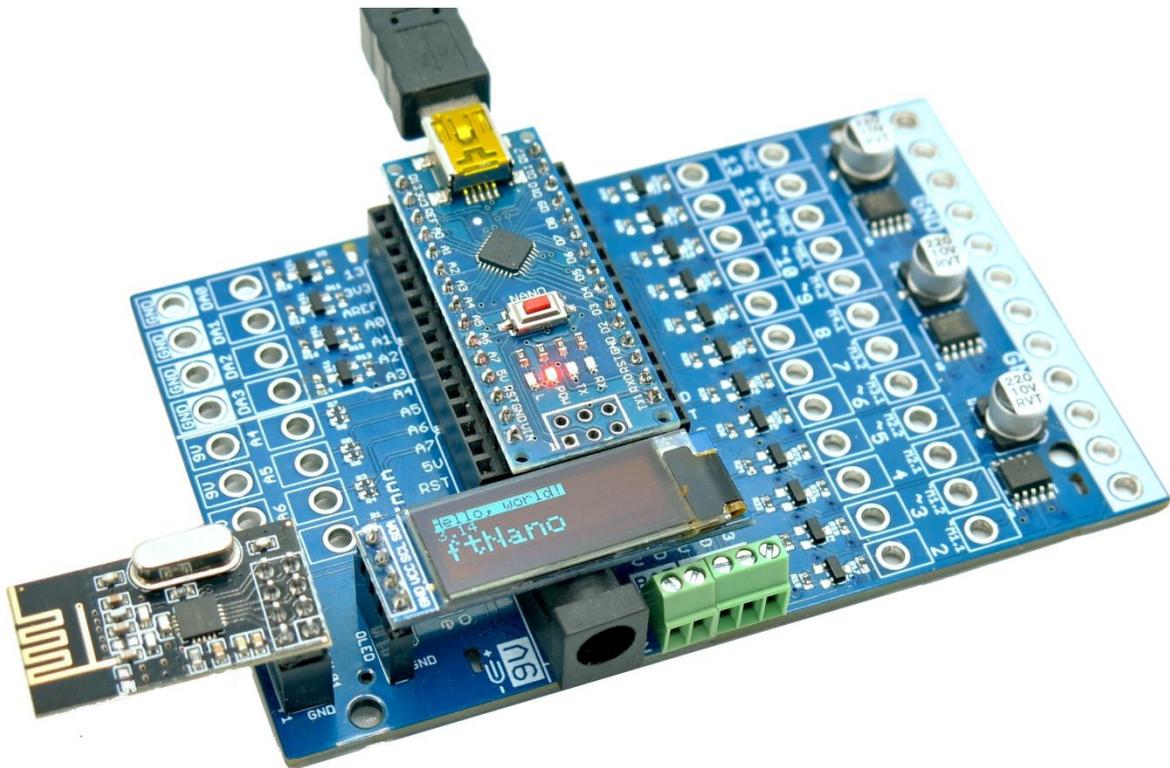


# ftNano



## Benutzerhandbuch

Florian Schäffer

<https://www.blafusel.de/>

<https://www.obd2-shop.eu/>

Die Informationen in der vorliegenden Dokumentation werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die erwähnten Soft- und Hardwarebezeichner, Marken, Gebrauchsnamen, Handelsnamen, Warenbezeichner usw. können auch dann eingetragene Warenzeichen sein und unterliegen gesetzlichen Bestimmungen, wenn darauf nicht besonders hingewiesen wird.

Bei der Zusammenstellung der Texte, Abbildungen und Beiwerke wurde mit großer Sorgfalt vorgegangen – trotzdem können Fehler nicht ausgeschlossen werden. Der Autor übernimmt keine Haftung oder juristische Verantwortung für Schäden, die durch Fahrlässigkeit, Fehler oder Auslassungen etc. in diesem Werk verursacht werden.



ISBN: [9783754348796](https://www.isbn-international.org/product/9783754348796)

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b> .....	<b>4</b>
<b>2</b>	<b>Funktionsübersicht</b> .....	<b>5</b>
2.1	Steckplatz für Arduino Nano .....	6
2.2	Buchsenleisten für Arduino-Pins .....	6
2.3	DC Eingang Hohlbuchse.....	6
2.4	DC Eingang Schraubklemme.....	6
2.5	Schraubklemme Spannungsausgang.....	7
2.6	Buchsenleiste für OLED-Display .....	7
2.7	Buchsenleiste für NRF24L01-Modul.....	7
2.8	Fischertechnik-Kontakt GND/Masse .....	7
2.9	Fischertechnik-Kontakt 9 V (VCC).....	7
2.10	Fischertechnik-Kontakt digitale I/O.....	8
2.11	Fischertechnik-Kontakt analoge I/O.....	8
2.12	Fischertechnik-Kontakt digitale I/O.....	8
2.13	Fischertechnik-Kontakt digitale Last-Ausgänge.....	8
2.14	Fischertechnik-Kontakt GND/Masse .....	8
<b>3</b>	<b>Schaltplan</b> .....	<b>9</b>
3.1	(Motor-) Treiber D2–D13 .....	9
3.2	Buchsenleiste NRF24L01 und OLED.....	9
3.3	Analoge Eingänge A4–A7.....	10
3.4	Digitale Ein-/Ausgänge D2–D13 .....	10
3.5	Digitale Ein-/Ausgänge A0–A3.....	11
<b>4</b>	<b>Inbetriebnahme</b> .....	<b>12</b>
4.1	Vorbereitung .....	12
4.2	Arduino programmieren.....	12
4.3	Digitale Ausgänge D2–D13 .....	12
4.4	Digitale Ausgänge DA0–DA3.....	13
4.5	PWM an digitalen Ausgängen .....	14
4.6	Digitale Eingänge D2–D13 .....	15
4.7	Digitale Eingänge DA0–DA3.....	16
4.8	Interrupt-Eingänge (Zähler) D2 und D3.....	17
4.9	Digitale Last-Ausgänge M1.1–M6.2.....	18
4.10	Analoge Eingänge A4–A7.....	20
<b>5</b>	<b>OLED-Display</b> .....	<b>22</b>
<b>6</b>	<b>NRF24L01 Wireless RF Transmitter</b> .....	<b>23</b>

## 1 Einführung

Konstruktionsbausteine von Fischertechnik eignen sich gut, um technische Funktionsmodelle aufzubauen. Diese können über elektronische Aktoren (bspw. Lampen und Motoren) sowie Sensoren (bspw. Taster) verfügen. Mit diesen ist bereits eine einfache Steuerung des Modells möglich. Erweiterte Möglichkeiten ergeben sich, wenn Elektronik, ein Computer oder Mikrocontroller das Modell steuert.

Alternativ zu den von Fischertechnik angebotenen Controllern bietet der ftNano die Verbindung der Modelle mit dem beliebten Arduino-Entwicklungsboard. Dadurch stehen alle weitere Funktionen zur Verfügung, die das einsteigerfreundliche Arduino-Konzept bietet, wie zum Beispiel:

- Programmierung in der Programmiersprache C/C++
- Einbinden von zahlreichen Sensoren (bspw. Helligkeit, Temperatur, Entfernung), die sonst nicht unterstützt werden
- Anschluß von sogenannten *Shields* auf denen beliebte Module etc. fertig aufgebaut sind (z. B.: SD-Speicherkarten oder LC-Displays, I<sup>2</sup>C, SPI)
- Kommunikation über sämtliche Netzwerktechniken wie WLAN, Bluetooth usw.

Der ftNano ist so konzipiert, daß er vor allem die unterschiedlichen Spannungen, die bei Fischertechnik und Mikrocontrollern wie dem ATmega (der eigentliche Mikrocontroller bei Arduino-Boards) anpaßt und zueinander kompatibel macht. Zudem wird für Motoren und Lampen ein größerer Strom benötigt, als es Mikrocontroller liefern können.

