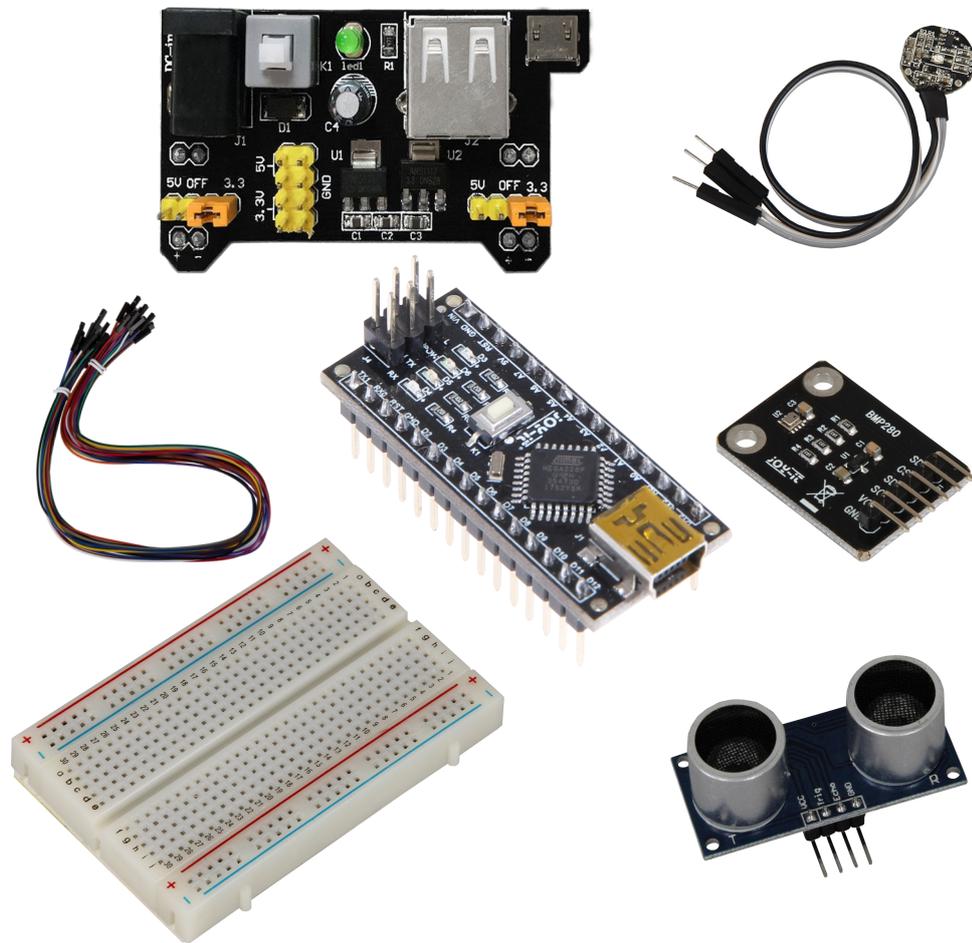


# ARDUINO NANO SENSOR SET



## 1. ALLGEMEINE INFORMATIONEN

Sehr geehrter Kunde,  
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

In dieser Anleitung finden Sie detaillierte Erklärungen und Beispiele zu allen in diesem Set enthaltenen Module.

Probieren sie es aus und erweitern sie die Beispiele nach Ihren eigenen Vorstellungen.

