



Digitalelektronik 3

Digitale Eingabe

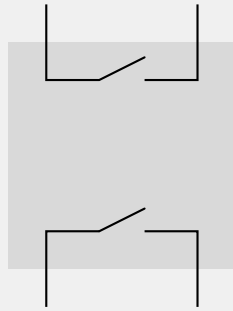
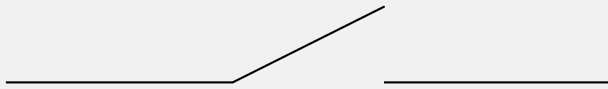
Stefan Rothe

2015-04-21



Schalter / Taster

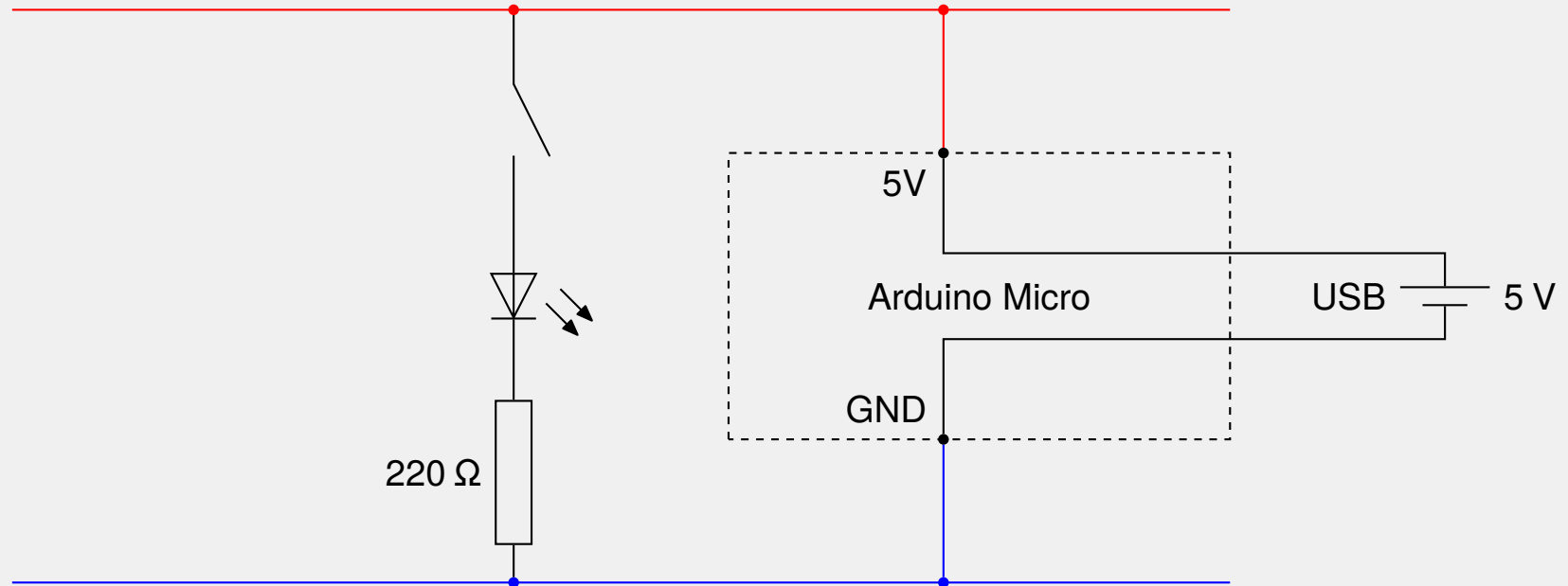
- **Schalter** unterbrechen den Stromkreis bei einer Betätigung und stellen den Kontakt bei der nächsten Betätigung her.
- **Taster** schliessen den Stromkreis für die Dauer der Betätigung.





Aufgabe 3: Taster

Bauen Sie eine Schaltung mit einer Leuchtdiode, die leuchtet, sobald ein Taster gedrückt wird.

