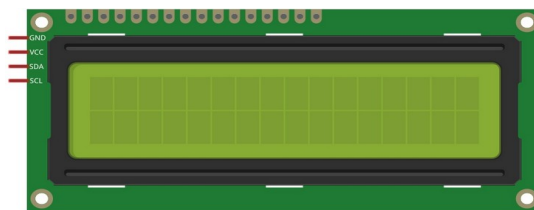
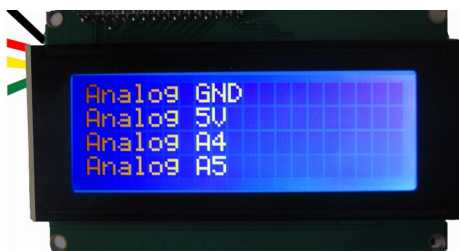


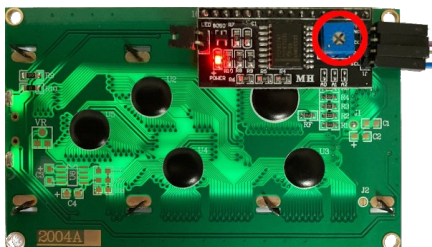
Der Sensor BMP280 kann die Temperatur messen und den Luftdruck bestimmen. Die gemessenen Werte sollen auf einem LCD angezeigt werden.



SchlieÙe das LCD an:



fritzing

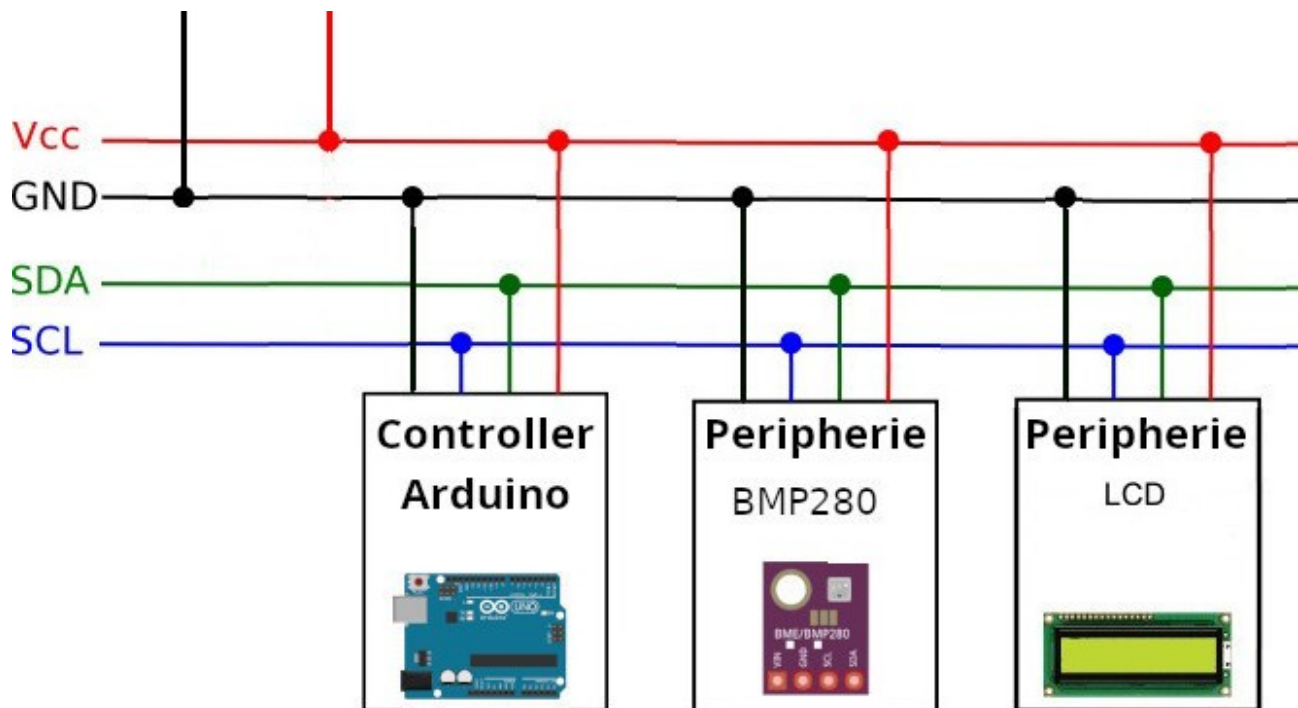


Auf der Rückseite befindet sich ein Potentiometer mit dem die Helligkeit eingestellt werden kann.

Normalerweise wäre eine komplexe Verkabelung zum Betrieb eines LCDs nötig. Der I²C-Bus regelt über einen eigenen Mikroprozessor die Kommunikation der Datenleitungen untereinander. Es werden deshalb nur vier Anschlüsse benötigt. Er wird über I²C angesteuert.

