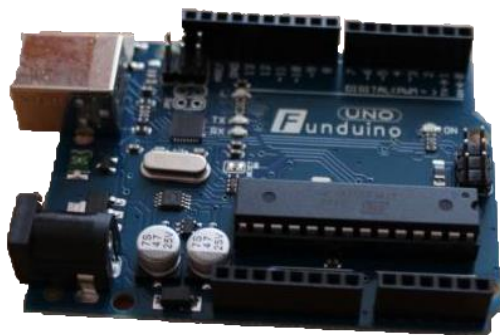




# Mikrocontroller programmieren mit Arduino und Scratch

Rita Kadletz



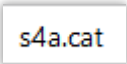
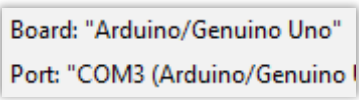
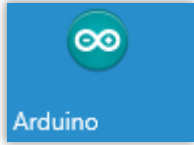



# Inhaltsverzeichnis

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Programmierung</b>	<b>3</b>
2.1	<i>Eingebaute Led zum Blinken bringen / Schleife / digital Output</i>	3
2.2	<i>LEDs an den digitalen Pins anschließen und zum Blinken bringen</i>	3
2.3	<i>Piezo-Summer einsetzen</i>	4
2.4	<i>Pulsweitenmodulation / Variable / analog Output</i>	5
2.5	<i>Input: einen Taster anschließen / Verzweigung / digital Input</i>	6
2.6	<i>Phototransistor / analog Input</i>	7
2.7	<i>RGB-LED in den Grundfarben</i>	8
2.8	<i>Potentiometer (Drehregler)</i>	9
2.9	<i>Servo ansteuern</i>	10

Die Grafiken, die den Aufbau der Schaltungen darstellen, wurden mit fritzing erstellt.

# 1 Installation

1. S4A installieren		4. Firmware öffnen	
2. Firmware herunterladen		5. Auf Port und Board achten	
3. Arduino-IDE öffnen		6. Auf den Arduino laden	

# 2 Programmierung

## 2.1 Eingebaute Led zum Blinken bringen / Schleife / digital Output (Arduino / USB-Kabel)

Auf dem Arduino ist an Pin 13 bereits eine LED am Board für Testzwecke eingebaut. Wir werden programmieren, dass die LED an Pin 13 blinkt.



### 2.1.1 Aufgaben

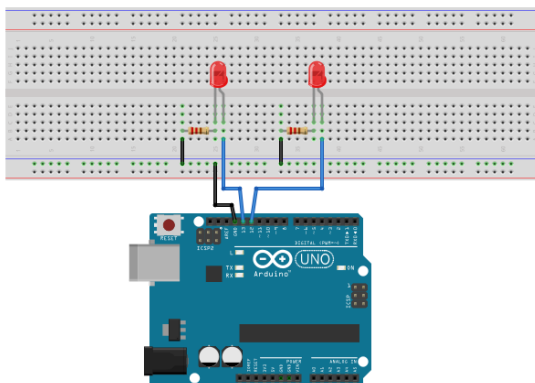
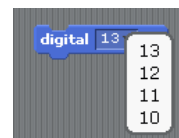
**BL1** Programmiere LED13 so, dass die Leuchtdiode auf dem Board abwechselnd 2 Sekunden leuchtet und dann eine halbe Sekunde nicht leuchtet.

## 2.2 LEDs an den digitalen Pins anschließen und zum Blinken bringen

Wechselblinker (Arduino / 2 blaue LED / 2 Widerstände mit 100 Ohm / Breadboard / Kabel)

In S4A sind die digitalen Pins 10, 11, 12 und 13 als Ausgangspins vordefiniert.

Hier können Leuchtdioden,... angeschlossen werden.



### 2.2.1 Aufgaben Blinker

**WE1** Schalte die beiden Leuchtdioden so, dass sie gleichzeitig zwei Sekunden eingeschaltet und wieder zwei Sekunden ausgeschaltet werden.

**WE2** Schließe zusätzlich eine Leuchtdiode bei Pin11 an und versuche die Leuchtdioden so zu steuern, dass alle drei gleichzeitig eine Sekunde leuchten und dann eine Sekunde ausgeschaltet sind.

