

# Temperatur, Luftfeuchte und Luftdruck:

## Spielereien mit dem Wettersensor BME280 und Arduino

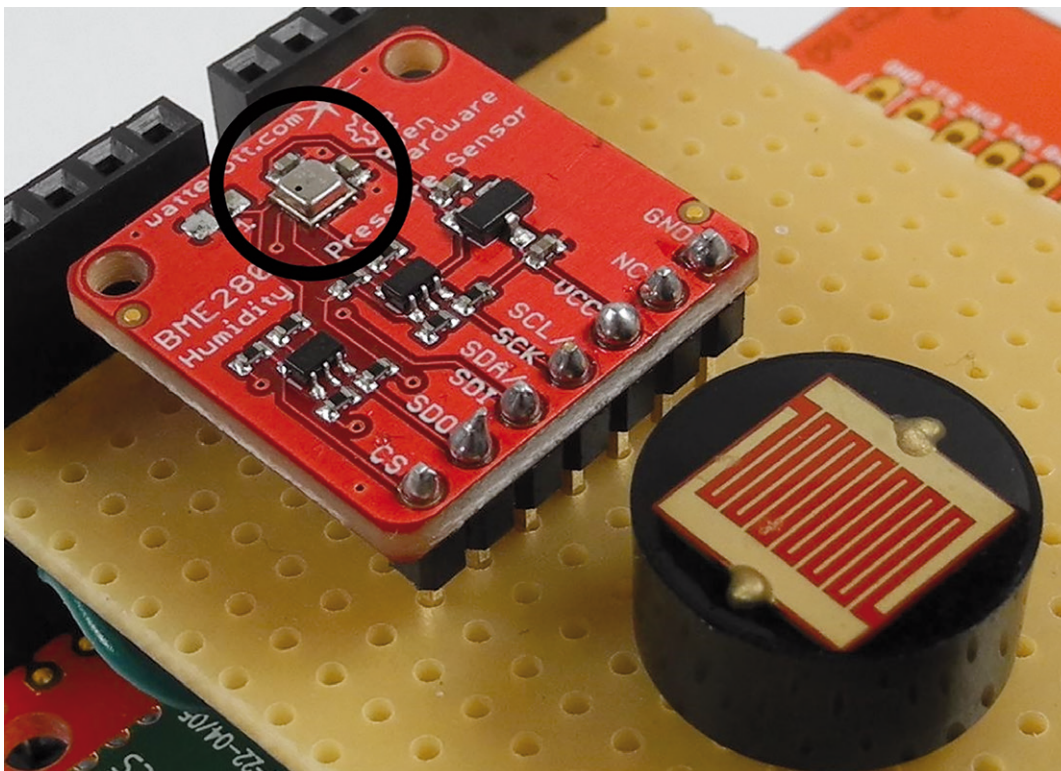


Bild 1: BME280 Dreifachsensor für Temperatur, Luftfeuchte und Luftdruck auf dem Breakout Board.

*Der Bosch-Sensor BME280 misst auf kleinstem Raum Temperatur, Luftfeuchte und Luftdruck. Das reizt zu einigen Spielereien. Mögliche Anwendungen sind eine Wetterstation, Datenlogger oder ein Höhenmesser. Lesen Sie, wie man den BME280 mit einem Arduino UNO ansteuert und mit recht wenig Aufwand erstaunliche Anwendungen realisiert.*

Nach dem noch vergleichsweise voluminösen Sensor BMP085, dem kleineren BMP180 (2011) hat Bosch 2015 der neuesten Kreation die Bezeichnung BMP280 gegeben - ausgestattet mit einer höheren Präzision als seine Vorgänger und alles in ein noch winzigeres Gehäuse untergebracht. Der Baustein selbst misst nur 2,5 mal 2,5 mm, ist nicht einmal einen Millimeter hoch und besitzt auf der Unterseite acht für den LötKolben unerreichbare Anschlüsse. Glücklicherweise gibt es den BME280 fertig montiert zusammen mit einigen sinnvollen Bauteilen auf einer kleinen Platine, als sogenanntes Breakout Boards fertig zu kaufen. Der Sensor erfreut mit