Dabei befindet sich auf dem ftNano-Board kein Mikrocontroller. Das Board enthält alle notwendigen Komponenten, um auf die vorgesehenen Steckplätze einen handelsüblichen Arduino Nano ohne Löten einfach aufzustecken. Dadurch wird es einfach, den Arduino zu wechseln, sollte dieser beispielsweise beschädigt werden. Außerdem werden die Kosten gesenkt.

Bewußt wurde darauf geachtet, den ftNano so zu konstruieren, daß keine zusätzlichen Bibliotheken für die Programmierung erforderlich sind und sich am Konzept der Programmierung mit der Arduino-IDE und der Benutzung von I/O-Anschlüssen nichts ändert. Die Programmierung und Nutzung sollen so möglichst der üblichen Arduino-Technik entsprechen. Außerdem sind alle Erweiterungen (Shields) etc. die es für den Arduino gibt uneingeschränkt nutzbar, da sämtliche Pins des Arduino Nano über zwei Buchsenleisten zugänglich sind.

Wichtige Merkmale:

- bis zu 16 digitale I/O kompatibel zu 9 V/20 mA
- bis zu 6 digitale Leistungsausgänge 9 V mit je bis zu 200 mA
- 4 analoge Eingänge 0 V – 10 V
- Buchsenleiste für OLED-Display
- Buchsenleiste für NRF24L01 Wireless RF-Transmitter
- Buchsenleisten für sämtliche I/O-Pins des Arduino Nano

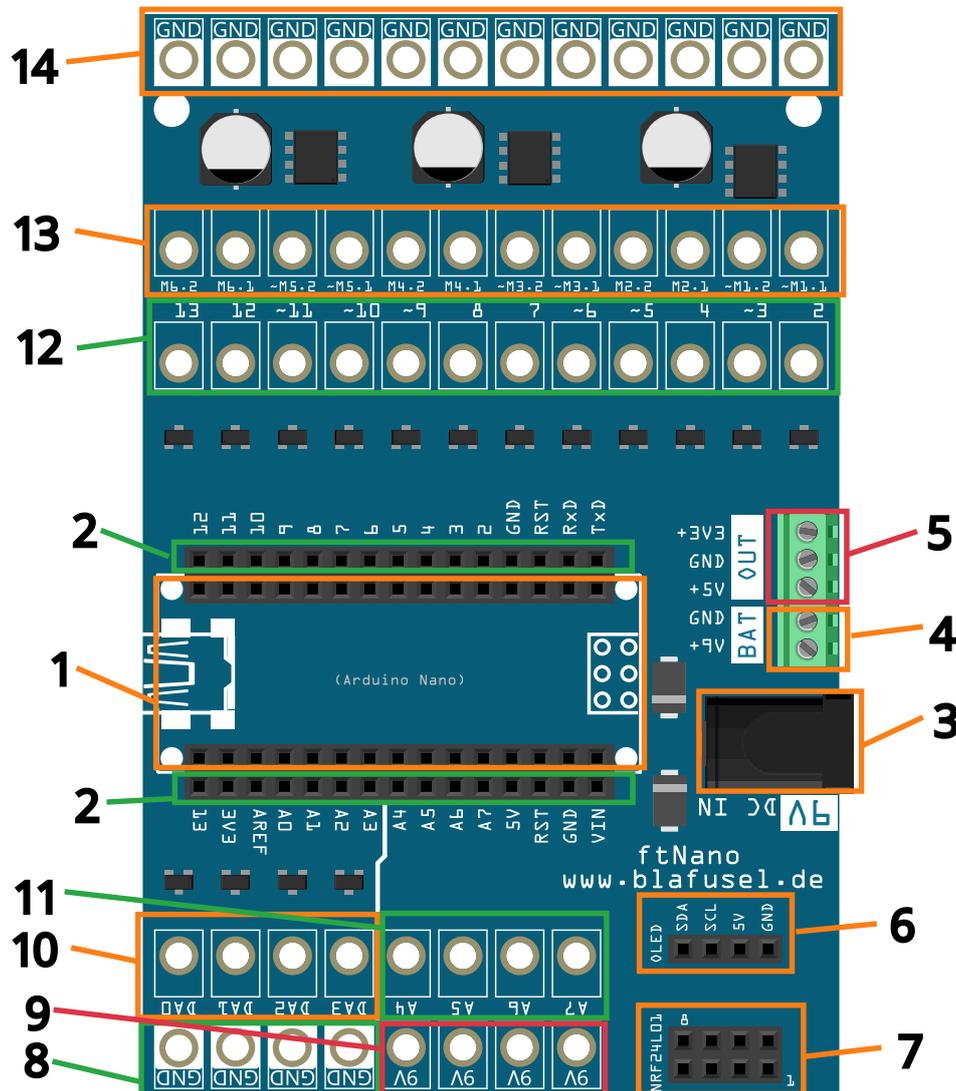
---

Diese Anleitung setzt voraus, daß der Benutzer mit der Arduino-Programmierung in C/C++ mittels Arduino-IDE sowie dem Arduino an sich vertraut ist. Ebenfalls werden Elektronikkenntnisse vorausgesetzt. Für die Darstellung von Schaltungen wird Fritzing benutzt, womit der Benutzer ebenfalls vertraut sein sollte, wenn er damit arbeiten möchte.

---

## 2 Funktionsübersicht

Der ftNano bietet folgende Anschlüsse:



1. Steckplatz für Arduino Nano
2. Buchsenleiste für Arduino-Pins
3. DC Eingang Hohlbuchse
4. DC Eingang Schraubklemme
5. Schraubklemme Spannungsausgang
6. Buchsenleiste für OLED-Display
7. Buchsenleiste für NRF24L01-Modul
8. Fischertechnik-Kontakt GND/Masse
9. Fischertechnik-Kontakt 9 V (VCC)
10. Fischertechnik-Kontakt digitale I/O
11. Fischertechnik-Kontakt analoge I/O
12. Fischertechnik-Kontakt digitale I/O
13. Fischertechnik-Kontakt digitale Last-Ausgänge
14. Fischertechnik-Kontakt GND/Masse

## 2.1 Steckplatz für Arduino Nano

In diese Buchsenleisten wird der Arduino Nano eingesetzt. Die USB-Buchse weist wie im Bestückungsdruck zu sehen, nach links. <https://store.arduino.cc/products/arduino-nano>

Über die USB-Buchse des Arduino wird der Mikrocontroller programmiert und der Arduino mit Spannung versorgt. Diese Spannung liegt auch am ftNano an. Der Arduino kann nicht genügend Strom für die Versorgung von Fischertechnik-Bauteilen etc. liefern. Aus diesem Grund ist eine (zusätzliche) Spannungsversorgung über einen der DC-Eingänge notwendig.

## 2.2 Buchsenleisten für Arduino-Pins

Über die zwei Buchsenleistenreihen besteht die Möglichkeit, mit Jumper-Kabeln direkten Kontakt zu den I/O-Pins des Arduinos herzustellen. Bei direkter Nutzung der Arduino Nano-Anschlüsse gelten die Parameter zu Strom und Spannung für den Arduino Nano: <https://docs.arduino.cc/hardware/nano>

## 2.3 DC Eingang Hohlbuchse

Über die Hohlbuchse kann der ftNano (sowie der Arduino) mit Spannung versorgt werden.

Die Eingangsspannung darf **7–12 Volt** betragen (bzw. gemäß Spezifikation Arduino Nano). Für Fischertechnik werden 9 V Gleichspannung benötigt. **Die Spannung darf nicht größer sein**, da ansonsten der Arduino und der ftNano beschädigt werden. Es ist vorab sicherzustellen, daß ein geeignetes Netzteil bzw. Spannungsquelle benutzt wird. Die Spannung sollte möglichst frei von Störsignalen und Spannungsschwankungen sein. Die Leistung hängt von der benutzten externen Komponenten ab, die vom ftNano gesteuert werden. Eine Ausgangsleistung von 2 A dürfte für die meisten Anwendungen ausreichend sein. Am besten eignen sich handelsübliche Schaltnetzteile (Kurzschlußfest und mit Überlastschutz) in Form von Steckern für die Steckdose für um die 10 €. Alternativ kann auch eine 9 V-Blockbatterie o. ä. genutzt werden.