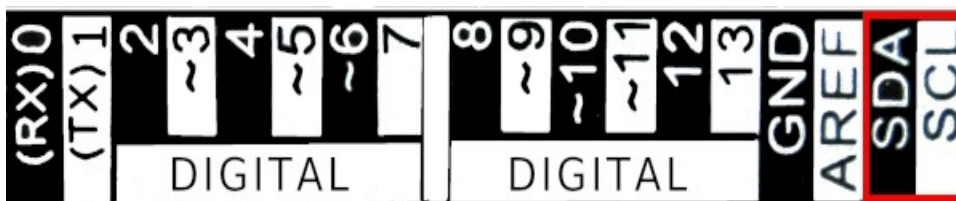


Der I²C-Bus (Inter Integrated Circuit) wurde ursprünglich von Philips entwickelt, er sollte die Kommunikation mit einem Master (dem Arduino) und den verschiedenen Bauelementen (den Slaves) ermöglichen.



Der I²C-Bus kommt mit zwei Datenleitungen aus:

- die Taktleitung SCL (Serial Clock) → A5
- die Datenleitung SDA (Serial Data) → A4



Statt A4 (SDA) und A5 (SCL) kannst du auch die mit SCL und SDA beschrifteten Pins verwenden.

Auf einem LCD sollen Temperatur und Luftdruck angezeigt werden.

Benötigte Bauteile:

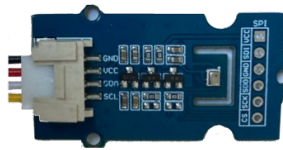
- ➔ LCD I²C
- ➔ Temperatur-/Feuchtigkeitssensor BMP280
- ➔ Leitungsdrähte



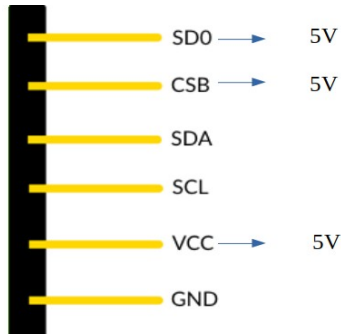
Der BMP280 wird auch in anderen Bauformen mit unterschiedlicher Pinbelegung angeboten.