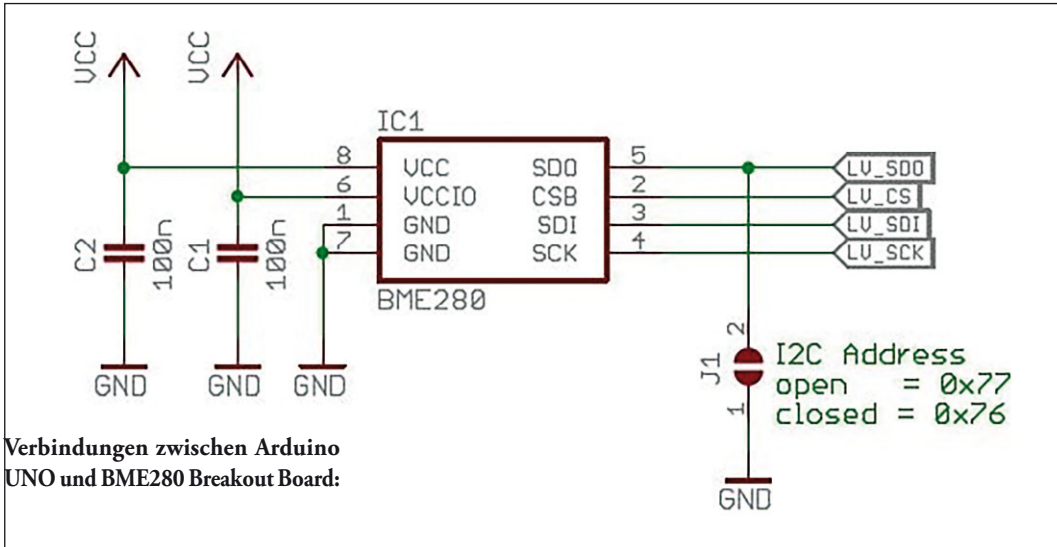
einer hohen Präzision: Die Messung der Luftfeuchtigkeit erfolgt mit einer Toleranz von nur  $\pm 3\%$ , die Temperatur ist auf  $\pm 1.0^\circ\text{C}$  exakt – sagt das Datenblatt - und die Ermittlung des Luftdrucks gelingt mit einer Auflösung von  $\pm 1$  hPa (Hektopascal). Von einem Mikrocontroller (oder wie hier einem Arduino UNO mit Atmel ATMEGA328) lässt sich der Baustein wahlweise über I2C (Adresse 0x76 oder 0x77) oder per SPI ansprechen. Für den Arduino nutzen wir I2C, sollen jedoch mehrere BME280-Sensoren in einem Projekt verbaut werden, bietet sich das schnellere SPI-Interface an. Die Betriebsspannung beträgt 1,71 bis 3,6 Volt und verbraucht während

einer Messung maximal  $3,6 \mu\text{A}$  bei einer Messfrequenz von einem Hertz. Im „Schlafmodus“ sind es gerade einmal  $0,1 \mu\text{A}$ . Die Antwortzeit des Sensors auf eine Messanfrage dauert etwa 1 Sekunde - somit sind max. 60 Messungen je Minute möglich. Doch so fix ändern sich weder Temperatur noch Luftdruck oder die Feuchte. Mit einer Ausnahme: Als Höhenmesser reagiert der BME280 auf Höhenänderungen von einigen Metern. Bei Messungen im 1-Herztakt kann man beim Treppensteigen die Veränderung des Luftdrucks beobachten.

Für den BME280 existieren Breakaout-Boards unterschiedlicher Hersteller. Ich habe mich für

das Produkt von Watterott [1] aus hiesigen Landen entschieden: Neben dem Sensor selbst sind Level-Shifters für die Signale und ein 3,3-Volt-Regler bereits vorhanden. Zudem lässt mir diese BME280-Platine die Wahl, den Baustein mit SPI oder I2C anzusteuern. Die Ansteuerung kann ohne Änderungen der Hardware wahlweise mit 3,3 Volt oder – gut für den Arduino UNO – mit 5 Volt erfolgen, und das, ohne Gedanken an korrekte Pegel zu verschwenden.