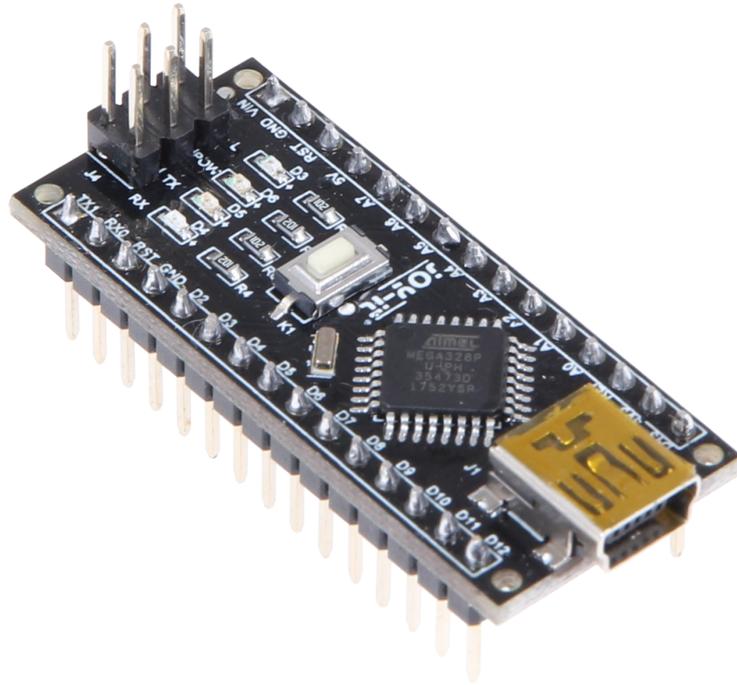
Im Lieferumfang enthalten sind:

Arduino Nano, Breadboard, Breadboard-Powermodul, Ultraschallsensor, Herzschlagsensor, Druck- und Temperatursensor, Kabelsatz

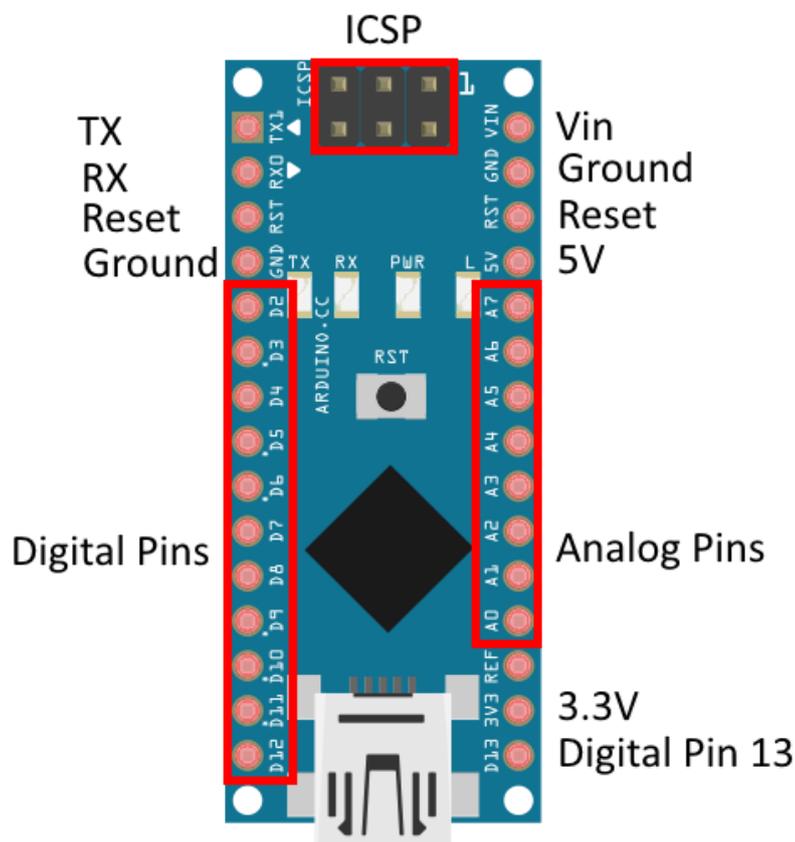
## 2. DER ARDUINO NANO

Das Herzstück dieses Sets ist der Arduino Nano, er eignet sich sehr gut für Breadboardschaltungen, da man ihn direkt auf ein Breadboard stecken kann.

Über die Mini-USB Schnittstelle kann man den Arduino programmieren und mit Strom versorgen.