*Festspannungsnetzteil*

Die DC-Buchse ist passend für Stecker mit den Maßen 5,50 mm × 2,10 mm. Der innere Stift ist GND/Masse und außen VCC (Pluspol). Der Eingang ist gegen Verpolung geschützt.

## 2.4 DC Eingang Schraubklemme

Schraubklemme zum Anschluß der Versorgungsspannung. Es gelten die Angaben aus dem vorherigen Abschnitt.

## 2.5 Schraubklemme Spannungsausgang

Ausgang für die Spannungen, die auf dem Arduino erzeugt werden. Die maximale Belastbarkeit des Arduino Nano ist zu beachten.

## 2.6 Buchsenleiste für OLED-Display

Buchsenleiste an dem die Signale SDA (A4), SCL (A5), 5 V und GND anliegen. Hier kann bspw. ein OLED-Display der Größe 0.91 inch, I2C, 128 x 32 Pixels aufgesteckt werden (Achtung: unterschiedliche Anordnung der Pins möglich).

---

Bei Verwendung des Buchsenleiste ist zu beachten, daß die angegebenen I/O-Pins belegt werden und nicht durch für andere Zwecke genutzt werden: A4 und A5

---

## 2.7 Buchsenleiste für NRF24L01-Modul

Buchsenleiste an dem die folgenden Signale anliegen:

MISO (12)	SCK (13)	CE (2)	GND
NC	MOSI (11)	CSN (7)	3,3 V

Auf die Buchsenleiste kann ein NRF24L01-Modul gesteckt werden, um Daten drahtlos zu übertragen.

---

Bei Verwendung des Buchsenleiste ist zu beachten, daß die angegebenen I/O-Pins belegt werden und nicht durch für andere Zwecke genutzt werden: 2/M2.1, 7/M4.1, 11/M5.2, 12/M6.1, 13/M6.2  
Im Programmcode ist die Zuordnung zu CE und CSN ggf. anzupassen.

---

## 2.8 Fischertechnik-Kontakt GND/Masse

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

Verbindung mit GND.

---

Die Löcher für die Fischertechnik-Kontakte sind so dimensioniert, daß ein sicherer Kontakt und Halt für die Flachstecker von Fischertechnik entsteht. Sitzt der Stecker zu locker, können die Kontaktelemente am Flachstecker mit einem kleinen Schraubenzieher minimal gespreizt werden.

---

## 2.9 Fischertechnik-Kontakt 9 V (VCC)

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

Verbindung mit der Versorgungsspannung VCC

## 2.10 Fischertechnik-Kontakt digitale I/O

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

Die Kontakte sind den entsprechenden I/O-Pins des Arduinos gemäß der Beschriftung zugeordnet. Analoge Arduino-Eingänge können immer wie digitale Ein-/Ausgänge benutzt werden. Hier werden die vier analog/digitalen I/Os des Arduinos als digitale I/Os benutzt.

Die digitalen Ein-/Ausgänge (I/O) verfügen über einen Treiber, um die Spannung des Arduino Nano (5 V) an Fischertechnik 9 V anzupassen. Die maximale Belastbarkeit eines Ausgangs liegt bei etwa 20 mA.

## 2.11 Fischertechnik-Kontakt analoge I/O

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

Die Kontakte sind den entsprechenden I/O-Pins des Arduinos gemäß der Beschriftung zugeordnet. Hier handelt es sich um vier analoge Eingänge.

Die Eingänge sind über einen Spannungsteiler mit dem Arduino verbunden ( $2 \times 47 \text{ k}\Omega$ , siehe Schaltplan). Die maximale Eingangsspannung an den analogen Eingänge des Arduinos beträgt 5 V. Über den Spannungsteiler kann eine Spannung von maximal 10 V über die analogen Fischertechnik-Kontakte eingespeist werden.

## 2.12 Fischertechnik-Kontakt digitale I/O

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

Die Kontakte sind den entsprechenden digitalen I/O-Pins des Arduinos gemäß der Beschriftung zugeordnet. Es gibt keinerlei Einschränkungen hinsichtlich der normalen Funktionalität. Die mit einer Tilde („~“) gekennzeichneten Anschlüsse bieten die Möglichkeit, ein PWM-Signal auszugeben (Standardfunktion des Arduino)

Die digitalen Ein-/Ausgänge (I/O) verfügen über einen Treiber, um die Spannung des Arduino Nano (5 V) an Fischertechnik 9 V anzupassen. Die maximale Belastbarkeit eines Ausgangs liegt bei etwa 20 mA.

## 2.13 Fischertechnik-Kontakt digitale Last-Ausgänge

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

Diese Kontakte sind über einen (Motor-) Treiber mit dem Arduino verbunden und dienen als Ausgänge zum Anschluß von höheren Lasten wie Lampen oder Motoren. Jeder Ausgang kann ca. 200 mA liefern. Die Treiber-ICs verfügen über einen Überlastschutz.

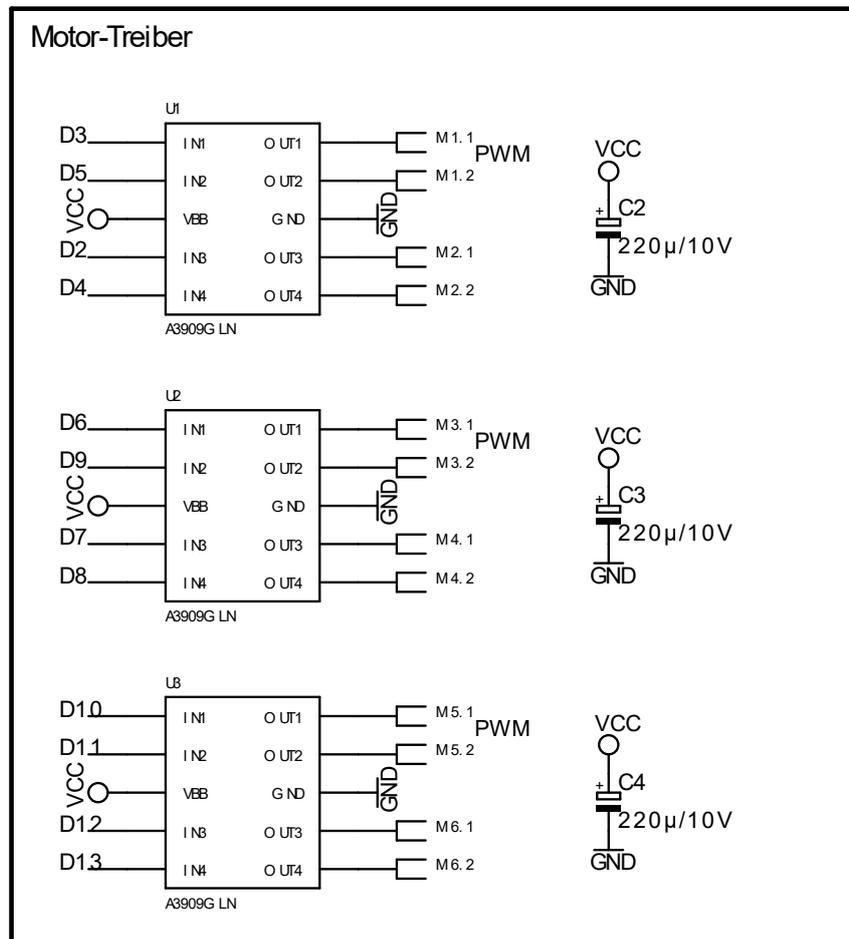
## 2.14 Fischertechnik-Kontakt GND/Masse

Kontakt-Bohrungen, um Fischertechnik-Flachstecker aufzunehmen.