Wie Tabelle 1 zeigt, genügen vier Verbindungen zwischen Arduino und der Sensorplatine. Letztere habe ich auf ein selbstgebautes, provisorisch



Verbindungen zwischen Arduino UNO und BME280 Breakout Board:

Bild 2: Ausschnitt aus dem Schaltplan des BME280 Breakout Boards: Anschlüsse des BME280. Quelle: [1]

Arduino	BME280-Platine
3.3 Volt <->	VCC
GND <->	GND
A4/SDA <->	SDA
A5/SCL <->	SCL

sches Arduino-Shield (Aufsteckplatine) gelötet, das ich aus einem Rest einer vorhandenen Lochrasterplatine und den benötigten Steckerleisten gefertigt habe. Damit „thront“ das Arduino-Shield mechanisch stabil über dem Arduino UNO - mit dem Vorteil kurzer Signalverbindungen.

Fast kann das erste Beispielprogramm auf den Arduino geladen und gestartet werden. Doch zuvor ist die Programmibliothek für den BME280 zu installieren. Dieses Programmpaket wurde von der englischen Firma Embedded Adventures

entwickelt und ist nach GNU-Lizenz für nichtkommerzielle Nutzung freigegeben. Die Library ist unter [2] zu laden, befindet sich aber auch auf der Heft-DVD im Verzeichnis *.Artikel im Heft\BME280*. Dort ist auch bereits die folgende kleine Anpassung für das hier verwendete Breakout Board realisiert:

```
// address of BME280 MOD-1022 board
#define addrBME280 0x76

// address of BME280 Watterott board
#define addrBME280 0x77
```

Die Änderung in der Datei *BME280\_MOD-1022.h* wird nötig, weil das Board der Firma Embedded Adventures die I2C-Adresse 0x76 nutzt, das hier genutzte Watterott-Board jedoch als Standard 0x77 (Jumper J1 bleibt dabei unverändert

entw.). Das Verzeichnis der Bibliothek *BME280-master* wird mit ihren Dateien nun in das Verzeichnis *libraries* der Arduino-Installation hineinkopiert (üblicherweise *C:\Program Files\Arduino\libraries*). Startet man daraufhin die Arduino-IDE, kann man unter *Examples BME280-test.ino* laden, kompilieren und starten. Auf der DVD finden Sie zudem ein ähnliches Sketch (Programm) unter

## ...und nun: Anzeige auf einer LCD

Schöner wäre es, wenn man Sensor und Arduino unabhängig vom PC betreiben könnte, dabei bietet sich die Anzeige der Messwerte auf einer LC-Anzeige an. Ein LCD mit 16 Zeichen und zwei Zeilen lag noch in der Bastelkiste und wurde wie folgt an den Arduino UNO angeschlossen:

So ist das LCD mit dem Arduino verbunden:

LCD	Arduino UNO
Pin 1 VSS	GND
Pin 2 VCC	5V
Pin 4 RS	Digital Pin 13
Pin 5 R/W	GND
Pin 6 Enable	Digital Pin 12
Pin 11 D4	Digital Pin 11
Pin 12 D5	Digital Pin 10
Pin 13 D6	Digital Pin 9
Pin 14 D7	Digital Pin 8

Bild 3: Das Selbstbau-Shield trägt neben der BME-280-Platine einen lichtempfindlichen Widerstand (LDR), um grob die Helligkeit der Umgebung zu erfassen (Tag, Nacht, Dämmerung etc.). Gemeinsam mit einem Widerstand bildet der LDR einen einfachen Spannungsteiler.

