



🔍 Suche...

Funduino Education ▾ 3D-Druck ▾ Elektronische Module ▾ Sensoren ▾ B:

Der Rotary Encoder KY-040

Inhaltsverzeichnis



1. Für dieses Projekt benötigst du
2. Den Rotary Encoder KY-040 an einem Arduino Mikrocontroller verwenden
 - 2.1. Wie funktioniert der Encoder?
 - 2.2. Pinbelegung
 - 2.3. Beispielsketch
3. Funduino – Dein Onlineshop für Mikroelektronik

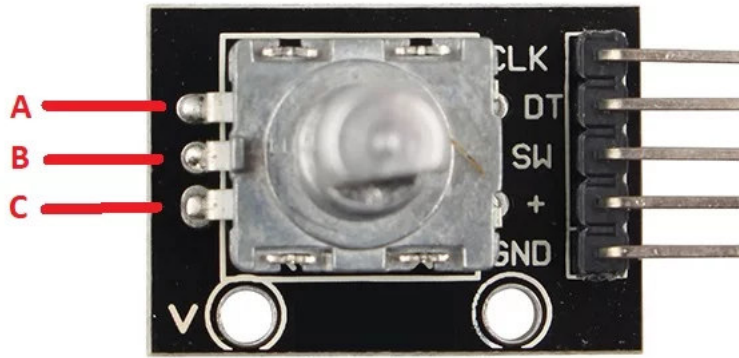
Für dieses Projekt benötigst du

- 1* Mikrocontroller mit passendem USB-Kabel
- 1* Steckbrett
- 40* Steckkabel männlich/weiblich
- 1* Rotary Encoder

Den Rotary Encoder KY-040 an einem Arduino Mikrocontroller verwenden

Wie funktioniert der Encoder?

Der Rotary Encoder KY-040 ist im Grunde ein rotierendes Eingabemodul, das einem Potentiometer stark ähnelt. Die Besonderheit des Rotary Encoders besteht darin, dass relativ simpel ausgelesen werden kann, in welche Richtung der Nutzer den Knopf (*Rotary Head*) des Encoders bewegt hat und wie viele Schritte dabei zurückgelegt wurden. Die einzelnen Schritte können dabei während des Drehens durch den Benutzer auch erfüllt werden.



Die zu sehenden Pins werden **A**, **B** und **C** genannt. Dabei sind jeweils **Pin A** und **Pin C**, sowie **Pin B** und **Pin C** miteinander verbunden. In **jeder** Encoder Position sind beide Verbindungen entweder geöffnet oder geschlossen. Rotiert man den Knopf nun in eine beliebige Richtung (im oder gegen den Uhrzeigersinn), verändert sich der Status wie folgt:

Wenn beide Verbindungen **geschlossen** sind, bewirkt eine Rotation des Knopfes die **Öffnung** der Verbindungen.

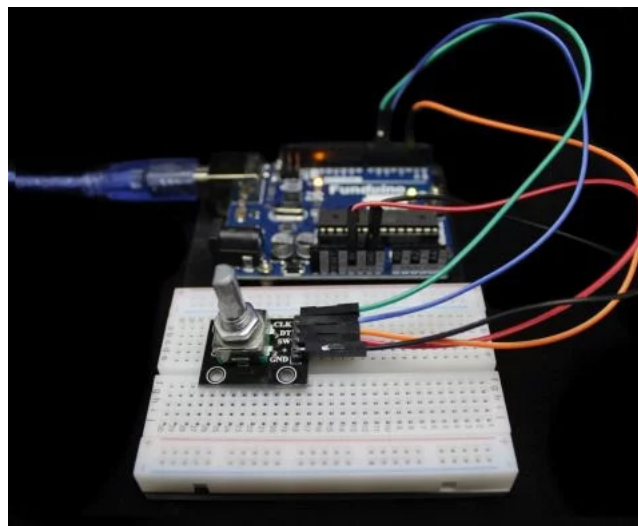
Wenn beide Verbindungen **geöffnet** sind, bewirkt eine Rotation des Knopfes die **Schließung** der Verbindungen.