Dem nachfolgendem Bild können Sie einen detaillierten Anschlussplan für den Arduino Nano entnehmen

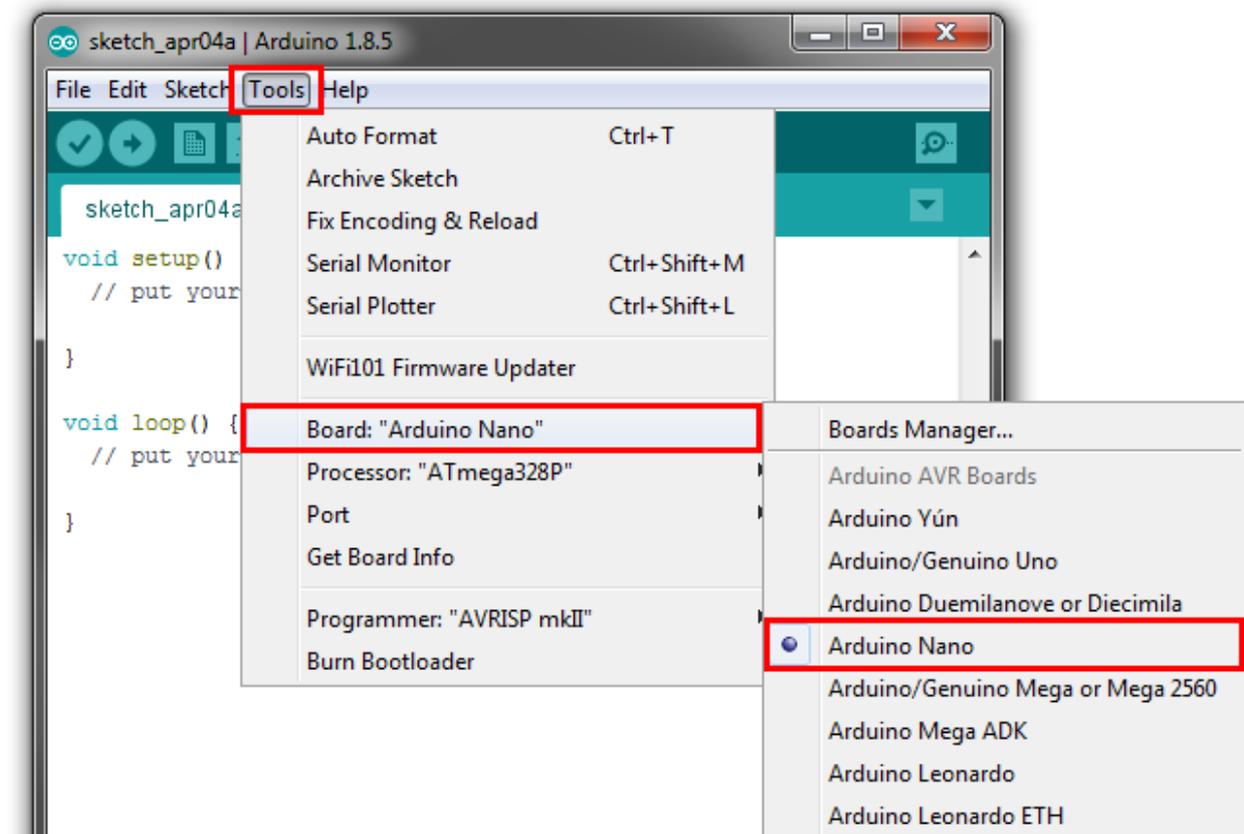


Digital Pin 13 ist bereits intern mit der LED auf dem Arduino Nano verbunden.

Um den Arduino Nano programmieren zu können, benötigen Sie zunächst die entsprechende Arduino Software. Diese ist [hier](#) erhältlich. Laden Sie das Softwarepaket (Arduino IDE) herunter und installieren Sie dieses.

Nachdem Sie die Software gestartet haben muss diese auf das Board eingestellt werden.