JOY-IT BMP280



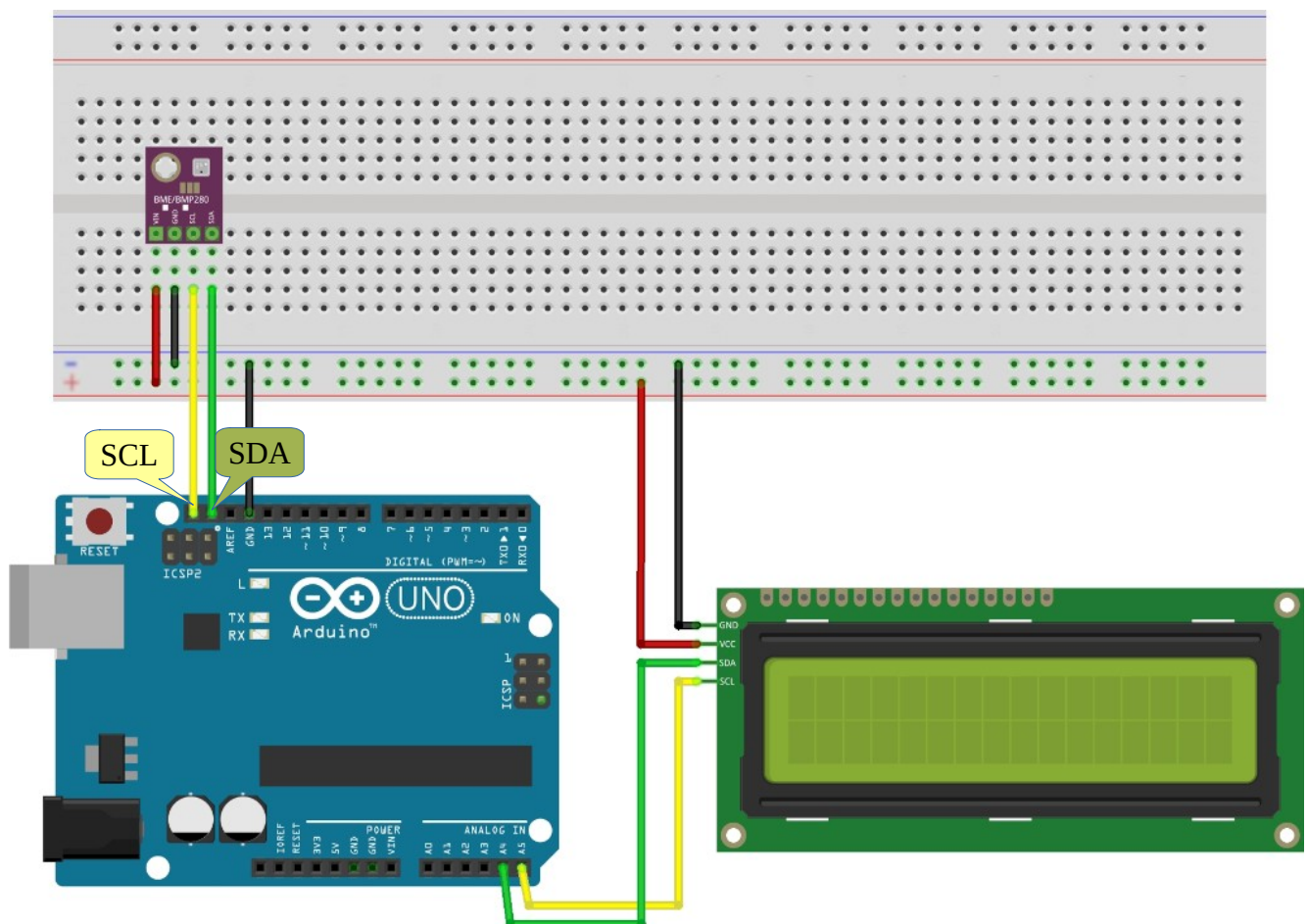
BMP280 mit Grove-Schnittstelle



Bei einem BMP280 mit sechs Pins müssen SDO und CSB ebenfalls an VCC (5V) angeschlossen werden.

Quelle: <https://sensorkit.joy-it.net/de/sensors/ky-052> (abgerufen am 01.03.24)

Baue die Schaltung auf.



Benötigte Bibliotheken:

Sketch → Bibliothek einbinden → Bibliotheken verwalten



Die Standard-HEX-Adresse des BMP280 ist 0x77. In diesem Fall genügt der Aufruf `bmp.begin()`. Wenn das nicht funktioniert, kannst du die HEX-Adresse mit folgenden Programm herausfinden.

```
#include "Wire.h"

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  delay(1000);
  Serial.print("I2C Scanner");
}

void loop()
{
  byte Fehler, Adresse;
  int Geraete = 0;
  Serial.println("Starte Scanvorgang");

  for (Adresse = 1; Adresse < 127; Adresse++ )
  {
    // Übertragung starten
    Wire.beginTransmission(Adresse);

    // wenn die Übertragung beendet wird
    Fehler = Wire.endTransmission();

    if (Fehler == 0)
    {
      Serial.print("I2C Gerät gefunden - Adresse: 0x");
      if (Adresse < 16) Serial.print("0");
      Serial.print(Adresse, HEX);
      Serial.println("");
      Geraete++;
    }
  }
}
```

```

if (Geraete == 0) Serial.println("Keine I2C Geräte gefunden\n");
else Serial.println("Scanvorgang abgeschlossen");

delay(5000);
}

```

Dem Aufruf von `bmp.begin` muss dann in den Klammern die HEX-Adresse mitgeteilt werden.

Binde die benötigten Bibliotheken und definiere die Bauteile.

```

# include <Adafruit_BMP280.h>
# include <LiquidCrystal_I2C.h>

// LCD definieren
// 0x27 -> Hex-Adresse, 20 Zeichen, 4 Zeilen
LiquidCrystal_I2C lcd(0x27, 20, 4);

Adafruit_BMP280 bmp;

```

Im `setup`-Teil werden das LCD und der BMP280 gestartet:

```

void setup()
{
  // BMP280 starten
  bmp.begin();

  Serial.begin(9600);
  // LCD starten
  lcd.init();
  lcd.backlight();
}

```

Im `loop`-Teil wird die Temperatur und der Luftdruck gemessen. Beachte die Kommentare.

```

void loop()
{
  // readTemperature() Temperatur messen und Messergebnis formatieren
  String Temperatur = String(bmp.readTemperature());

  // . durch , ersetzen
  Temperatur.replace(".", ",");

  /*
    readPressure() Luftdruck messen und Messergebnis formatieren
    readPressure() liest in Pascal, ausgabe in hPa (Hekto-Pascal)
    Ergebnis durch 100 teilen
  */
  String Luftdruck = String(bmp.readPressure() / 100);
  Luftdruck.replace(".", ",");

  // Ausgabe Serieller Monitor
  Serial.println("Temperatur: " + Temperatur + "°C");
  Serial.println("Luftdruck: " + Luftdruck + " hPa");
}

```

```
// Ausgabe LCD
lcd.setCursor(0, 0);
lcd.print("Temperatur: ");
lcd.setCursor(0, 1);

// \337C -> °
lcd.print(Temperatur + "\337C");

lcd.setCursor(0, 2);
lcd.print("Luftdruck: ");
lcd.setCursor(0, 3);
lcd.print(Luftdruck + " hPa");

delay(3000);
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 18.06.25