Dabei gilt ebenfalls:

Rotiert man den Knopf **im** Uhrzeigersinn, verändert sich zuerst der Status der Verbindung zwischen den **Pins A und C**.

Rotiert man den Knopf **gegen** den Uhrzeigersinn, verändert sich zuerst der Status der Verbindung zwischen den **Pins B und C**.

Die Reihenfolge dieser Statusänderungen werden vom Controller ausgelesen und ausgewertet.



Pinbelegung

Der Rotary Encoder KY-040 verfügt über **fünf** Pins:

CLK = „Serial Clock“ Pin, Daten-Output 1, anzuschließen am digitalen **Pin 6** des Mikrocontrollers.

DT = Einzigartiger Pin des Moduls, Daten-Output 2, anzuschließen am digitalen **Pin 5** des Mikrocontrollers.

Die Pins CLK und DT können an einen digitalen Pins eurer Wahl angeschlossen werden. In unserem folgenden Sketch verwenden wir die oben genannten Pins D5 und D6.

SW = **Switch-Pin** – Dieser gibt ein „**LOW**“ Signal weiter, wenn der Knopf des Encoders betätigt wird. Anzuschließen an einen Interrupt-Pin des Mikrocontrollers.

Eine Übersicht aller Interrupt-Pins der Mikrocontroller findest du auf der offiziellen Seite von Arduino.

+ = Pin für die Spannungsversorgung (sonst VCC), anzuschließen an den **5V Pin** des Mikrocontrollers.

GND = *Ground Pin*, anzuschließen an den **GND Pin** des Mikrocontrollers.



Nun widmen wir uns dem Auslesen der jeweiligen Schrittposition des Encoders. Zudem möchten wir auslesen können, ob der Knopf des Encoders betätigt wurde. Die erhobenen Daten sollen im seriellen Monitor angezeigt werden. Dies gelingt uns unter anderem durch die Verwendung des **attachInterrupt()** Befehls.

Eine ausführliche Dokumentation dieser Funktion finden Sie hier.

Achtung: Das Rotary Encoder Modul KY-040 benötigt eine eigene Bibliothek. Diese Bibliothek mit dem Namen „**Encoder Library**“ findest du im nachfolgenden Link.

<https://github.com/PaulStoffregen/Encoder>

Beispielsketch

```
#include <Encoder.h>    // Verwendung der Bibliothek

const int CLK = 6;      // Definition der Pins. CLK an D6, DT an D5.
const int DT = 5;
const int SW = 2;       // Der Switch wird mit Pin D2 Verbunden. ACHTUNG : Ver
long altePosition = -999; // Definition der "alten" Position (Diese fiktive a

Encoder meinEncoder(DT,CLK); // An dieser Stelle wird ein neues Encoder Proje

void setup()    // Beginn des Setups
{
    Serial.begin(9600);

    pinMode(SW, INPUT); // Hier wird der Interrupt installiert.

    attachInterrupt(digitalPinToInterrupt(SW), Interrupt, CHANGE); // Sobald sic
}

void loop()
{

    long neuePosition = meinEncoder.read(); // Die "neue" Position des Encode
```

```
28
29     if (neuePosition != altePosition) // Sollte die neue Position ungleic
30     {
31         altePosition = neuePosition;
32         Serial.println(neuePosition);    // ...soll die aktuelle Position im
33     }
34
35 }
36
37
38 void Interrupt() // Beginn des Interrupts. Wenn der Rotary Knopf betätigt wird
39 {
40     Serial.println("Switch betaetigt"); //... das Signal ausgegeben wurde, wird
41 }
42 }
```

SALE

Funduino – Dein Onlineshop für Mikroelektronik

- ✓ Dauerhaft bis zu 10% Rabatt für Schüler, Studenten und Lehrkräfte
- ✓ Mehr als 3.500 Artikel sofort verfügbar!
- ✓ Über 8 Jahre Erfahrung mit Arduino, 3D-Druck und co.

[Zu unserem Onlineshop](#)

Funduino Starter Kit

Die Funduino Starter Kits sind speziell für Arduino Einsteiger entwickelt. Du kannst mit den vielen verschiedenen Komponenten unterschiedlichste Schaltungen aufbauen und spielerisch in die Welt der Mikroelektronik eintauchen.