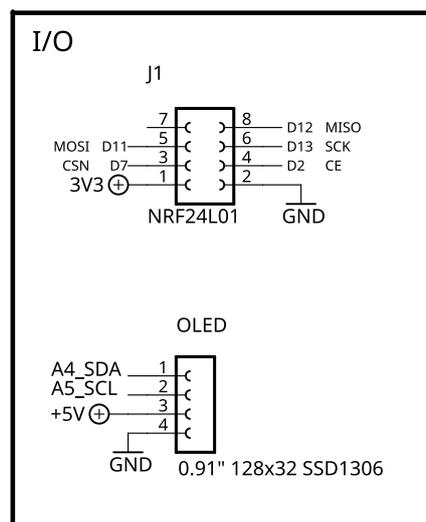
Verbindung mit GND.

### 3 Schaltplan

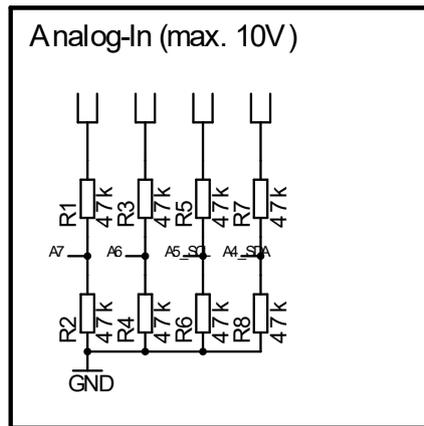
#### 3.1 (Motor-) Treiber D2–D13



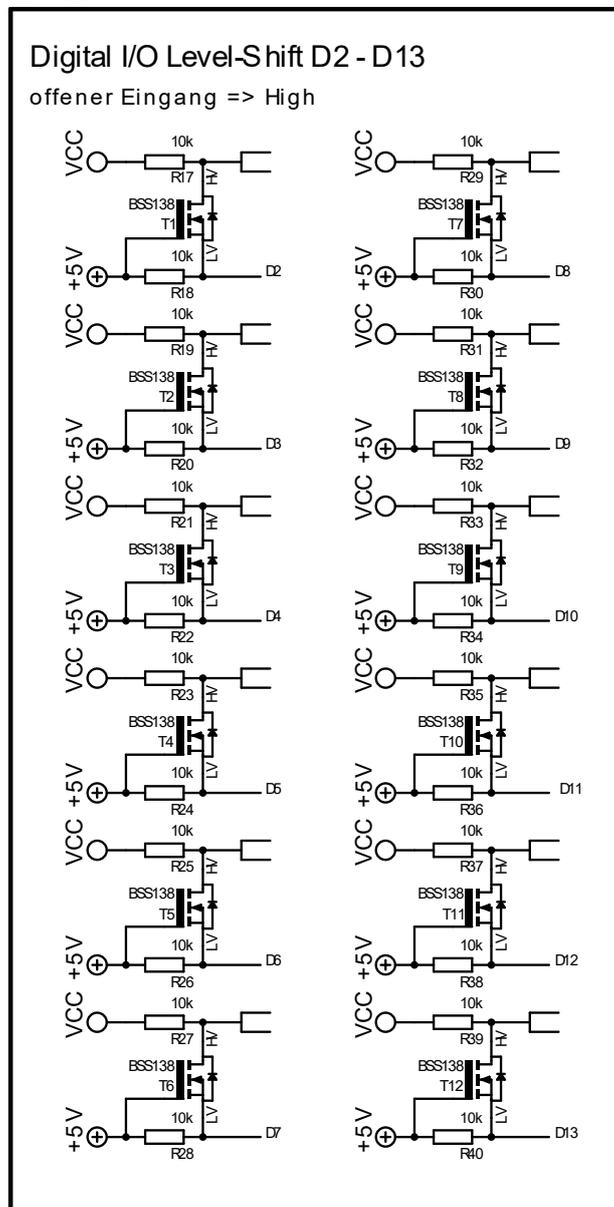
#### 3.2 Buchsenleiste NRF24L01 und OLED



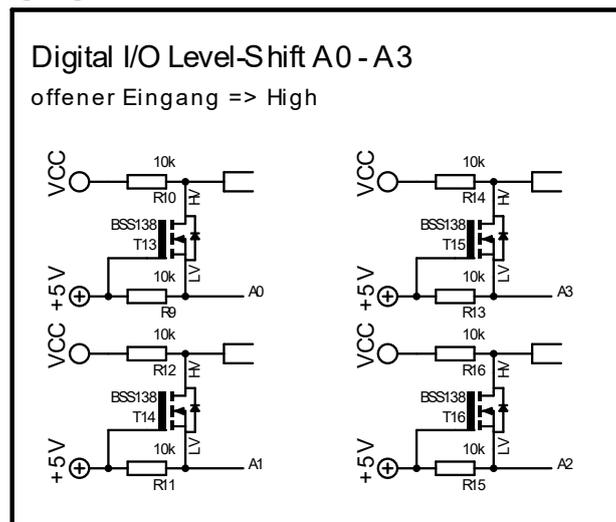
### 3.3 Analoge Eingänge A4–A7



### 3.4 Digitale Ein-/Ausgänge D2–D13



### 3.5 Digitale Ein-/Ausgänge A0–A3



## 4 Inbetriebnahme

### 4.1 Vorbereitung

Nachdem auf die Buchsenleiste für den Arduino Nano (s. Kapitel 2.1) ein solcher gesteckt wurde, ist der ftNano betriebsbereit und kann mit Spannung über einen der DC-Eingänge mit Gleichspannung versorgt werden (s. Kapitel 2.3 und 2.4). Die Power-LED auf dem Arduino Nano leuchtet daraufhin.

---

Die hier verwendeten Bauteile für das Programm Fritzing (<https://fritzing.org/>) können auf der Seite <https://github.com/coderfls/Fritzing-Parts> heruntergeladen werden.

---

### 4.2 Arduino programmieren

Der Arduino wird wie üblich in C/C++ programmiert. Hierfür wird in dieser Dokumentation die Arduino-IDE (<https://www.arduino.cc/en/software>, <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2>) unter Windows benutzt, es können aber auch andere Entwicklungsumgebungen eingesetzt werden.

Zur Programmübertragung vom PC auf den Arduino muß der Arduino Nano mit einem USB-Kabel am PC angeschlossen sein.

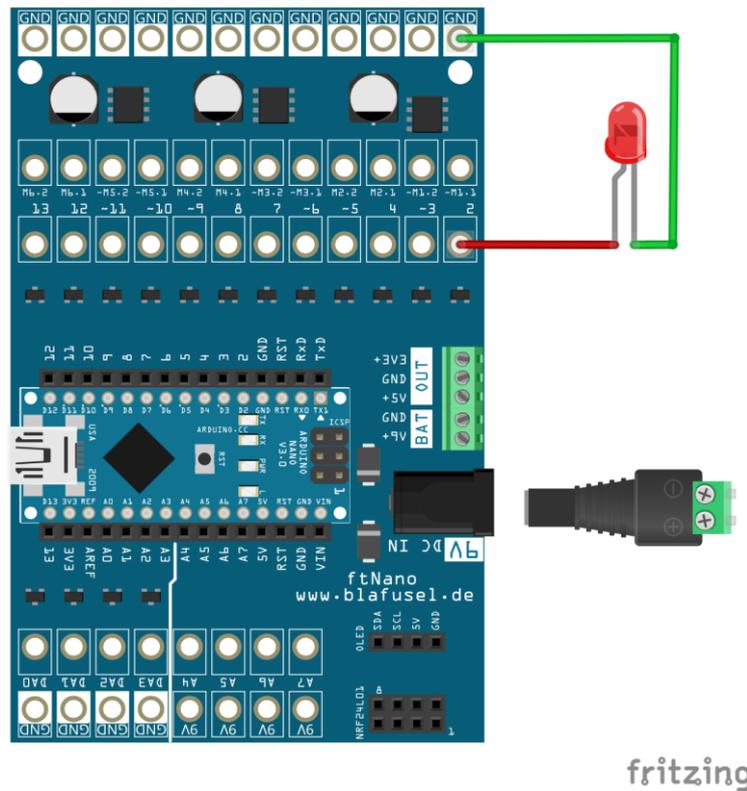
In der IDE muß wie üblich der passende COM-Port gewählt werden (*Werkzeuge/Port*), an dem der Arduino Nano zu finden ist. Außerdem muß bei *Werkzeuge/Board* der Eintrag *Arduino Nano* stehen bzw. über das Untermenü *Arduino AVR Boards* gewählt werden. Je nach Modell des Arduino muß bei *Werkzeuge/Prozessor* gewählt werden, welcher ATmega (ATmega128 oder ATmega328) auf dem Nano verbaut ist und ob ggf. der alte Bootloader (*old*) installiert ist.