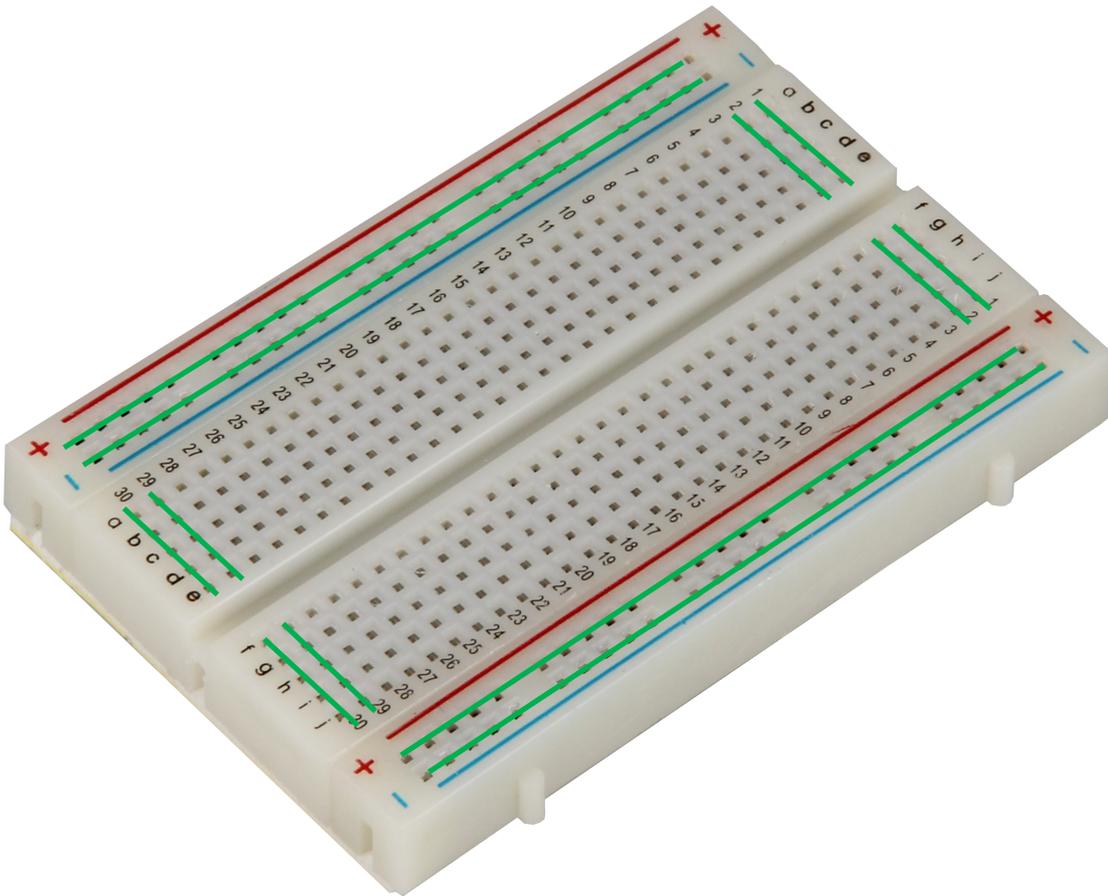
Öffnen Sie dazu den Reiter „**Tools**“ und wählen Sie unter „**Boards**“ das „**Arduino Nano**“-Board aus.



### 3. DAS BREADBOARD

Mit Hilfe des Breadboards lassen sich benutzerdefinierte Schaltungen einfach und übersichtlich gestalten.

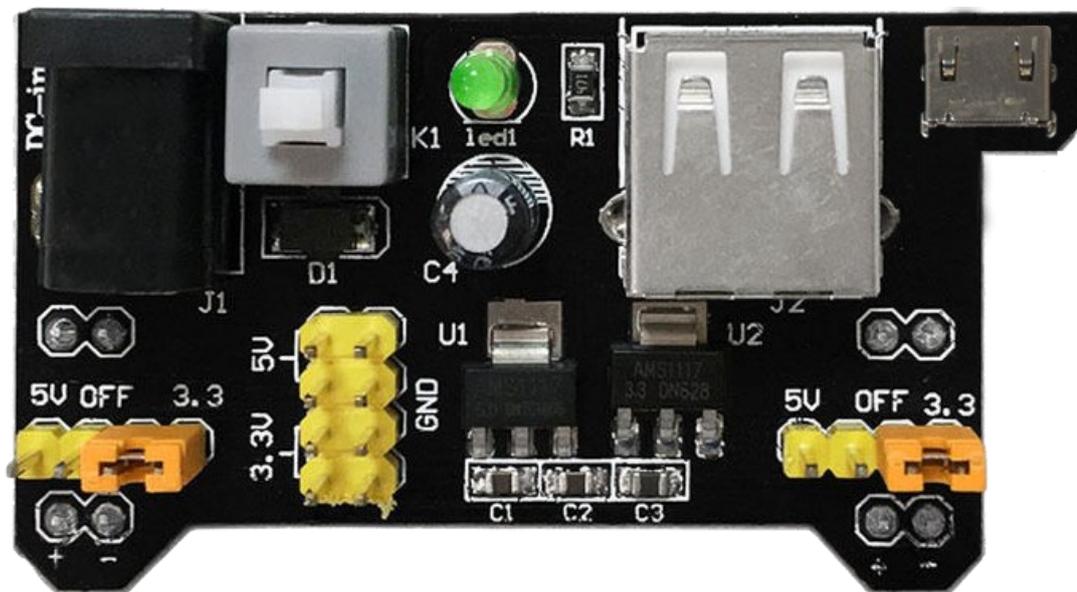
Die Spalten + und - sind jeweils senkrecht durchverbunden.