**WE3** Ändere das Programm so, dass die drei LEDs der Reihe nach je eine halbe Sekunde leuchten.

**WE4** Schalte die drei Leuchtdioden so dass sie der Reihe nach eingeschaltet werden (erst 1, dann 1+2, dann 1+2+3) und dann der Reihe nach wieder ausgeschaltet werden (1+2+3, dann nur mehr 1+2 und dann nur mehr 1 und am Ende leuchtet keine mehr). Zeitabstand ist beliebig.

**WE5** Verwende drei verschiedenfarbige Leuchtdioden und versuche eine Ampelsteuerung zu simulieren. (grün leuchtet 2 Sekunden, blinkt dann 4 mal je eine halbe Sekunde, gelb leuchtet eine Sekunde, rot leuchtet 2 Sekunden alleine, dann wird gelb eine halbe Sekunde dazu geschaltet / das grüne Blinken ist typisch für Österreich und kann auch weggelassen werden)

## 2.3 Piezo-Summer einsetzen

(Arduino / 1 Piezo-Speaker / Breadboard / Kabel)

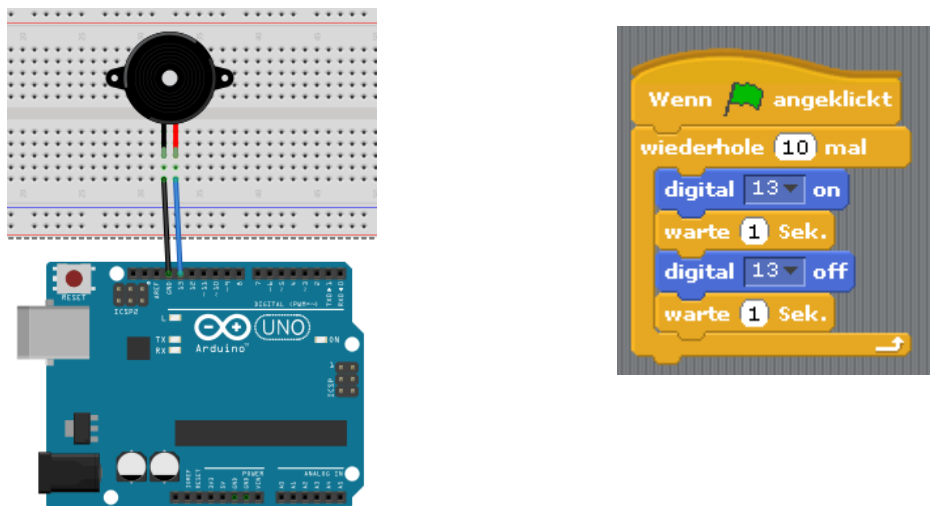


Abbildung 1: Piezo-Summer (erstellt mit fritzing)

### 2.3.1 Aufgabe Summer

**PI1** Bei Pin12 soll noch eine blaue Leuchtdiode (Vorwiderstand 100 oder 220 Ohm) angeschlossen werden. Jedes Mal wenn der Summer ertönt soll gleichzeitig die Leuchtdiode leuchten.

## 2.4 Pulsweitenmodulation / Variable / analog Output

(Arduino / eine LED (blau) / Ein Widerstand mit 100 Ohm / Breadboard / Kabel)

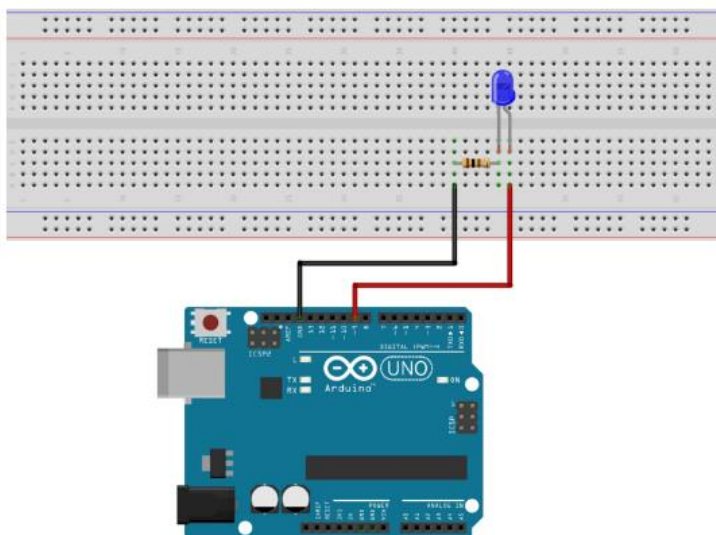
Ziel: eine LED soll heller und dunkler werden (englisch: fade)

