

**Teil 2**

# Lektion

# 12

**Analoger Joystick**

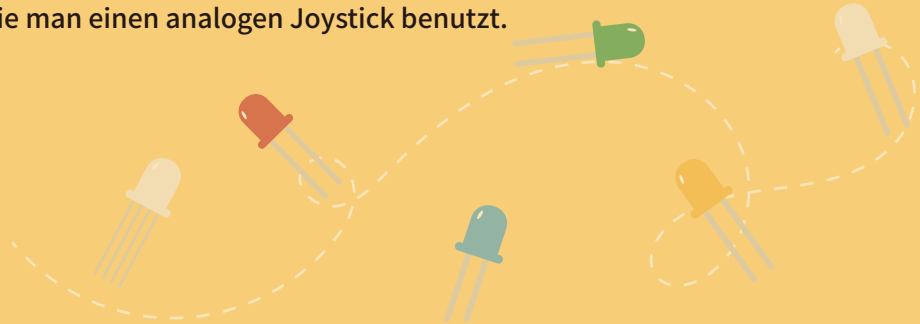
## Übersicht

Analoge Joysticks eignen sich gut, um Ihre Projekte präzise steuern zu können.

In dieser Lektion lernen Sie, wie man einen analogen Joystick benutzt.

### Benötigte Bauteile:

- (1) x Elegoo Uno R3
- (1) x Joystick Modul
- (5) x W-M Kabel (Weiblich zu Männlich DuPont Jumper Kabel)