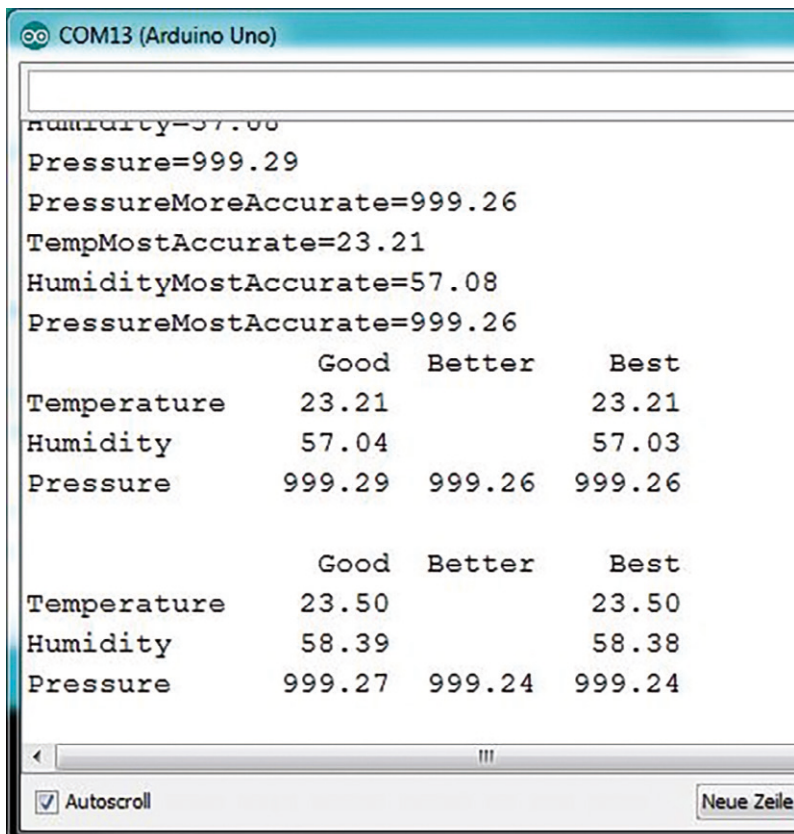
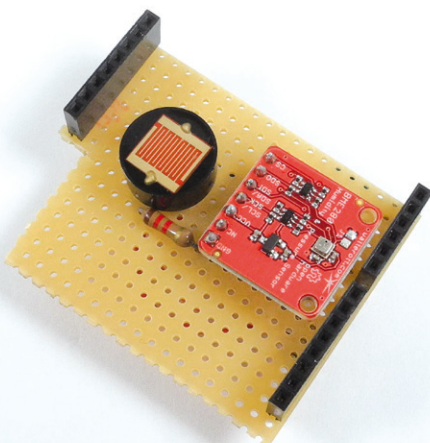
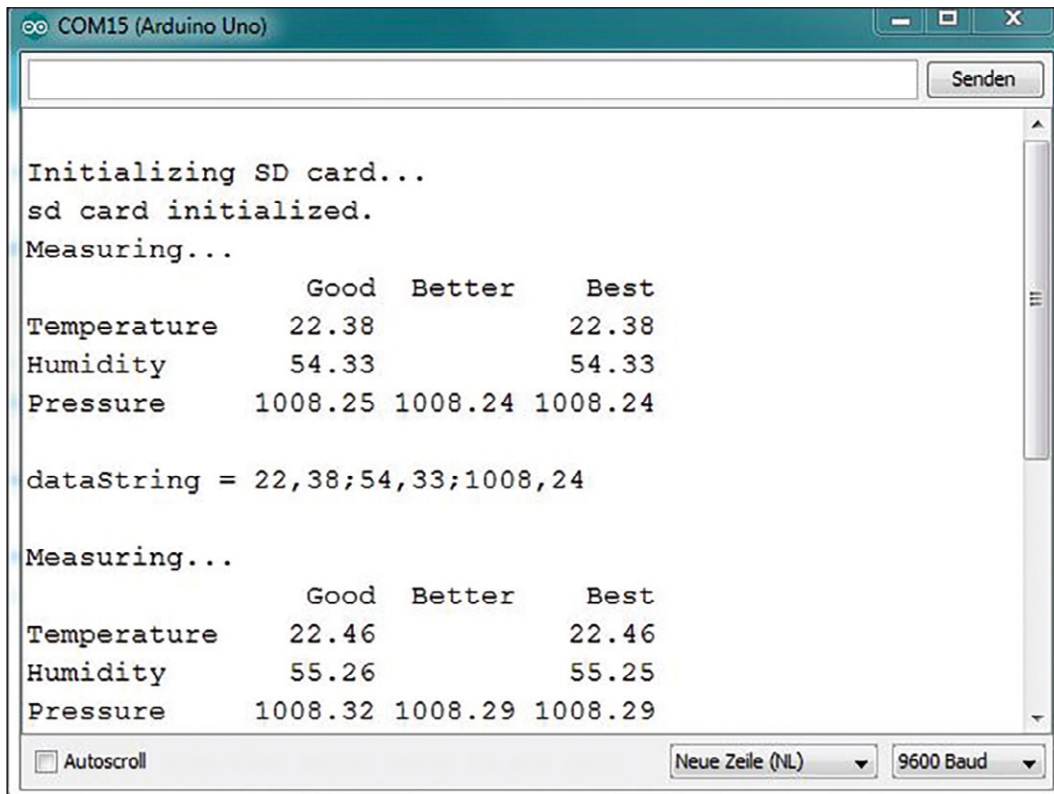