Der Arduino kennt an seinen Ausgängen nur „5 Volt an“ oder „5V aus“. Um die Helligkeit einer LED zu variieren, müsste man die Spannung jedoch ändern können. Zum Beispiel 5V wenn die LED hell leuchtet, 4 Volt, wenn sie etwas dunkler leuchtet usw. Das funktioniert mit digitalen Pins aber nicht. Es gibt jedoch eine Möglichkeit, die Pulsweitenmodulation (PWM genannt). Die PWM lässt die 5V Spannung pulsieren. Die Spannung wird also im Millisekundenbereich ein und ausgeschaltet. Bei einer hohen PWM, also gegen 255, liegt das 5V Signal nahezu durchgehend am jeweiligen Pin an. Bei einer geringen PWM ist das 5V Signal kaum noch vorhanden.

In S4A ist PWM durch die Funktion [analog 9 value 255] vordefiniert. Nur die Pins 5, 6 und 9 können damit angesteuert werden.



### 2.4.1 Aufbau der Schaltung:



Das folgende Skript bewirkt, dass das PWM-Signal langsam von 0 auf 255 erhöht wird. Die Leuchtdiode wird immer heller. Gelöst wird das mit einer Variable mit dem Namen helligkeit.



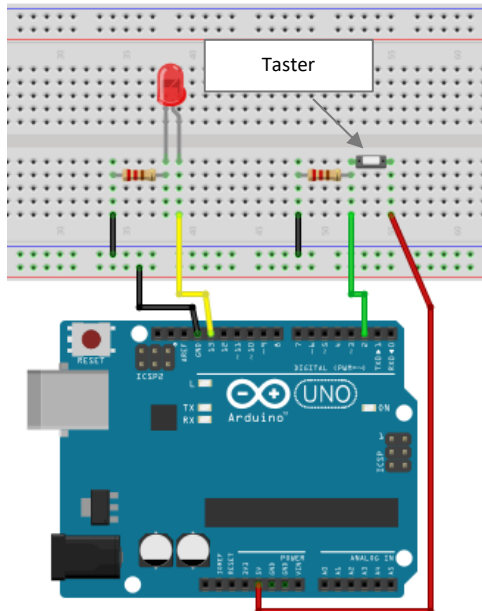
### 2.4.2 Aufgaben

**PWM1** Die Leuchtdiode soll langsam heller werden und dann langsam wieder dunkler werden.

## 2.5 Input: einen Taster anschließen / Verzweigung / digital Input

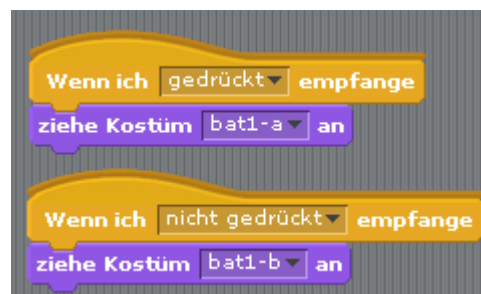
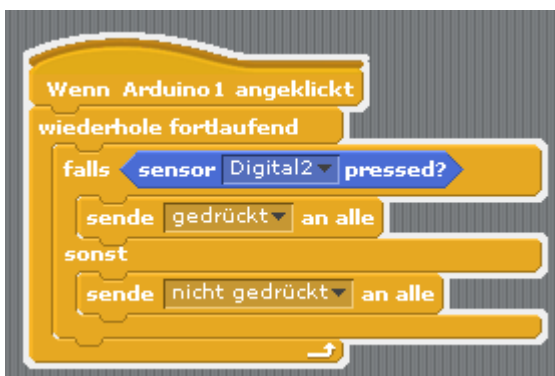
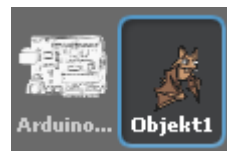
(Arduino / Widerstand 10kΩ / Breadboard / Kabel / Taster / LED / Widerstand 220 Ohm)