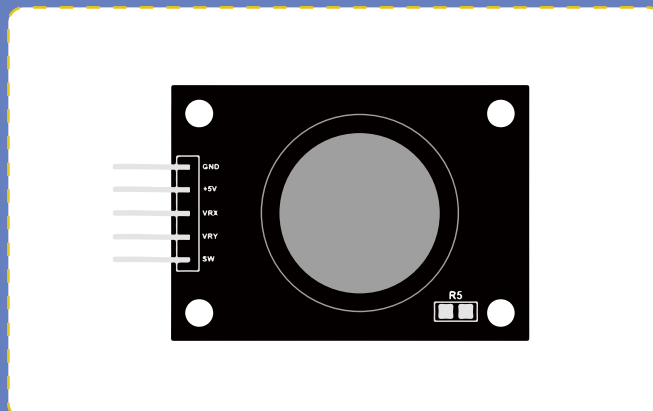


## Einführung in die Komponenten

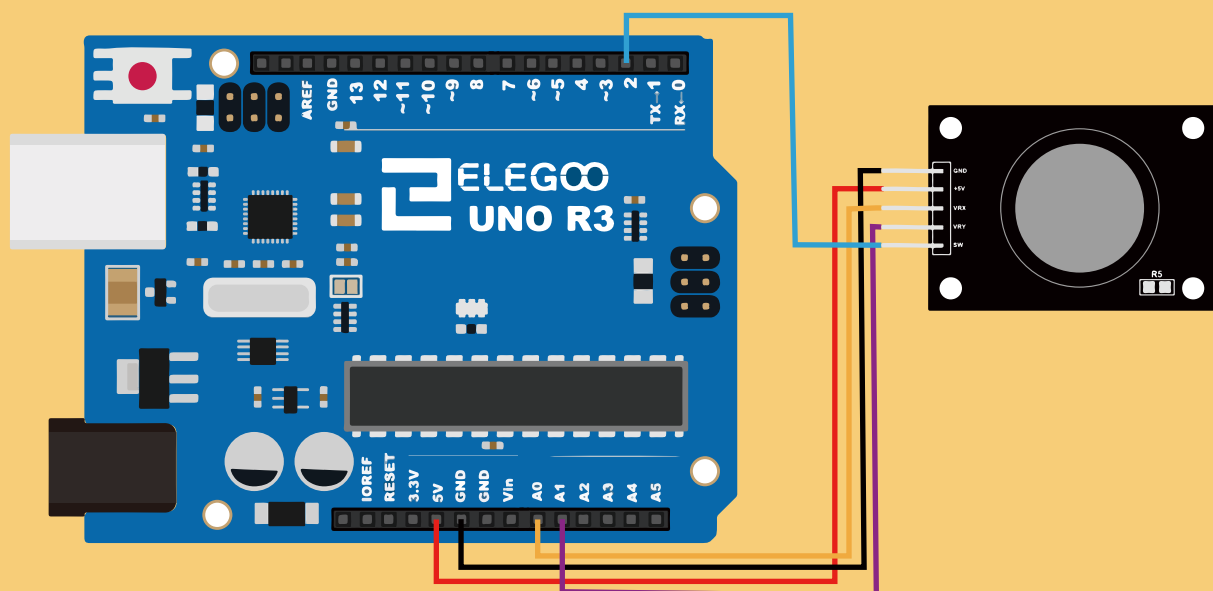
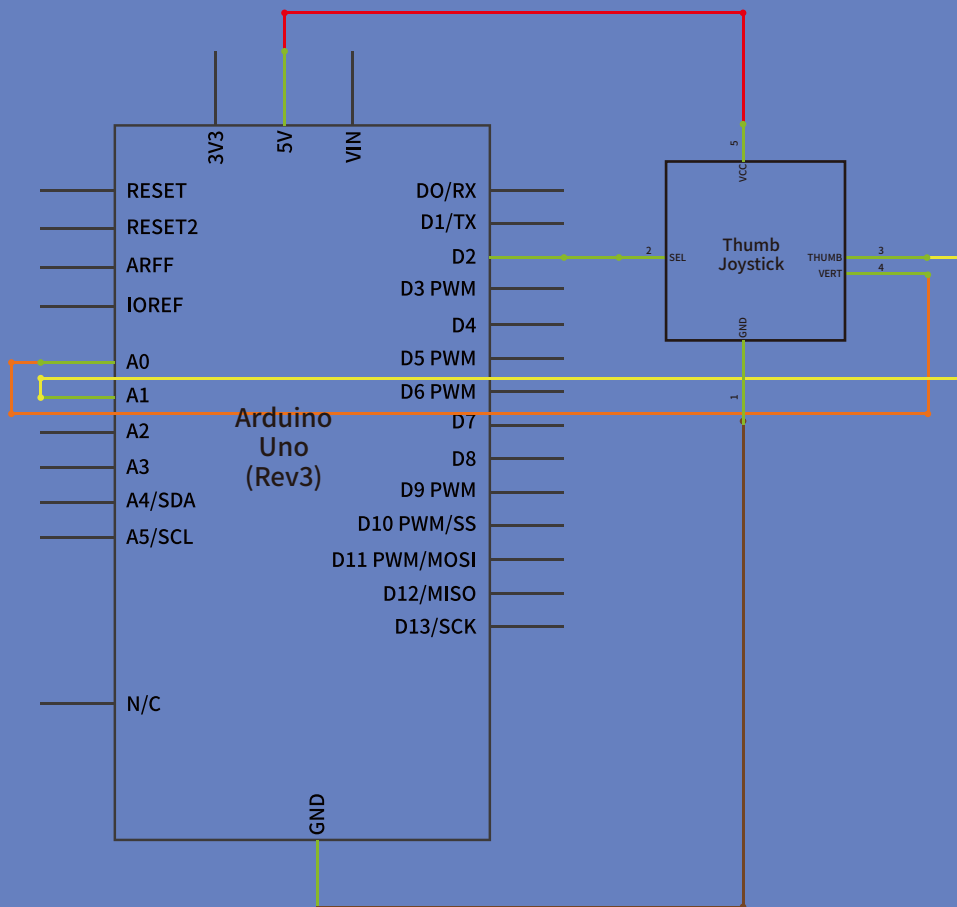
### Joystick

**Das** Modul hat 5 Pins: VCC (Stromversorgung), GND, X, Y und Key. Beachten Sie, dass die Bezeichnungen bei Ihrem Modell leicht abweichen können, je nachdem wo das Modul gekauft wurde. Der Joystick ist analog und erlaubt eine viel präzisere Steuerung als bei einfachen mechanischen Schaltern. Zusätzlich zur horizontalen und vertikalen Steuerung, können Sie den Joystick herunterdrücken, wobei der Joystick als Schalter fungiert.