Bild 4: Erste Ausgabe der Messwerte über die serielle Schnittstelle zum seriellen Monitor.





**Bild 8:** An den Ausgaben des seriellen Monitors kann man den Programmverlauf kontrollieren. Die Zeile *datastring* schreibt der Datenlogger in die Datei *WXDATA.CSV* auf die SD-Karte.

wird, bedeutet ein hoher gemessener Wert eine hohe Umgebungshelligkeit, ein geringer Wert Dämmerung bzw. Dunkelheit. Im Programm findet man zudem die Zuordnung der gemessenen Helligkeitswerte zu Aussagen wie „Dämmerung“ oder „Sonne“. Diese sehr grobe Bewertung ist anhand eigener Erfahrungswerte dem jeweiligen Aufstellungsort anzupassen.

## ...und ein Datenlogger

Eine interessante Anwendung des BME280-Sensors ist ein transportabler Datenlogger, den man für einige Zeit an einen interessanten Ort deponiert und dort relevante Wetterdaten aufzeichnen lässt. Mit einem Arduino UNO, dem Arduino Ethernet-Shield und unserem kleinen Selbstbau-Shield mit dem BME280 und der LDR darauf ist das leicht möglich. Das Arduino Ethernet Shield kommt hier deshalb zum Einsatz, da es im Gegensatz um RedFly WiFi-Shield über einen SD-Kartenslot verfügt. Ich verwende eine 2GB-SD-Karte, die unter Windows als FAT16 formatiert wurde und deren Kapazität

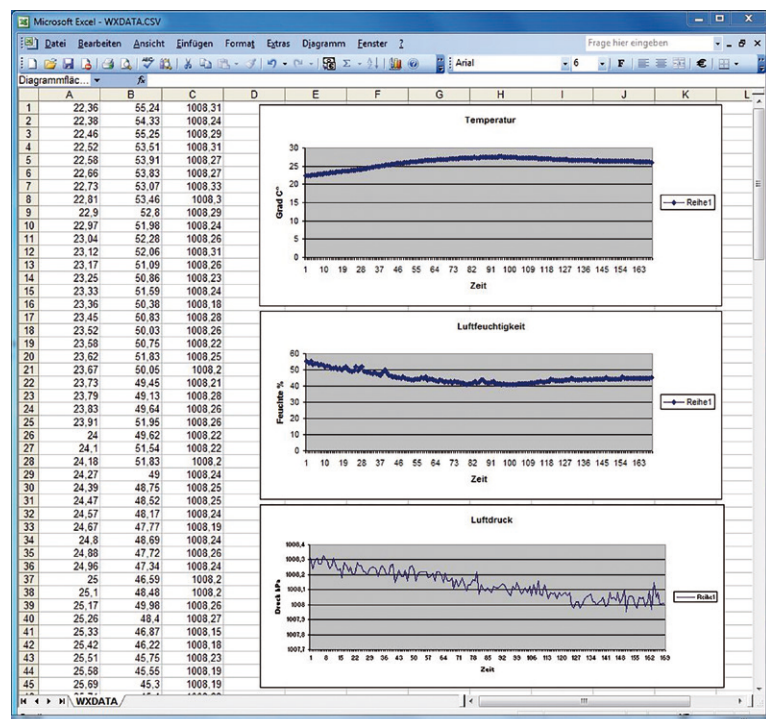
dazu ausreicht, über mehrere Tage oder Wochen eine Aufzeichnung der Daten zu ermöglichen.