### 4.3 Digitale Ausgänge D2–D13

 `digital_out`

Die digitalen I/O-Anschlüsse können wie üblich einzeln als Ausgang genutzt werden.

Weil der Pegelwandler die Versorgungsspannung (9 V) über einen Widerstand mit 10 kΩ bereitstellt, kann eine LED i. d. R. direkt und ohne Vorwiderstand angeschlossen werden.



```

void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  delay(200);
}

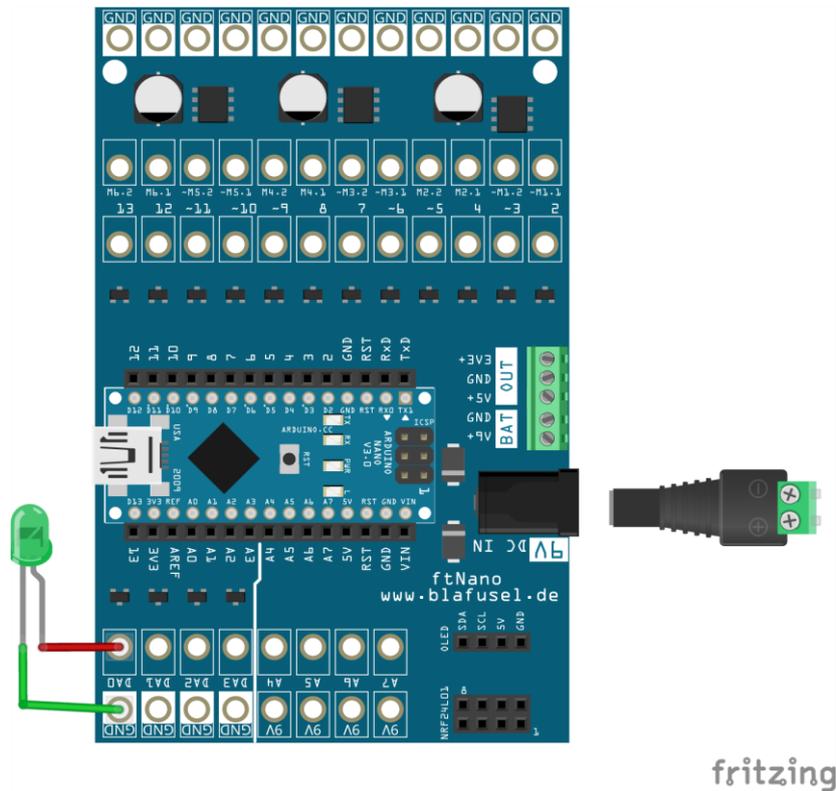
```

#### 4.4 Digitale Ausgänge DA0–DA3

##### da\_out

Die I/O-Pins DA0–DA4 können wie üblich einzeln als digitale Ausgang genutzt werden.

Weil der Pegelwandler die Versorgungsspannung (9 V) über einen Widerstand mit 10 kΩ bereitstellt, kann eine LED i. d. R. direkt und ohne Vorwiderstand angeschlossen werden.



```

void setup() {
  pinMode(A0, OUTPUT);
}

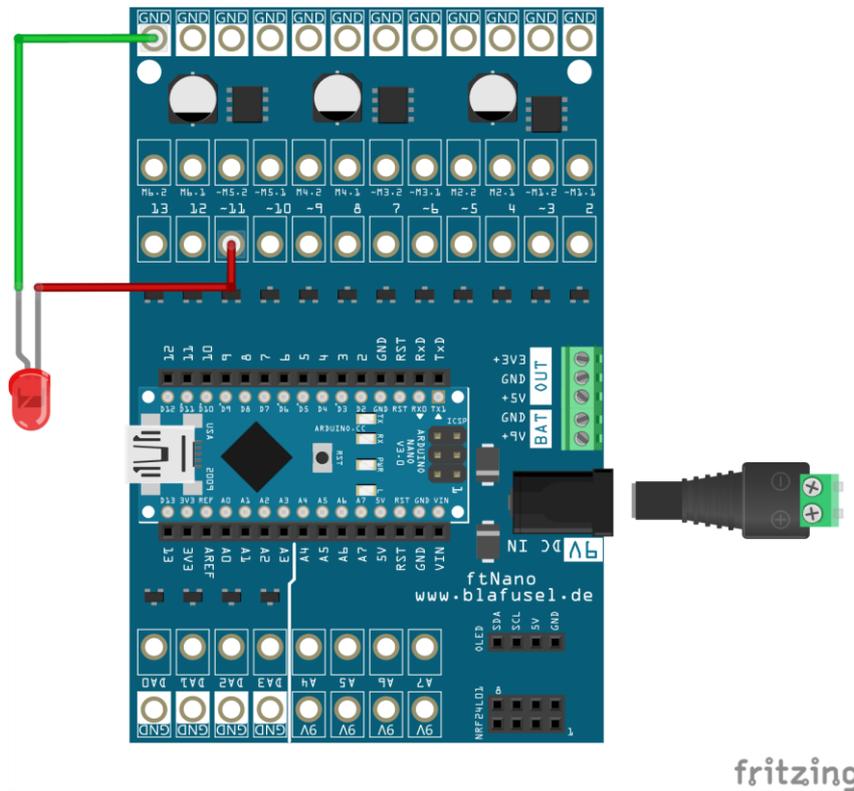
void loop() {
  digitalWrite(A0, HIGH);
  delay(2000);
  digitalWrite(A0, LOW);
  delay(200);
}

```

## 4.5 PWM an digitalen Ausgängen

### pwm

Der Arduino bietet einige Ausgänge, die ein Pulsweitenmoduliertes Signal (PWM) ausgeben können, ohne daß hierfür Ressourcen im Programm notwendig sind. Diese sind mit einer Tilde („~“) gekennzeichnet. Seitens Arduino wird dies als analoges Ausgangssignal bezeichnet, was faktisch falsch ist <https://docs.arduino.cc/learn/microcontrollers/analog-output>.



```
uint8_t i;

void setup() {} // useless stupid stuff

void loop() {
  for (i = 0; i <= 255; i++) {
    analogWrite(11, i); // PWM
    delay(20);
  }
}
```

