. Dazu wird ein Wert zwischen 0 und 180 (entspricht 0° bis 180°) angegeben.

Setze digitalen Pin auf

Einem angegebenen digitalen Ausgang wird per Klick ein Wahrheitswert (*true* oder *false*) zugewiesen.

Setze Pin auf

Einem angegebenen analogem Ausgang wird ein Wert zugewiesen.

lies digitalen Pin

Mit diesem Block können analog angeschlossene Sensoren (Pins **A0** bis **A5** am Arduino) ausgelesen werden. Die am Pin anliegenden Spannungswerte werden dabei in *Zahlenwerte zwischen 0 und 1023* umgerechnet.

lies analogen Pin

Mit diesem Block können digital angeschlossene Sensoren ausgelesen werden. Die am Pin anliegenden Spannungswerte werden dabei in Wahrheitswerte (wahr oder falsch) umgerechnet.

Tipp: Per Klick auf einen der beiden letztgenannten Blöcke kann der aktuelle Wert des gewählten Pins angezeigt werden.

Aufgaben

Im Folgenden wird eine LED verwendet. Diese hat ein kürzeres (Minus) und ein längeres (Plus) Ende.

Hinweis: Bei der Verwendung einer LED sollte immer auch ein Widerstand ($200\Omega - 500\Omega$) verwendet werden.

Aufgabe 1 – Eine LED zum Leuchten bringen.

Stecke die LED und den Widerstand auf dem Steckbrett in Reihe. Verbinde den Pluspol mit der positiven Versorgungsspannung (V-Pin) und den Minuspol mit GND (G-Pin).

Die LED sollte nun leuchten.

Aufgabe 2 – Eine LED steuern.

Im Folgenden soll die LED über ein einfaches Programm derart gesteuert werden, dass sie blinkt.

Verbinde dazu den Pluspol nicht mehr mit dem V-Pin, sondern mit dem Steuer-Pin (S-Pin). Dieser ist für die Steuerung der LED zuständig und regelt, ob Spannung anliegt oder nicht.

Die LED leuchtet nun, wenn man den entsprechenden digitalen Pin auf wahr setzt. Analog leuchtet die LED nicht, wenn dieser auf falsch gesetzt wird.

Setze digitalen Pin 8 auf

Ein Snap-Programm startet idealerweise immer mit einem Kopf-Block. Anschließend folgt das eigentliche Programm.

Wenn angeklickt

Entwickle ein Programm, das die LED blinken lässt.

Tipp: Warte beim Umschalten zwischen Leuchten und nicht Leuchten immer eine Sekunde. Ansonsten passiert das An- und Ausschalten zu schnell und ist für den Benutzer nicht sichtbar.

warte 1 Sek.

Aufgabe 3 – Eine LED abhängig vom Lichtsensor an-/ausschalten.

Im Folgenden soll die LED nun abhängig vom Lichtsensor leuchten.

Schließe dazu als erstes den Lichtsensor an. (siehe Anleitung Lichtsensor)

1. Probiere aus, welchen Wert der entsprechende digitale Pin bei Dunkel-/Helligkeit liefert.
2. Lass abhängig von diesem Wert die LED leuchte bzw. nicht leuchten.

lies digitalen Pin

Setze digitalen Pin 8 auf