Die Aufzeichnung der Daten des BME280-Sensors erfolgt in einem CSV-Format, das sich ohne weitere Bearbeitung in eine deutschsprachige MS Excel-Version importieren lässt. Dazu werden die gelesenen Messwerte aus dem englischen Format (Punkt als Dezimaltrenner) zu Komma umformatiert. Die Datenfelder werden durch Semikolon getrennt und sind damit kompatibel zu Excel. Die linke Spalte der Excel-Tabelle enthält die Temperatur, die mittlere die Luftfeuchte und rechts ist der Luftdruck zu sehen. Die Datei *BME280\_Datenlogger\_SD\_Card.ino* finden Sie auf der Heft-DVD.

Nach der Messung „im Feld“ wird zuerst die Spannungsversorgung abgeklemmt, anschließend die SD-Karte dem Ethernet-Shield entnommen. Auf dem PC kann die Datei *WXDATA.CSV* in Excel (oder anderen Kalkulationsprogrammen) geöffnet werden. Nach einigen wenigen Mausklicks zur Einfügung dreier Diagramme kann man in der grafischen Darstellung leicht den Trend von Temperatur, Luft-

feuchte und -druck erlesen. Eine Programmversion, die zudem die Lichtstärkemessung über die LDR in einer vierten Spalte mit einbezieht, ist auf der Heft-DVD unter der Bezeichnung *BME280\_LDR\_Datenlogger\_SD\_Card.ino* abgelegt.

**Bild 9:** Auswertung der Datei *WXDATA.CSV* in Excel. Aus den Spalten lassen sich leicht Diagramme fertigen, welche den Trend der Messwerte grafisch deutlich herausstreifen lassen.



## ...und nun UDP über das Netzwerk

Nicht immer möchte man die Daten des Sensors aufzeichnen, andererseits ist es mühsam, zum Ablesen in den Keller zu gehen oder auf den Dachboden zu steigen und auf das LCD zu schauen. Besser wäre es, wenn man vom PC aus den BME280 ständig abfragen könnte. Das leistet bereits der Webserver, doch ist es lästig, dort immer wieder R5 für ein Auffrischen der Webseite zu drücken. Es geht auch anders: Man kann sich ein Datenpaket über das Netzwerk (LAN) auf den eigenen PC senden lassen. Ein kleines Programm läuft in einer Ecke des Bildschirms und zeigt die jeweils aktuellen Werte an. Auf der Arduino-Seite benötigt man dazu dieselbe Hardwareausstattung wie für den Datenlogger: Arduino UNO, das Arduino Ethernet-Shield und das Selbstbau Shield mit BME280 und LDR.

Die Daten sendet der Arduino UNO als UDP-Datenpakete in regelmäßigen Abständen. UDP ist eine einfache und verbindungslose Verbindungsart, die TCP/IP als Grundlage nutzt. Ein in C# geschriebenes PC-Programm für Windows empfängt die Datenpakete und stellt sie als Temperatur etc. auf dem Bildschirm dar. Zudem kann man in der Listbox links die eintreffenden Datenpakete beobachten.