## 4.6 Digitale Eingänge D2–D13

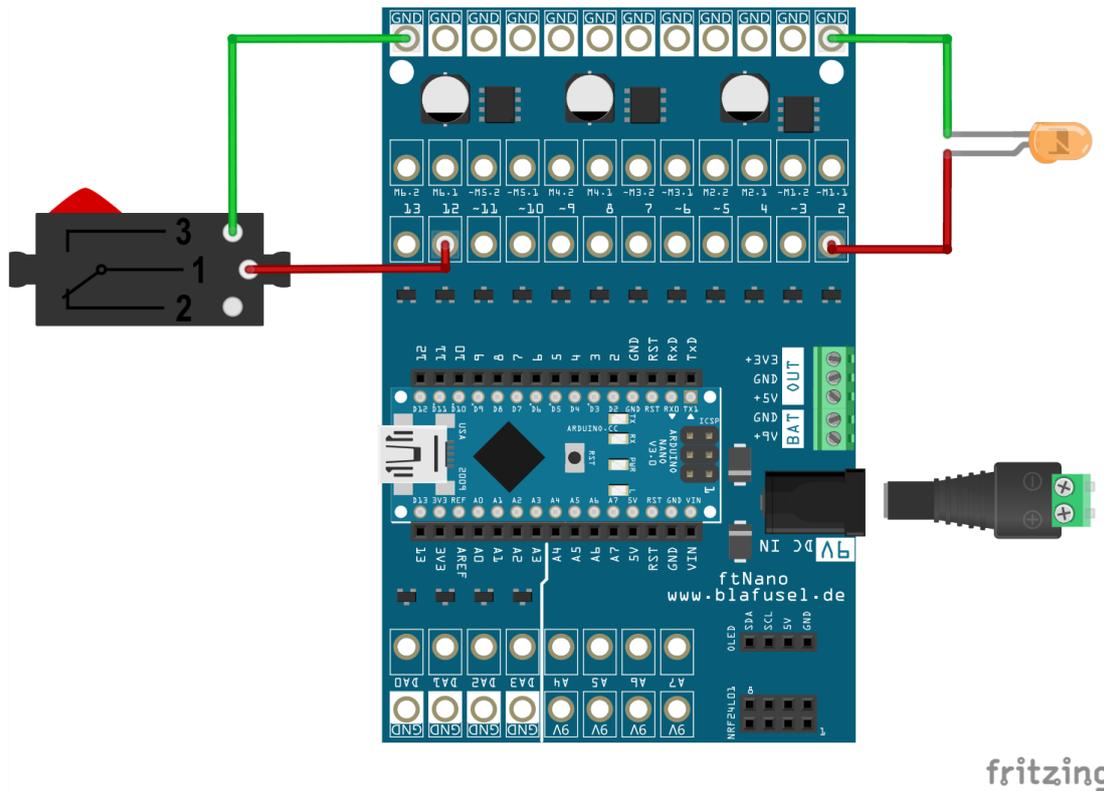
### digital\_in

Die I/O-Pins können über die Pegelwandler als Eingang für 0 V bzw. 9 V Signalpegel genutzt werden.

---

Die Eingänge sind Low-Aktiv, weil der Pegelwandler einen Pull-Up-Widerstand besitzt. Das bedeutet, daß das Eingangssignal im unbeschalteten Zustand High (1) führt und gegen GND (Low/0) geschaltet werden muß.

---



fritzing

```

void setup() {
  pinMode (2, OUTPUT);
  pinMode (12, INPUT);
}

void loop() {
  if (!digitalRead(12))           // Input is low active
    digitalWrite (2, HIGH);      // key pressed (closed)
  else
    digitalWrite (2, LOW);
}

```

---

Auf dem Arduino ist die interne LED „L“ an Pin 13 angeschlossen. Je nach Board kann es sein, daß dies die Nutzung des Pins als Eingang behindert, weil intern immer Low-Pegel am Eingang erkannt wird. Es ist empfehlenswert, Pin 13 nur als Ausgang zu nutzen.

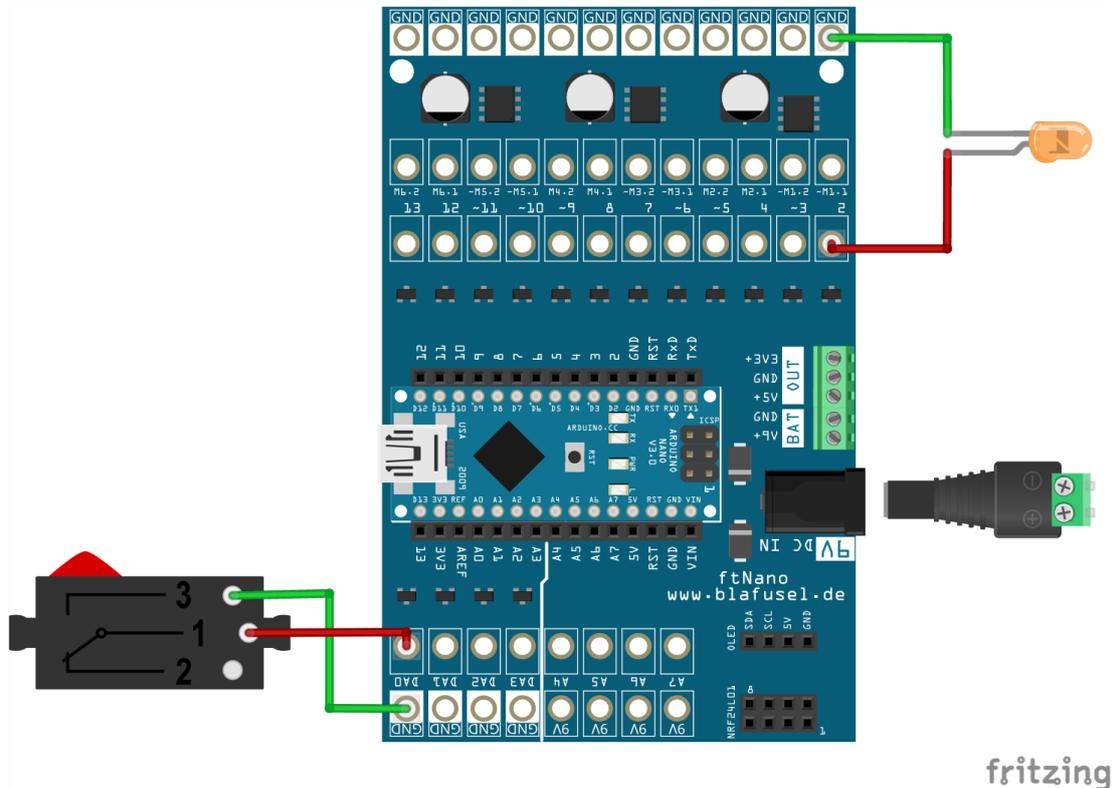
---

## 4.7 Digitale Eingänge DA0–DA3

### da\_in

Die I/O-Pins können über die Pegelwandler als Eingang für 0 V bzw. 9 V Signalpegel genutzt werden.

Die Eingänge sind Low-Aktiv, weil der Pegelwandler einen Pull-Up-Widerstand besitzt. Das bedeutet, daß das Eingangssignal im unbeschalteten Zustand High (1) führt und gegen GND (Low/0) geschaltet werden muß.



fritzing

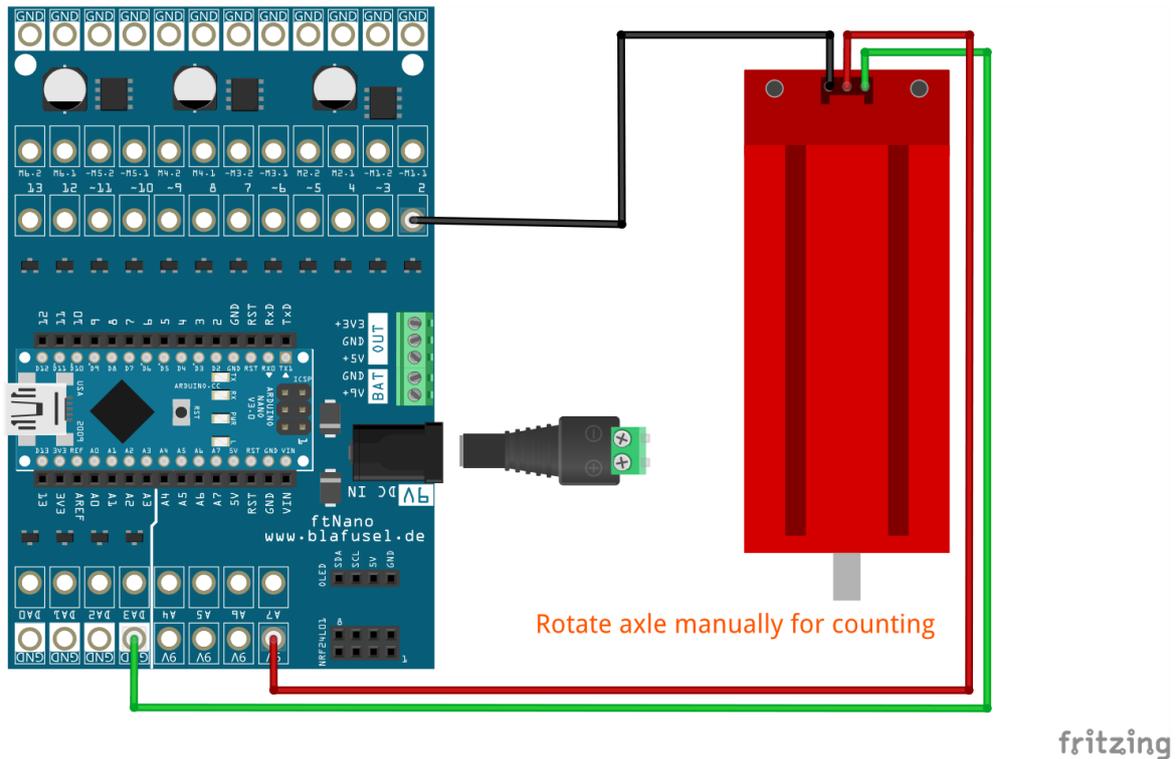
```
void setup() {
  pinMode (2, OUTPUT);
  pinMode (A0, INPUT);
}

void loop() {
  if (digitalRead(A0))          // Input is low active
    digitalWrite (2, HIGH);    // key pressed (opened in this example!)
  else
    digitalWrite (2, LOW);
}
```