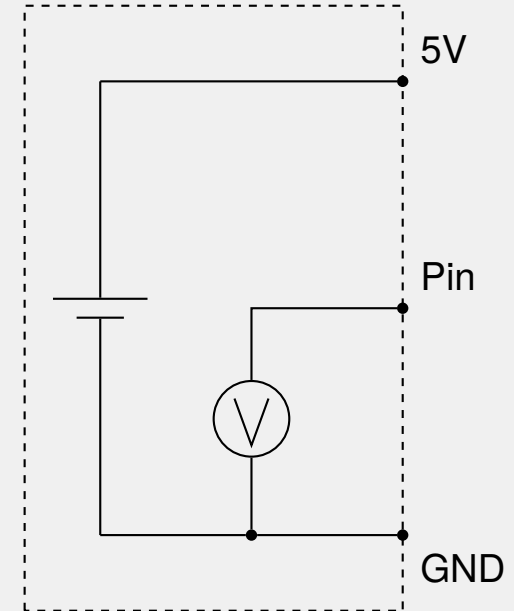




Digitale Eingabe

Bei der digitalen Eingabe wird die Spannung zwischen dem Eingabepin und der Masse (GND) gemessen.

- 0 V bedeutet **LOW**
- 5 V bedeutet **HIGH**
- undefiniert, falls Pin nicht mit 5V oder GND verbunden ist (**hochohmig**)

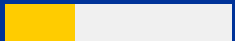
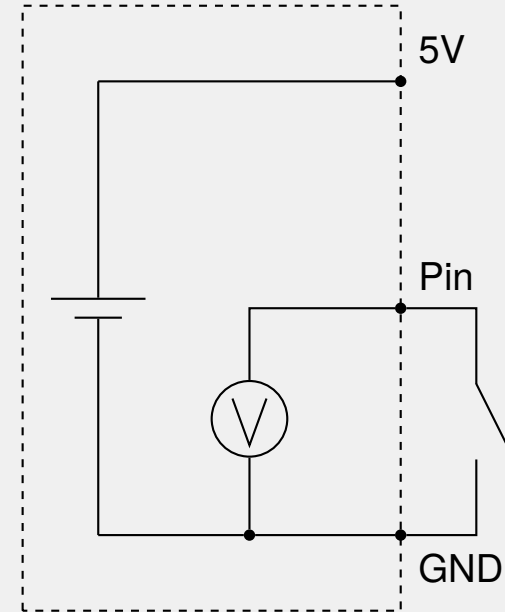




Digitale Eingabe mit Taster

Bei der digitalen Eingabe wird die Spannung zwischen dem Eingabepin und der Masse (GND) gemessen.

- Spannung 0 V, wenn Taster gedrückt.
- undefiniert, wenn Taster nicht gedrückt (**hochohmig**).

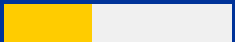
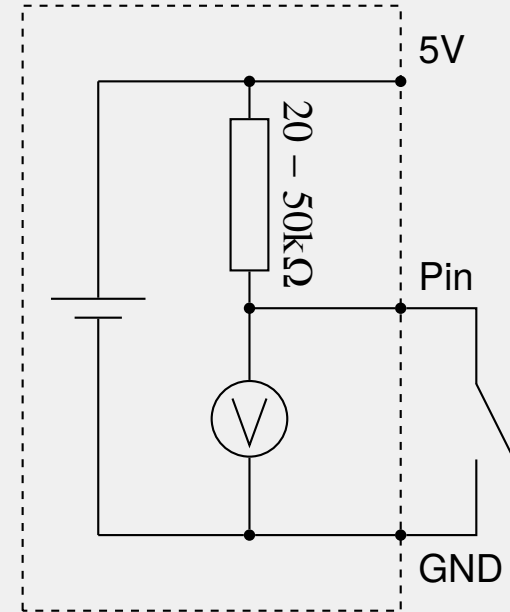




Pull-Up-Widerstand

Bei der digitalen Eingabe wird die Spannung zwischen dem Eingabepin und der Masse (GND) gemessen.

- Spannung 0 V, wenn Taster gedrückt.
- Spannung 5 V, wenn Taster nicht gedrückt.