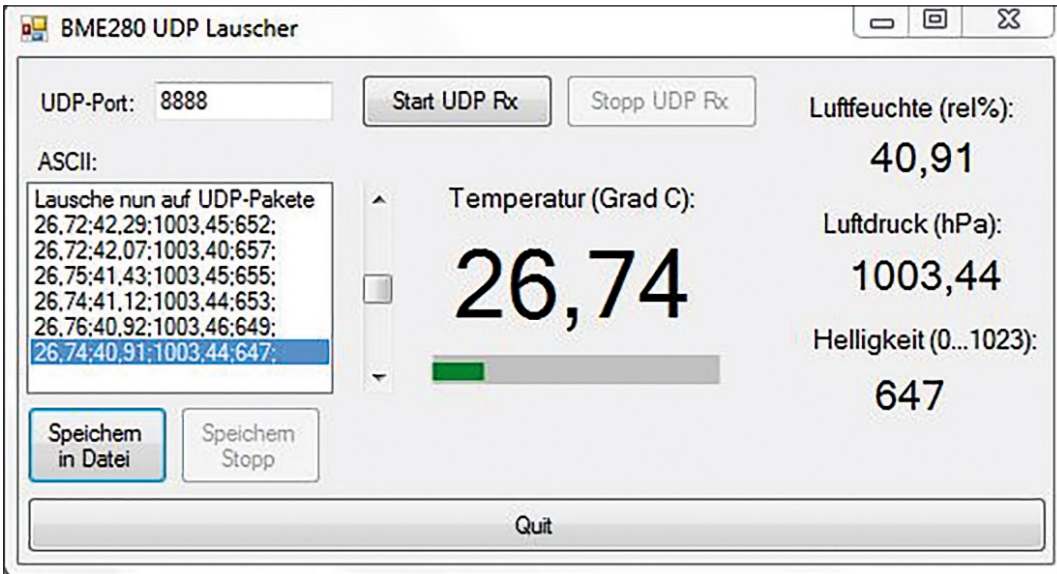


Bild 10: Der Lauscher hört am Port 8888 auf UDP-Datenpakete vom Arduino. Wer möchte, kann die empfangenen Daten in die Datei WXDATA.CSV speichern.

Wie beim Datenlogger beschrieben, besteht auch auf dem Windows-PC die Möglichkeit, die eintreffenden Daten in die Datei WXDATA.CSV zu sichern. Es wird die in Programmverzeichnis bestehende Datei am Ende erweitert, besteht die Datei nicht, wird sie im selben Verzeichnis erzeugt. Auch diese WXDATA.CSV lässt sich problemlos in eine deutsche Excel-Version importieren. Wem der Dateiname nicht gefällt, kann ihn in den Arduino Sketches und in den C#-Programmen verändern. Letztere wurden mit MS Visual Studio 2010 Express (kostenfrei) unter .NET 4.0 erstellt.

Damit der Transfer der Datenpakete funktionieren kann, sind im Arduino-Sketch *BME280\_UDP\_Ethernet.ino* einige Zeilen an die Gegebenheiten des eigenen LANs anzupassen:

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x9A, 0xB2 };
```

In obiger Zeile ist die MAC-Adresse Ihres Arduino Ethernet-Shields einzutragen. Man kann sie dem kleinen Aufkleber, der sich auf der Platine des Ethernet-Shields befindet, entnehmen.

```
IPAddress ip(192, 168, 178, 33);
```

Hier ist die IP-Adresse des Arduino einzutragen, passend zum eigenen Netzwerk (LAN). Wählen Sie eine freie IP-Adresse.

```
unsigned int localPort = 8888;
```

Der Arduino hört auf Port 8888. Dieser Wert kann so bleiben. Der PC sendet keine Pakete an den Arduino. Sollten der Arduino dennoch Datenpakete per UDP empfangen, sendet er sie an den Sender zurück mit der Information „acknowledged“.

```
IPAddress empfaenger_ip(192, 168, 178, 24);
```

Hier ist die IP-Adresse des empfangenen Windows-PC einzutragen. Diese Adresse können Sie unter Windows wie folgt in Erfahrung bringen: Unter Windows *CMD.EXE* starten, dort *ipconfig /all* eingeben.