## 4.8 Interrupt-Eingänge (Zähler) D2 und D3

### irq

Der Arduino Nano kann an den Eingängen 2 und 3 Interrupts erkennen, die die normale Programmausführung unterbrechen (<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>). Diese Eingänge werden genutzt, um jederzeit auf Ereignisse zu reagieren oder unabhängig vom eigentlichen Programm Impulse zu zählen (bspw. von Tastern, Drehencodern oder Motoren)



```
volatile uint16_t cnt;    // must be volatile

void setup() {
  pinMode(2, INPUT);
  attachInterrupt(digitalPinToInterrupt(2), count, CHANGE);
  Serial.begin(9600);    // for serial output open "serial monitor"
}

void loop() {
  delay(1000);
  Serial.println(cnt);  // output actual counter value
  cnt = 0;
}

// count interrupts
void count() {
  cnt++;    // increment
}
```

## 4.9 Digitale Last-Ausgänge M1.1–M6.2

### motor

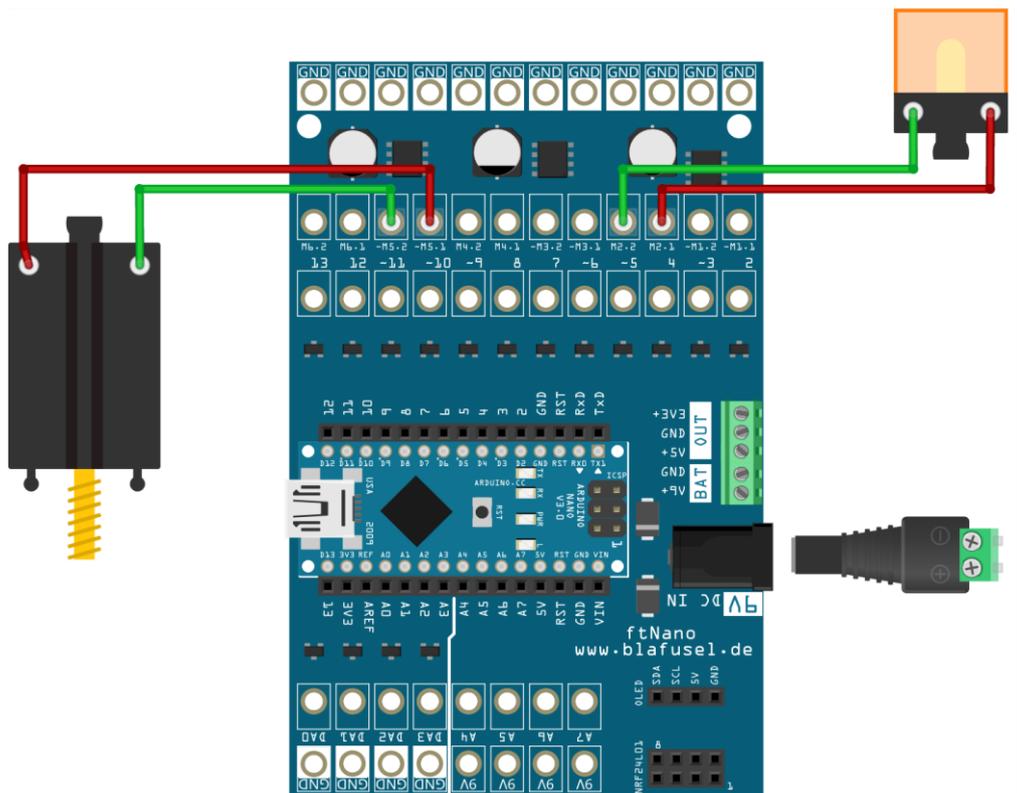
Über (Motor-) Treiber stehen leistungsfähige Ausgänge als Alternative zu den digitalen Ausgängen D2–D13 bereit. Je zwei digitale Ausgänge D2–D13 bilden eine Gruppe von zwei Lastausgängen. Die beiden Ausgänge müssen unterschiedliche Signalpegel bzw. beide Low führen. Damit kann dann auch die Drehrichtung eines Elektromotors beeinflusst werden. Einzelne Ausgänge können mit einem vom Arduino automatisch erzeugten PWM-Signal gesteuert werden (mit Tilde gekennzeichnet).

Zuordnung der Ausgänge (vgl. mit dem Schaltplan):

Digital D...	Last M...	PWM?
3	1.1	✓
5	1.2	✓
2	2.1	—
4	2.2	—
6	3.1	✓
9	3.2	✓
7	4.1	—
8	4.2	—
10	5.1	✓
11	5.2	✓
12	6.1	—
13	6.2	—

Es vereinfacht die Nutzung, wenn für die Lastausgänge im Programm mittels #define ein Makro erstellt wird, bei dem sinnvoll benannten Bezeichnern die digitalen Ports zugeordnet werden. Bspw.:

```
#define M1_1 3
#define M1_2 5
```