Ground (mit Widerstand) und Pin2 an einer Seite anschließen, 5V auf der anderen Seite. Dazu kommt eine Leuchtdiode bei Pin 13 mit einem Widerstand von 220 Ohm.



### 2.5.1 Andere Figuren steuern

Zusätzlich wird eine andere Figur mit zwei Kostümen eingefügt.



### 2.5.2 Aufgabe

**Ta1** (Schwierig!) Wenn der Taster gedrückt wird, soll eine Leuchtdiode leuchten, beim nächsten Mal drücken soll sie wieder ausschalten. Eine Variable wird erstellt. Ist die LED eingeschaltet, so wird sie auf 1 gesetzt, ist sie ausgeschaltet, wird sie auf 0 gesetzt.

Mögliche Lösung zur Aufgabe:



## 2.6 Phototransistor / analog Input

s4a.cat (Arduino / Widerstand 10kΩ / Breadboard / Kabel / Phototransistor)

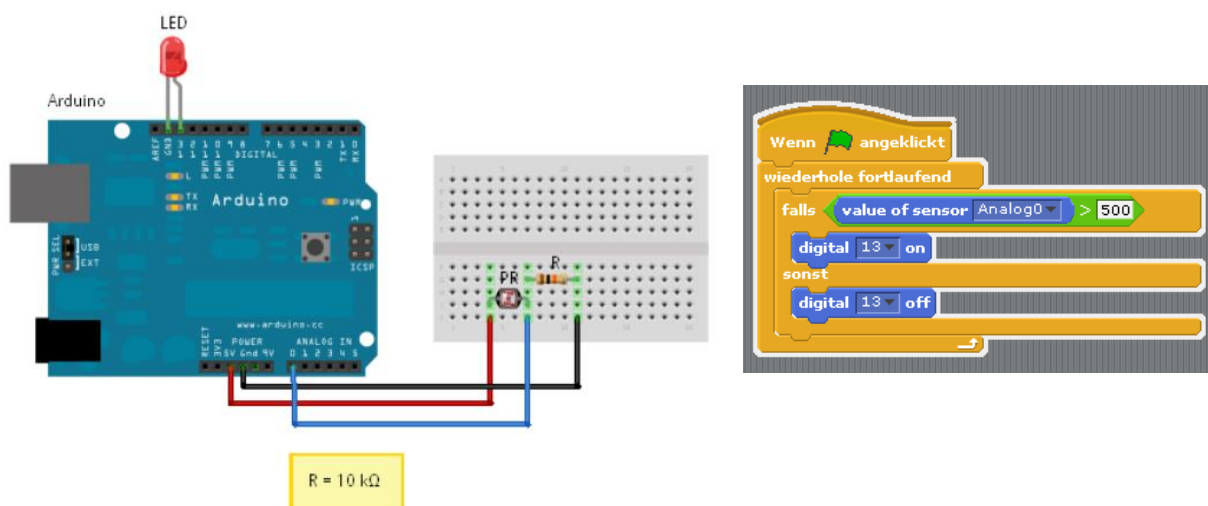


Abbildung: Phototransistor (Quelle s4a.cat)

Wenn der Phototransistor abgedunkelt wird, erlischt die Leuchtdiode.

### 2.6.1 Aufgabe

**Ph1** Wenn's dunkel wird, soll die Leuchtdiode aufleuchten und bei Helligkeit ausgeschaltet sein.

**Ph1** Bei Dunkelheit soll die Leuchtdiode ausgeschaltet sein, bei mittlerer Helligkeit soll sie blinken und bei Helligkeit soll sie dauerhaft leuchten.