```
unsigned int empfaenger_port = 8888;
```

Auch dieser Port sollte unverändert bleiben. Das PC-Programm (Bild oben) ist ebenfalls auf Port 8888 voreingestellt.

### ...UDP die zweite: mit Redfly

Nicht immer steht ein LAN-Kabel des eigenen Netzwerkes dort bereit, wo man Arduino und BME280 gern positionieren möchte, beispielsweise in einem Kellerraum, auf dem Dachboden oder im Gartenhaus. Glücklicherweise strahlt das WLAN oft erstaunlich weit, sodass man den Sensor auch an recht ausgefallenen Orten positionieren kann, weil man ihn „über Funk“ betreibt. Daher wäre eine zweite UDP-Lösung via WiFi die flexiblere Lösung. In diesem Beispiel wird wieder das RedFly-WiFi-Shield genutzt. Im Sketch *BME280\_UDP\_RedFly\_WLAN.ino* müssen Sie auch hier die IP-Adressen ihrem Netzwerk anpassen sowie SSID (Bezeichnung des WLAN) und das Passwort eingeben. Auf dem PC kommt dasselbe Pro-

gramm wie zuvor zum Einsatz: Entweder der *BME280\_UDP\_Lauscher* oder *BME280\_UDP\_LauscherMini*.

## Erweiterungsfähig

Die Messung von Temperatur, Luftfeuchte und -druck ist für eine Wetterstation nur die halbe Miete. Eine sinnvolle Erweiterung der Schaltung wäre – bei entsprechendem Platz auf dem Balkon oder im Garten – die Messung von Windgeschwindigkeit und -richtung, evtl. sogar die der Regenmenge. Je nach gewünschter Anwendung kommen evtl. völlig andere Sensoren – etwa ein GPS-Empfänger oder eine Echtzeituhr hinzu. Letztere böte die Markierung der Messergebnisse mit einem Zeitstempel. Eine Freigabe des Ports 8888 für UDP auf dem DSL-Modem bzw. Router macht den Arduino auch über das eigenen LAN hinaus im Internet bekannt, etwa zur Überwachung des Klimas im Ferienhaus. Eine interessante Anwendung des BME280-Sensors ist die Veröffentlichung der Messwerte im APRS-Netzwerk. Für die Arduinos existiert auf [3] eine APRS-Bibliothek von Mark Qvist, mit dessen Hilfe man APRS-kompatible AX25-Datenpakete erzeugen und über ein Funkgerät mit 1200 Baud auf der Frequenz 144,8 MHz senden könnte. In diesem Fall träte der Arduino als Wetterstation mit geringem Stromverbrauch auf, z. B. zu betrachten auf der Webseite *aprs.fi*. Die auf der DVD abgelegten und hier erwähnten Arduino-Sketches basieren auf die jeweiligen Beispielprogramme und kombinieren sie auf sinnvolle Weise. Das Kopieren einiger Zeilen von einem Beispielprogramm zu einem anderen gelingt sicherlich auch vielen Nicht-Programmierern. Nehmen Sie das als Anregung, selbst tätig zu werden.

Literatur/Verweise:

- [1] Watterott electronic GmbH, Breitenhoelzer Str. 6, 37327 Leinefelde, Homepage: [www.watterott.com](http://www.watterott.com)
- [2] BME280-Lib für Arduino: <https://github.com/embeddedadventures/BME280>
- [3] Arduino Programmbibliothek für APRS: <https://github.com/markqvist/LibAPRS>

mw

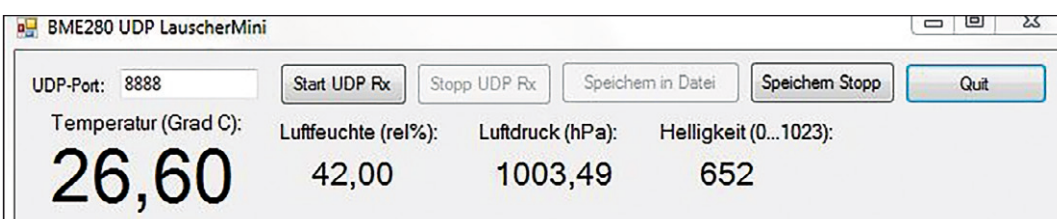


Bild 11: Für Dauerbetrieb: Der LauscherMini benötigt weniger Platz auf dem PC-Monitor bei gleichem Funktionsumfang.