Die Zeilen sind jeweils von a bis e und von f bis j waagrecht durchverbunden

## 4. DAS BREADBOARD-POWERMODUL

Dieses Modul erlaubt Ihnen die einfache Spannungsversorgung Ihrer eigenen Experimente oder Versuchsaufbauten auf Breadboards.



Die Platine wird einfach aufgesteckt und versorgt, dann das Breadboard mit 3,3 und 5V, die Spannung ist für jede Seite einzeln einstellbar.

Zusätzlich stehen auch 3,3 sowie 5V und Masse auf Pins zur Abnahme bereit.

Der maximale Ausgangsstrom liegt bei 700 mA

Die Stromversorgung der Platine dank des Micro USB Ports über die meisten Handyladegeräte erfolgen, zusätzlich steht auch ein Standard USB und ein DC IN Eingang zur Verfügung.

Wird der DC IN als Eingangsquelle genutzt können die USB Ports auch zur Stromentnahme genutzt werden.

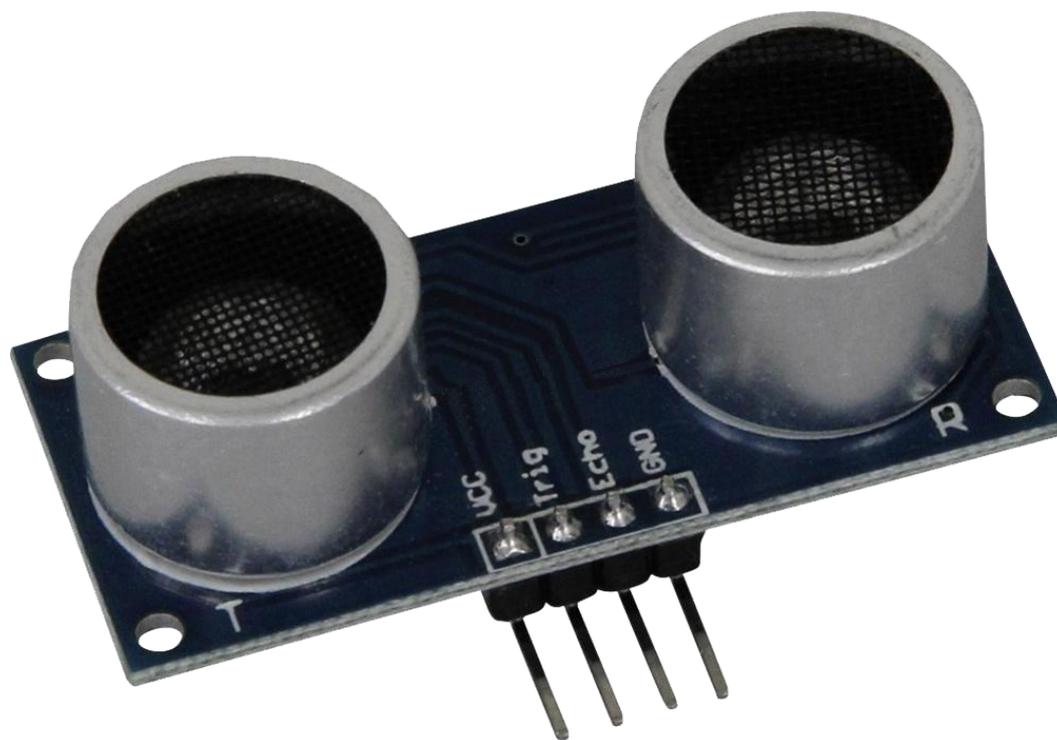
Über den Schalter können Sie die Eingangsquelle zwischen USB oder DC IN wählen.