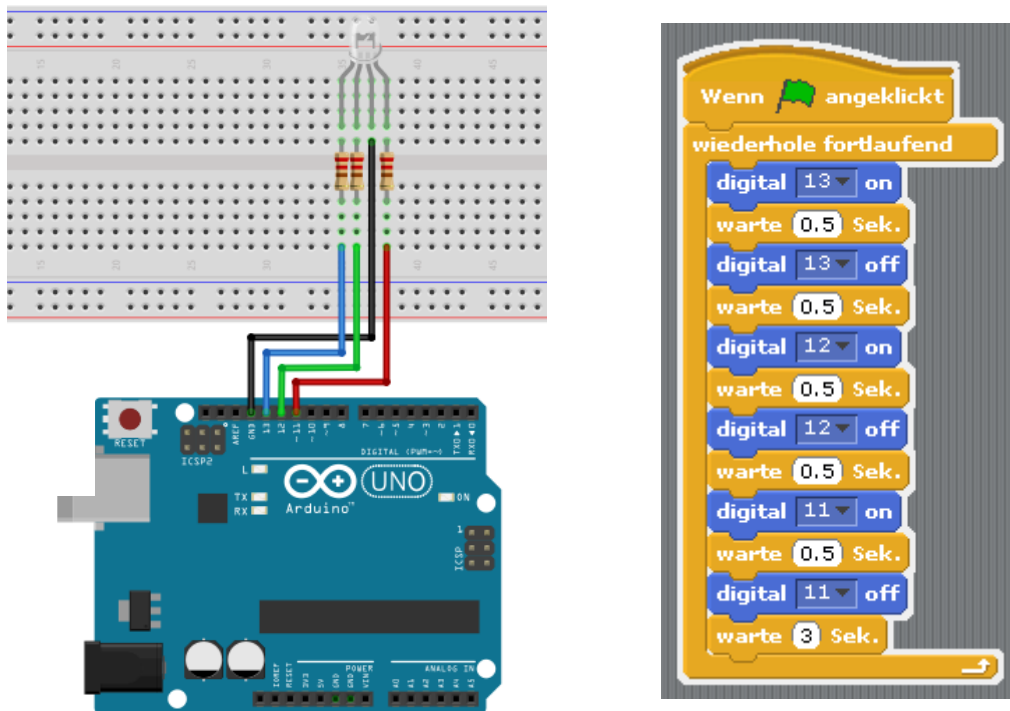
## 2.7 RGB-LED in den Grundfarben

(Arduino-Board / eine RGB-LED / drei Widerstände mit je 200 Ohm / Breadboard / Kabel)

Eine RGB-LED ist eine LED die in verschiedenen Farben leuchten kann. Sie besteht aus drei einzeln ansteuerbaren LEDs, die in den drei Farben „Rot“, „Grün“ und „Blau“ leuchten. Deswegen hat eine RGB-LED auch so viele Beine, nämlich genau vier.

Bei unseren RGB-LEDs ist das längste Beinchen „-“ und die drei kürzeren Beinchen werden über „+“ (Spannung) angesteuert.