Arduino: Digitale Eingabe 1

```
pinMode(Pin, INPUT_PULLUP)
```

- `pinMode()` legt den Modus des Pins fest
- `INPUT_PULLUP` ist eine vordefinierte Konstante
- Wird in `setup()` aufgerufen

```
const int PIN = 13;  
  
void setup() {  
    pinMode(PIN, INPUT_PULLUP);  
}  
  
void loop() {  
  
}
```



Arduino: Digitale Eingabe 2

```
digitalRead(Pin)
```

- `digitalRead()` misst aktuelle Spannung an Pin
- Liefert **LOW**, falls 0 V anliegen
- Liefert **HIGH**, falls 5 V anliegen
- Liefert zufällig **LOW** oder **HIGH**, falls der Pin hochohmig ist

```
const int PIN = 13;

void setup() {
    pinMode(PIN, INPUT_PULLUP);
}

void loop() {
    if (digitalRead(PIN) == LOW) {
        // Taste gedrückt.
    }
    else {
        // Taste nicht gedrückt.
    }
}
```




Aufgabe: Ein-/Ausschalter

Schreiben Sie ein Arduino-Programm, welches bei jedem Drücken des Tasters eine Variable hochzählt. Das Bitmuster der Variable wird durch Leuchtdioden über ein Schieberegister dargestellt.

Hinweis: Um festzustellen, ob die Taste neu gedrückt worden ist, muss sich das Programm den letzten Zustand der Taste merken.





Tastatur-Emulation mit Arduino

Start der Tastatur-Emulation:

```
Keyboard.begin();
```

Drücken einer Taste:

```
Keyboard.press('a');  
Keyboard.press(KEY_LEFT_ARROW);
```

Tippen einer Taste:

```
Keyboard.write('A');
```



Spezialtasten

KEY_LEFT_CTRL	KEY_BACKSPACE	KEY_F1
KEY_LEFT_SHIFT	KEY_TAB	KEY_F2
KEY_LEFT_ALT	KEY_RETURN	KEY_F3
KEY_LEFT_GUI	KEY_ESC	KEY_F4
KEY_RIGHT_CTRL	KEY_INSERT	KEY_F5
KEY_RIGHT_SHIFT	KEY_DELETE	KEY_F6
KEY_RIGHT_ALT	KEY_PAGE_UP	KEY_F7
KEY_RIGHT_GUI	KEY_PAGE_DOWN	KEY_F8
KEY_UP_ARROW	KEY_HOME	KEY_F9
KEY_DOWN_ARROW	KEY_END	KEY_F10
KEY_LEFT_ARROW	KEY_CAPS_LOCK	KEY_F11
KEY_RIGHT_ARROW		KEY_F12





Tastatur-Emulation mit Arduino

Aufruf	Bedeutung
<code>Keyboard.begin()</code>	Tastatur-Emulation starten
<code>Keyboard.end()</code>	Tastatur-Emulation beenden
<code>Keyboard.press(key)</code>	Beginn eines Tastendrucks
<code>Keyboard.release(key)</code>	Ende eines Tastendrucks
<code>Keyboard.releaseAll()</code>	Alle aktiven Tasten loslassen
<code>Keyboard.write(key)</code>	Tippen einer Taste simulieren
<code>Keyboard.print(text)</code>	Tippen eines Textes simulieren
<code>Keyboard.println(text)</code>	Tippen eines Textes simulieren



Übersicht Tastatur-Funktionen

Aufruf	Bedeutung
<code>Keyboard.begin()</code>	Tastatur-Emulation starten
<code>Keyboard.end()</code>	Tastatur-Emulation beenden
<code>Keyboard.press(key)</code>	Beginn eines Tastendrucks
<code>Keyboard.release(key)</code>	Ende eines Tastendrucks
<code>Keyboard.releaseAll()</code>	Alle aktiven Tasten loslassen
<code>Keyboard.write(key)</code>	Tippen einer Taste simulieren
<code>Keyboard.print(text)</code>	Tippen eines Textes simulieren
<code>Keyboard.println(text)</code>	Tippen eines Textes simulieren



Aufgabe: Eigene Tastatur

1. Schreiben Sie ein Arduino-Programm, das

- a) Tastendrücke als Druck einer beliebigen Taste der Tastatur an den Computer weiterleitet,
- b) bei einem Tastendruck einen bestimmten Text eintippt,
- c) bei einem Tastendruck Word startet.

2. Kleine Arduino-Mikrocontroller können problemlos in einem USB-Speicherstick oder einer Tastatur eingebaut werden. Überlegen Sie sich, wie Sie diese Technologie einsetzen könnten, um

- a) einen Virus auf einen Computer einzuschleusen,
- b) Passwörter auszuspionieren,
- c) ihre eigenen Passwörter sicherer zu machen.