## 5. DER ULTRASCHALLSENSOR

Der Ultraschallsensor eignet sich hervorragend zum Messen von Entfernungen.

Er wird auch sehr häufig in der Automobilindustrie für Parksensoren eingesetzt.

Der Sensor gibt ein Ultraschallsignal aus und empfängt das Echo, aus der Verzögerung zwischen den beiden Signalen kann er die Distanz bestimmen.



Anschlussbelegung

| Arduino Nano | Ultraschall |
|--------------|-------------|
| +5V          | VCC         |
| Pin 7        | Echo        |
| Pin 8        | Trigger     |
| GND          | GND         |

Dieser Beispielcode dient zum Messen von Entfernungen zwischen 2cm und 300cm.  
Bitte kopieren Sie den folgenden Code und fügen ihn vollständig in Ihren Sketch ein.

```
#define Echo_EingangsPin 7 // Echo Eingangs-Pin
#define Trigger_AusgangsPin 8 // Trigger Ausgangs-Pin

// Benoetigte Variablen werden definiert
int maximumRange = 300;
int minimumRange = 2;
long Abstand;
long Dauer;

void setup() {
  pinMode(Trigger_AusgangsPin, OUTPUT);
  pinMode(Echo_EingangsPin, INPUT);
  Serial.begin (9600);
}

void loop() {

  // Abstandsmessung wird mittels des 10us langen Triggersignals gestartet
  digitalWrite(Trigger_AusgangsPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trigger_AusgangsPin, LOW);

  // Nun wird am Echo-Eingang gewartet, bis das Signal aktiviert wurde
  // und danach die Zeit gemessen, wie lang es aktiviert bleibt
  Dauer = pulseIn(Echo_EingangsPin, HIGH);

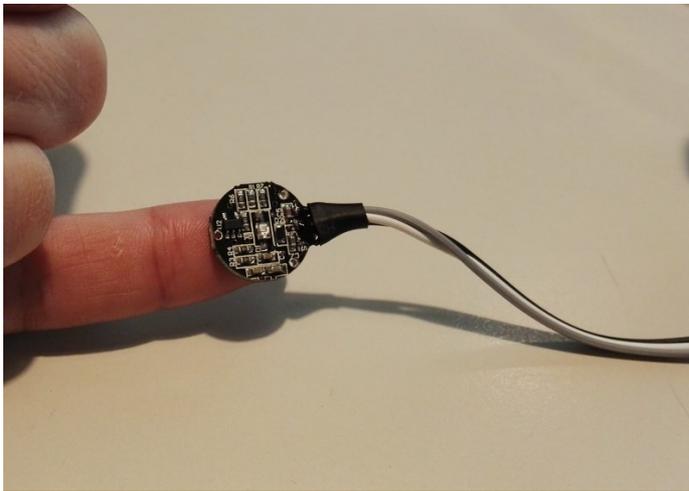
  // Nun wird der Abstand mittels der aufgenommenen Zeit berechnet
  Abstand = Dauer/58.2;

  // Überprüfung ob gemessener Wert innerhalb der zulässigen Entfernung liegt
  if (Abstand >= maximumRange || Abstand <= minimumRange)
  {
    // Falls nicht wird eine Fehlermeldung ausgegeben.
    Serial.println("Abstand außerhalb des Messbereichs");
    Serial.println("-----");
  }

  else
  {
    // Der berechnete Abstand wird in der seriellen Ausgabe ausgegeben
    Serial.print("Der Abstand betraegt:");
    Serial.print(Abstand);
    Serial.println("cm");
    Serial.println("-----");
  }
  // Pause zwischen den einzelnen Messungen
  delay(500);
}
```

## 6. DER HERZSCHLAGSENSOR

Dieser Herzschlagsensor ermittelt mit Hilfe einer Infrarot-Leuchtdiode und eines Foto-Transistors den Puls des Benutzers.