**Wir** müssen die analogen Pins des UNO Boards nutzen, um die analogen Werte des Joysticks auf der X- und Y-Achse empfangen zu können und einen digitalen Pin für den Schalter. Der Key-Pin ist im normalen Zustand frei und wird bei Drücken des Joysticks mit GND verbunden. Um zuverlässige Werte zu erhalten, muss der Key-Pin mittels eines Pullup-Widerstandes mit dem Board verbunden werden. Dabei können wir die eingebauten Pullup-Widerstände des UNO Boards benutzen. Für eine Anleitung, wie man Pins des UNO R3 Boards als Eingang mit Pullup-Widerstand benutzt, schauen Sie sich den Code-Teil von Lektion 5 erneut an.



## Verbindungsschema



Wir brauchen fünf Verbindungen zum Joystick:

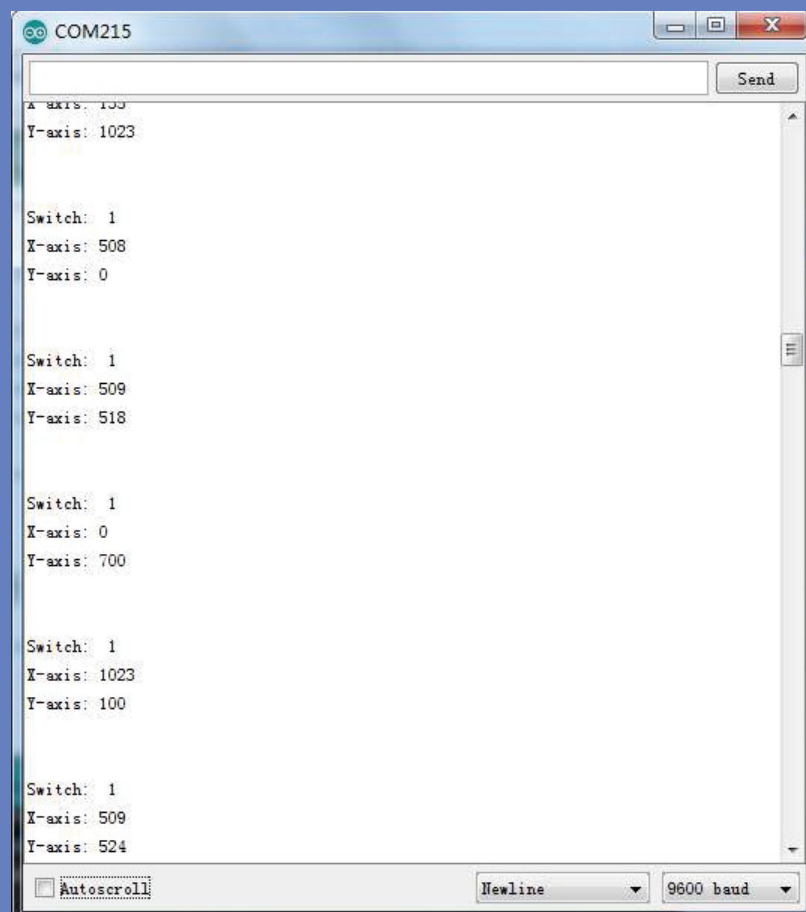
Key, Y, X, VCC und GND.

X und Y sind analog und Key ist digital. Wenn Sie den Schalter nicht benutzen, können Sie den Key-Pin weglassen und benötigen nur die restlichen vier Anschlüsse.