fritzing

```
#define M2_1 2
#define M2_2 4
#define M5_1 10
#define M5_2 11

void setup() {
  pinMode (M2_1, OUTPUT);
  pinMode (M2_2, OUTPUT);
  pinMode (M5_1, OUTPUT);
  pinMode (M5_2, OUTPUT);
}

void loop() {
  // rotate one direction
  digitalWrite (M5_1, LOW);
  analogWrite (M5_2, 40); // PWM - must maybe higher
  delay(2000);

  // stop
  digitalWrite (M2_1, LOW); // light bulb
  digitalWrite (M2_2, HIGH); // light bulb ON
  digitalWrite (M5_1, LOW);
  digitalWrite (M5_2, LOW);
  delay(500);
  digitalWrite (M2_1, LOW); // light bulb
  digitalWrite (M2_2, LOW); // light bulb OFF

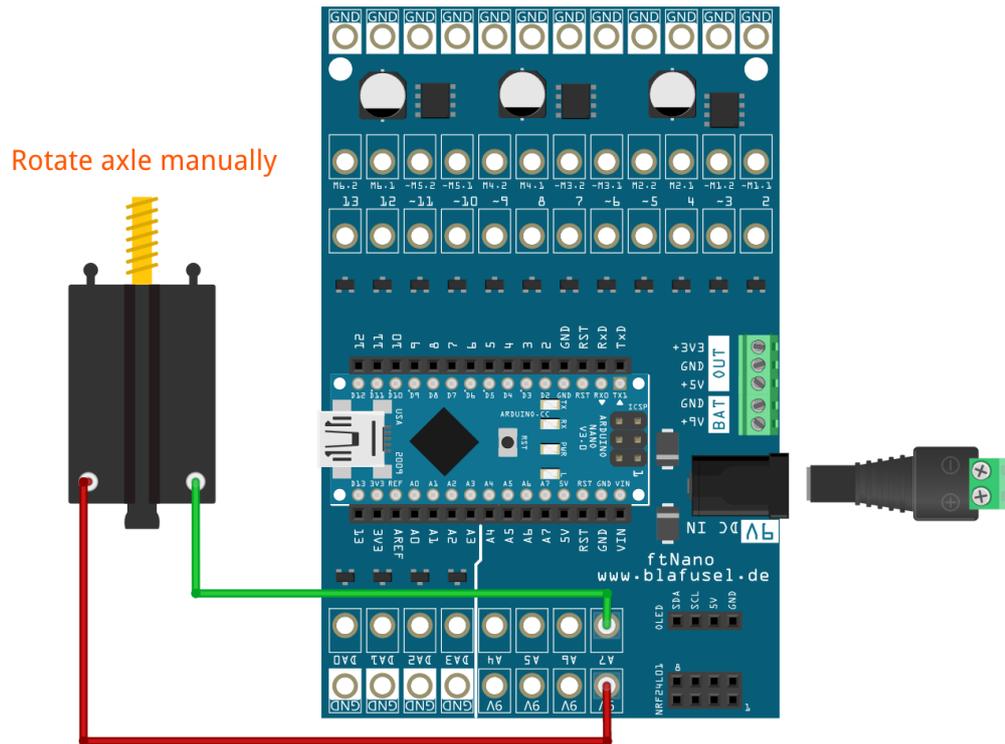
  // rotate other direction
  digitalWrite (M5_2, LOW);
  analogWrite (M5_1, 255); // PWM - must maybe higher
  delay(2000);

  // stop
  digitalWrite (M2_1, LOW); // light bulb
  digitalWrite (M2_2, HIGH); // light bulb ON
  digitalWrite (M5_1, LOW);
  digitalWrite (M5_2, LOW);
  delay(500);
  digitalWrite (M2_1, LOW); // light bulb
  digitalWrite (M2_2, LOW); // light bulb OFF
}
```

## 4.10 Analoge Eingänge A4–A7

### analog\_in

An vier analogen Eingängen befindet sich ein Spannungsteiler (s. Schaltplan). Über diesen können analoge Spannungen im Bereich 0 V – 10 V gemessen werden.



fritzing

```

void setup() {
  Serial.begin(9600);
}

void loop() {
  // rotate motor axle to work as an generator
  Serial.println(analogRead(A7)); //read analog value
  delay (50);
}

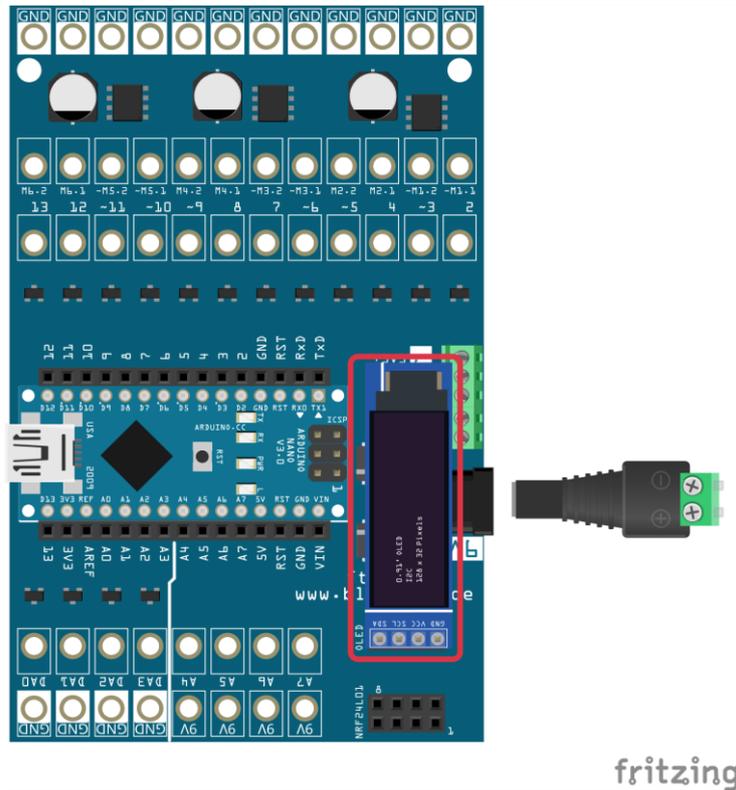
```



## 5 OLED-Display

### oled

Auf die Buchsenleiste OLED kann ein Display aufgesetzt oder angeschlossen werden. Die Signale an der Buchsenleiste können auch für andere Hardware benutzt werden. Beachten Sie die Hinweise in Kapitel 2.6.

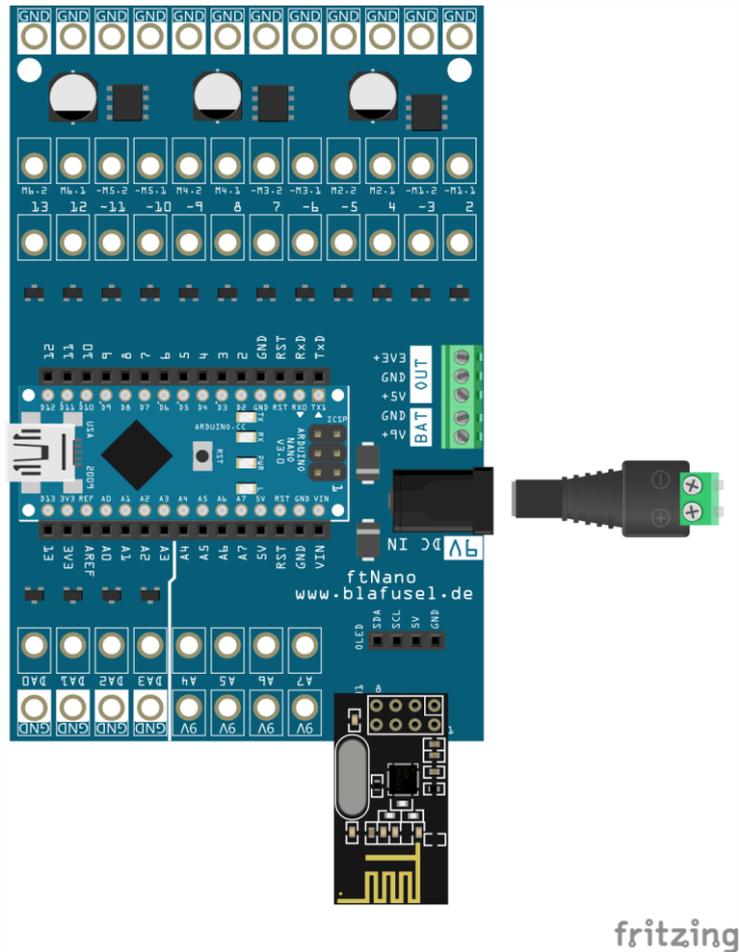


Zur Ansteuerung von Grafik-Displays gibt es eine Vielzahl an Bibliotheken. Diese müssen ggf. über die Arduino-IDE und *Werkzeuge/Bibliotheken verwalten* nachträglich installiert werden. Die Bibliotheken liefern meistens auch Beispielprogramme mit, die über *Datei/Beispiele/...* geöffnet werden können. Im Beispielordner befindet sich ein Programm, welches Text und eine Zahl auf dem Display ausgibt.

## 6 NRF24L01 Wireless RF Transmitter

### nrf24l01

Mit dem NRF24L01 lassen sich Daten zu einem anderen Transmitter übertragen oder auch per Bluetooth Low Energy (BLE) auf ein geeignetes Smartphone. Die Signale an der Buchsenleiste können auch für andere Hardware benutzt werden. Beachten Sie die Hinweise in Kapitel 2.7.



Im Beispielordner befindet sich ein einfaches Programm zum Senden von Daten.