Um die Genauigkeit des Sensors zu erhöhen, empfehlen wir den Sensor mittels eines Pflasters / Klebebandes / Isolierbandes an der Fingerkuppe zu fixieren.

| Arduino Nano | Herzschlagsensor |
|--------------|------------------|
| +5V          | VCC (Weiß)       |
| Analog Pin 0 | Signal (Schwarz) |
| GND          | GND (Grau)       |

## Codebeispiel:

```
////////////////////////////////////
/// Copyright (c)2015 Dan Truong
/// Permission is granted to use this software under the MIT
/// licence, with my name and copyright kept in source code
/// http://http://opensource.org/licenses/MIT
///
/// KY039 Arduino Heartrate Monitor V1.0 (April 02, 2015)
////////////////////////////////////

// German Comments by Joy-IT
////////////////////////////////////
/// @param[in] IRSensorPin Analog PI an welchen der Sensor angeschlossen ist
/// @param[in] delay (msec) Die Verzoegerung zwischen den Aufrufen der Abtastfunktion.
//      Die besten Ergebnisse erhaelt man, wenn man 5 mal Pro Herzschlag abtastet.
///      Nicht langsamer als 150mSec für z.B. 70 BPM Puls
///      Besser waere 60 mSec für z.B. bis zu einen Puls von 200 BPM.
///
/// @Kurzbeschreibung
/// Dieser Code stellt eine sog. Peak-Detection dar.
/// Es wird kein Herzschlagverlauf aufgezeichnet, sondern es
/// wird innerhalb der aufgezeichneten Daten nach "Peaks" (Spitzen) gesucht,
/// und per LED angezeigt. Mittels der bekannten Delay Abstaende, kann somit
/// grob der Puls errechnet werden.
////////////////////////////////////

int rawValue;

bool heartbeatDetected(int IRSensorPin, int delay)
{
    static int maxValue = 0;
    static bool isPeak = false;

    bool result = false;

    rawValue = analogRead(IRSensorPin);
    // Hier wird der aktuelle Spannungswert am Fototransistor ausgelesen und in der rawValue-Variable zwischengespeichert
    rawValue *= (1000/delay);

    // Sollte der aktuelle Wert vom letzten maximalen Wert zu weit abweichen
    // (z.B. da der Finger neu aufgesetzt oder weggenommen wurde)
    // So wird der MaxValue resetiert, um eine neue Basis zu erhalten.
    if (rawValue * 4L < maxValue) { maxValue = rawValue * 0.8; } // Detect new peak
    if (rawValue > maxValue - (1000/delay)) {
        // Hier wird der eigentliche Peak detektiert. Sollte ein neuer RawValue groeßer sein
        // als der letzte maximale Wert, so wird das als Spitze der aufgezeichnten Daten erkannt.
        if (rawValue > maxValue) {
            maxValue = rawValue;
        }
    }
}
```