**Schaltplan**

## Code

- Nach dem Verbinden der Komponenten öffnen Sie bitte den Sketch im Code-Ordner unter „Analog\_Joystick“ und laden ihn auf Ihr UNO Board hoch. Bei Fragen zum Hochladen eines Sketches schauen Sie sich bitte Lektion 5 in Teil 1 noch einmal an.
  - Analoge Joystick sind im Grunde Potentiometer, daher geben sie analoge Werte zurück.
  - Wenn sich der Stick in der Mitte befindet (Standardposition), sollte er am X- und am Y-Pin einen Wert von 512 ausgeben. Der Wertebereich geht von 0 – 1024.
  - Auf der X-Achse ist ganz links = 0 und ganz rechts = 1024, auf der Y-Achse ist ganz unten = 0 und ganz oben = 1024.
- Öffnen Sie den Seriellen Monitor:
- Klicken Sie auf das Symbol des Seriellen Monitors, um ihn zu öffnen. Die grundlegenden Informationen zum Seriellen Monitor haben Sie in Lektion 4 in Teil 2 kennengelernt.



```
Serial.print("X-axis: ");
Serial.print(analogRead(X_pin));
```

## analogRead()

### [Analog I/O]

#### Description

Liest den Wert vom angegebenen analogen Pin. Arduino-Devices enthalten einen 10-Bit-Analog-Digital-Wandler mit mehreren Kanälen. Dies bedeutet, dass Eingangsspannungen zwischen 0 und der Betriebsspannung (5 V oder 3,3 V) in ganzzahlige Werte zwischen 0 und 1023 abgebildet werden. Auf einem Arduino UNO ergibt dies beispielsweise eine Auflösung zwischen Messwerten von: 5 Volt / 1024 Einheiten oder 0,0049 Volt (4,9 mV) pro Einheit. In der folgenden Tabelle finden Sie die verwendbaren PINs.

Der Eingangsbereich kann mit der `analogReference()` geändert werden, während die Auflösung (nur für Zero-, Due- und MKR-Devices) mit `analogReadResolution()` geändert werden kann.

Auf ATmega-basierten Karten (UNO, Nano, Mini, Mega) dauert das Lesen eines analogen Eingang-Signals etwa 100 Mikrosekunden (0,0001 s), sodass die maximale Leseratte etwa 10.000 Mal pro Sekunde beträgt.

\* A0 bis A5 sind auf der Platine gekennzeichnet, A6 bis A11 sind jeweils an den Pins 4, 6, 8, 9, 10 und 12 erhältlich

\*\* Die Standardauflösung für `analogRead()` für diese Karten beträgt aus Kompatibilitätsgründen 10 Bit. Sie müssen `analogReadResolution()` verwenden, um es auf 12 Bit zu ändern.

BOARD	BOARD OPERATING VOLTAGE	USABLE PINS	MAX RESOLUTION
Uno	5 Volts	A0 to A5	10 bits
Mini, Nano	5 Volts	A0 to A7	10 bits
Mega, Mega2560, MegaADK	5 Volts	A0 to A14	10 bits
Micro	5 Volts	A0 to A11*	10 bits
Leonardo	5 Volts	A0 to A11*	10 bits
Zero	3.3 Volts	A0 to A5	12 bits**
Due	3.3 Volts	A0 to A11	12 bits**
MKR Family boards	3.3 Volts	A0 to A6	12 bits**

#### Syntax

```
analogRead(pin)
```

#### Parameters

**pin:** Der Name des analogen Eingangs-PINs, von welchem gelesen werden soll (A0 bis A5 auf den meisten Devices; A0 bis A6 auf Arduino MKR, A0 bis A7 auf dem Arduino Mini und Nano, A0 bis A15 auf dem Arduino Mega).

#### Returns

Den analogen Wert am PIN. Dabei ist der Wert auf die Auflösung des Analog-Digital-Wandlers beschränkt (0-1023 für 10 Bit oder 0-4095 für 12 Bit). Datentyp: Integer (int).