Durch eine Mischung der Farben können noch viele weitere Farben erzeugt werden



**RGB1** Die LED soll nur rot blinken.

**RGB2** Die LED soll nur blau blinken.

**RGB3** Die LED soll nur grün blinken.

**RGB4** Rot und Grün sollen gleichzeitig eingeschaltet und wieder gleichzeitig ausgeschaltet werden.

**RGB5** Blau und Grün sollen gleichzeitig eingeschaltet und wieder gleichzeitig ausgeschaltet werden.

**RGB6** Rot und Blau sollen gleichzeitig eingeschaltet und wieder gleichzeitig ausgeschaltet werden.

**RGB7** Simuliere eine Ampelsteuerung.

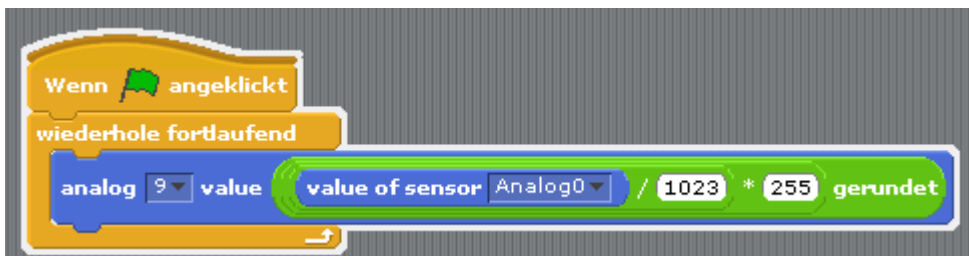
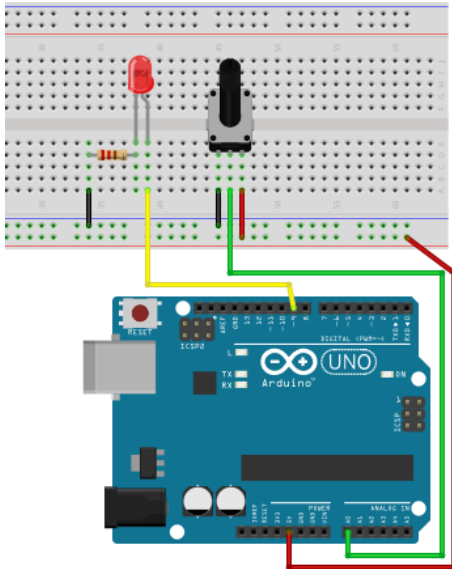


## 2.8 Potentiometer (Drehregler)

funduino.de Anleitungen Nr. 09 Potentiometer (Arduino / Potentiometer / Breadboard / Kabel)

Ein Potentiometer ist ein Widerstand, dessen Wert man durch einen Drehregler verändern kann. Er hat drei Anschlüsse: linker Fuß bei GND, mittlerer Fuß bei einem Analogen Eingangspin, rechter Fuß bei 5V.

Wir lesen den Wert des Potentiometers aus und verwenden ihn für die Helligkeit der LED. Diese variiert durch die PWM im Bereich von 0 bis 255.



### 2.8.1 Aufgabe Potentiometer

**Po1** (Schwierig!) Die Leuchtdiode soll alle 5 Sekunden blinken und schneller werden, wenn der Potentiometer den Widerstand erhöht.

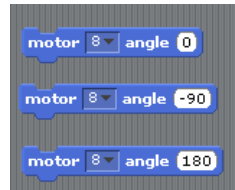
**Po2** Bei einer Überschreitung eines bestimmten Wertes soll ein Signal ertönen.

## 2.9 Servo ansteuern

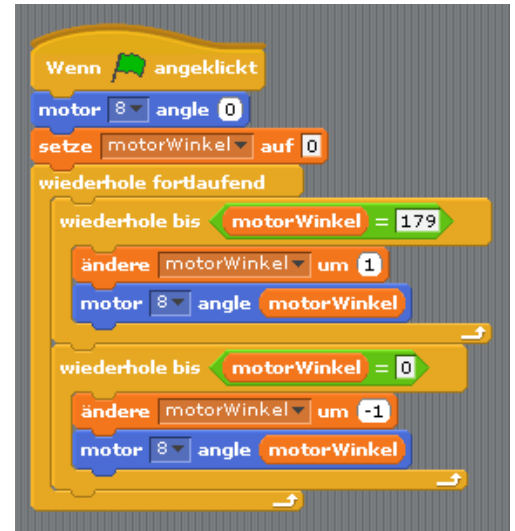
funduino.de Anleitungen Nr. 13 Servo ansteuern (Arduino / Servomotor / Kabel)

Drei Anschlüsse: braun bei GND, rot bei 5V, orange z.B. bei PIN 8

Durch Doppelklick auf die Schaltflächen können die Positionen getestet werden:



Mit diesem Skript dreht sich der Motor im Bereich von 0 bis 179 Grad hin und her.



Nun wird noch ein Potentiometer angeschlossen: Linker Fuß bei GND, mittlerer Fuß z.B. bei A0 (Analog Pin 0) und rechter Fuß bei 5 V. Der Wert des Potentiometers (von 0 bis 1023) wird in Grad umgerechnet. Das Runden ist wichtig, da die Funktion *motor 8 angle* nicht mit Kommazahlen funktioniert