## Codebeispiel fortgeführt:

```
//
ein Herzschlag zugewiesen werden
if (isPeak == false) {
    result = true;
}
isPeak = true;
} else if (rawValue < maxValue - (3000/delay)) {
isPeak = false;
// Hierbei wird der maximale Wert bei jeden Durchlauf
// etwas wieder herabgesetzt. Dies hat den Grund, dass
// nicht nur der Wert sonst immer stabil bei jedem Schlag
// gleich oder kleiner werden wuerde, sondern auch,
// falls der Finger sich minimal bewegen sollte und somit
// das Signal generell schwaecher werden wuerde.
maxValue-=(1000/delay);
}
return result;
}

// Arduino main code
int ledPin=13;
int analogPin=0;

void setup()
{
// Die eingebaute Arduino LED (Digital 13), wird hier zur Ausgabe genutzt
pinMode(ledPin,OUTPUT);

// Serielle Ausgabe Initialisierung
Serial.begin(9600);
Serial.println("Heartbeat Detektion Beispielcode.");
}
const int delayMsec = 60; // 100msec per sample
// Das Hauptprogramm hat zwei Aufgaben:
// - Wird ein Herzschlag erkannt, so blinkt die LED kurz aufgesetzt
// - Der Puls wird errechnet und auf der serriellen Ausgabe ausgegeben.

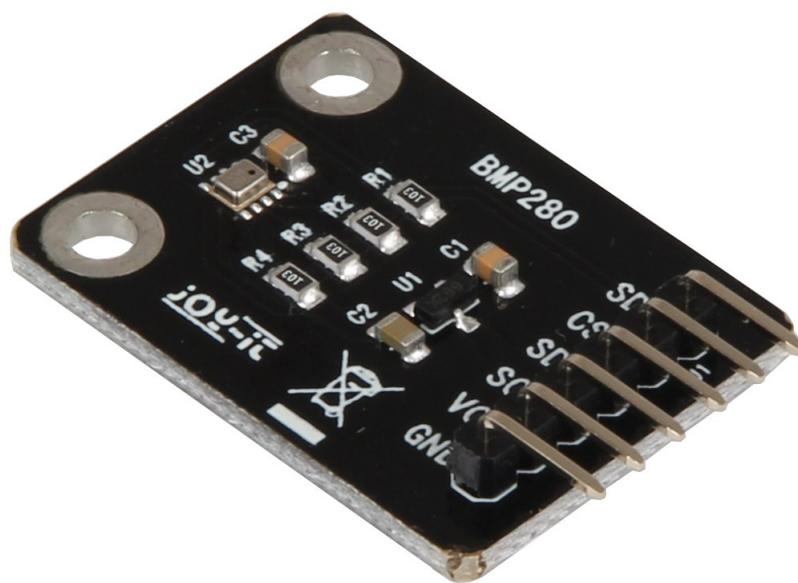
void loop()
{
static int beatMsec = 0;
int heartRateBPM = 0;
if (heartbeatDetected(analogPin, delayMsec)) {
    heartRateBPM = 60000 / beatMsec;
    // LED-Ausgabe bei Herzschlag
    digitalWrite(ledPin,1);
    // Serielle Datenausgabe
    Serial.print("Puls erkannt: ");
    Serial.println(heartRateBPM);
    beatMsec = 0;
} else {
    digitalWrite(ledPin,0);
}
delay(delayMsec);
beatMsec += delayMsec;
}
```

Zum erkannten Peak soll nur

## 7. DER DRUCK- UND TEMPERATUR SENSOR

Dieser Sensor vereint gleich zwei Funktionen. Er kann sowohl den Luftdruck, als auch die Temperatur erfassen und eignet sich damit hervorragend für zum Beispiel eine selbst gebaute Wetterstation.

Der Sensor kann über I<sup>2</sup>C oder SPi angesteuert werden.



Anschlussbelegung

| Arduino Nano | Herzschlagsensor |
|--------------|------------------|
| +5V          | SDO              |
| +5V          | CSB              |
| +5V          | VCC              |
| GND          | GND              |
| SDA          | Analog Pin 4     |
| SCL          | Analog Pin 5     |

Bitte kopieren sie den folgenden Code vollständig und fügen Sie ihn in Ihr Skript ein.

```
/*
*****
This is a library for the BMP280 humidity, temperature & pressure sensor

Designed specifically to work with the Adafruit BMEP280 Breakout
----> http://www.adafruit.com/products/2651

These sensors use I2C or SPI to communicate, 2 or 4 pins are required
to interface.

Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing products
from Adafruit!

Written by Limor Fried & Kevin Townsend for Adafruit Industries.
BSD license, all text above must be included in any redistribution
*****
#include
#include
#include
#include

#define BMP_SCK 13
#define BMP_MISO 12
#define BMP_MOSI 11
#define BMP_CS 10

Adafruit_BMP280 bmp; // I2C
//Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
//Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);

void setup() {
  Serial.begin(9600);
  Serial.println(F("BMP280 test"));

  if (!bmp.begin()) {
    Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));
    while (1);
  }
}

void loop() {
  Serial.print(F("Temperature = "));
  Serial.print(bmp.readTemperature());
  Serial.println(" *C");

  Serial.print(F("Pressure = "));
  Serial.print(bmp.readPressure());
  Serial.println(" Pa");
  Serial.print(F("Approx altitude = "));
  Serial.print(bmp.readAltitude(1013.25)); // this should be adjusted to your local forcase
  Serial.println(" m");
  Serial.println();
  delay(2000);
}
```

## 8. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)



### Symbol auf Elektro- und Elektronikgeräten:

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

### Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

### Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

### Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an [Service@joy-it.net](mailto:Service@joy-it.net) oder per Telefon an uns.

### Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

## 9. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: [service@joy-it.net](mailto:service@joy-it.net)

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

[www.joy-it.net](http://www.joy-it.net)