

## WETTERSTATION FÜR ENTWICKLER

# Der Do-it-yourself-Wetterfrosch

Ein meteorologischer Komponentenbausatz für den eigenen Wetterbericht.

**F**aszination Wetter – Bauernregeln können eine gewisse Faszination ausüben. Wem sie zu altertümlich sind oder wer gerne anhand von Messungen den besten Zeitpunkt zum Wohnungslüften finden möchte, der steht früher oder später auf der Suche nach einer Wetterstation vor einem Regal im Laden oder sucht im Internet auf unzähligen Seiten nach einer passenden Wetterstation. Viele Geräte, die es im Handel gibt, können tatsächlich sehr viel und erfassen bereits Unmengen an Daten.

Aber oft müssen Punkte aus dem individuellen Anforderungskatalog gestrichen werden: Eine Wetterstation mit WLAN? Feinstaubmessungen? Die Wetterstation als Basis ins eigene Smart Home einbinden? Individuelle Wünsche lassen sich daher oft nur mit einer Do-it-yourself-Lösung verwirklichen. Wie einfach diese ist, welche Hardware infrage kommt und wie die Programmierung dahinter aussehen kann, erklärt dieser Artikel.

## Wozu überhaupt eine eigene Wetterstation?

Zunächst erscheint es überflüssig, eine eigene Wetterstation zu besitzen, da es ja eine Vielzahl anderer Möglichkeiten gibt, um sich über die aktuellen meteorologischen Verhältnisse zu informieren. Für den einfachen Anwender sollten diese auch ausreichen. Das Internet ist dabei sicherlich eine der am häufigsten genutzten Quellen. Es gibt viele Anbieter, die in Echtzeit und rund um die Uhr Messdaten bereitstellen, und das teilweise auch mit einer einigermaßen genauen Standortangabe. Das Radio oder die Temperaturanzeige im Auto sind zusätzlich häufig genutzte und sehr zuverlässige Quellen, um sich über die aktuelle Lage zu informieren. Auch die Wetterfrösche im Fernsehen sind als Informationsquelle in den meisten Fällen sehr empfehlenswert.

Jedoch sind diese Quellen selten auf die eigenen Bedürfnisse abgestimmt. Als Fernsehzuschauer hat man zudem keine Möglichkeit, die Wetterdaten in digitaler Form zu erhalten, um diese zum Beispiel für eine automatisierte Bewässerung der heimischen Blumenbeete zu nutzen. Das Internet scheint in dieser Hinsicht im Vorteil zu sein.

Ebenso unmöglich ist es, Messdaten etwa aus dem eigenem Gewächshaus mit den verschiedenen Gemüsesorten zu erhalten. Die Messpunkte, die von Internetanbietern verwendet werden, befinden sich in der Regel an speziellen Punkten des Ortes. Daher sind Messwerte am Ort der eigenen vier Wände ausgeschlossen. Außerdem ist für Messungen ein In-

ternetzugang notwendig. Außerhalb des eigenen Hausnetzwerks ist dafür der mobile Zugang nötig. Das Radio liefert ähnlich wie das Fernsehen oft nur allgemeine Temperaturen, und zudem ebenfalls nicht in digitaler Form.

Zusammengefasst: Es gibt kaum eine Lösung, die ortsspezifisch ist, alle gewünschten Messdaten liefert und auch noch die Daten optimal für andere Anwendungen bereitstellen kann. Falls dies aber der Wunsch ist, kommt im Prinzip nur eine selbst gebaute Wetterstation in Frage. Denn auch bei Wetterstationen aus dem Handel ist der Kunde auf die vordefinierten Einstellungen und Sensoren des Herstellers beschränkt; Freiraum für eigene Ideen und Projekte ist nicht vorgesehen oder in den meisten Fällen auch nicht gewünscht. Die Wetterstation mit zusätzlichen Sensoren zu erweitern ist daher nicht möglich.

Außerdem befinden sich wenige Produkte am Markt, die zum Beispiel Feinstaub oder den CO<sub>2</sub>-Wert in der Luft messen können. Systeme, die diese Optionen mit anbieten, sind oft unbezahlbar. Bevor Sie also zum Kauf einer Wetterstation schreiten, ist es wichtig, die Anforderungen zu definieren und speziell dafür eine passende Lösung zu finden. Vermutlich werden Sie schnell feststellen, dass die Suche selten von Erfolg gekrönt ist.

## Besser selbst machen

Eine individuelle Lösung wird auf einem Massenmarkt kaum zu finden sein. Wer sie trotzdem will und sie direkt nach dem Kauf einsetzen will, muss oft tief ins Portemonnaie greifen. Zum Glück gibt es kostengünstige Anbieter von Selbstbausystemen, die den Wunsch nach einer eigenen, leistungsfähigen Wetterstation bereits gegen einen geringen Betrag erfüllen. Dabei gilt: Je geringer der Aufwand (zusammenschrauben, löten, programmieren), desto höher der Preis. Sie sollten also eigene Fähigkeiten und Ehrgeiz realistisch einschätzen und bewerten, bevor Sie sich auf einen Anbieter festlegen.

Ein System, das einen einfachen und zugleich günstigen Einstieg gewährleistet und gleichzeitig einen großen Umfang an Sensorik abdeckt, ist das modulare Baukastensystem der Firma Tinkerforge.

Das Unternehmen Tinkerforge ist als Start-up im Jahr 2011 mit dem Ziel angetreten, die Handhabung eingebetteter Systeme zu vereinfachen. Nach nunmehr acht erfolgreichen Jahren bietet Tinkerforge ein einfach bedienbares Baukastensystem mit über 70 verschiedenen Modulen, die ausschließ-

lich in Deutschland produziert werden, und deckt damit über 99 Prozent denkbarer Anwendungsfälle ab. Speziell für Rapid Prototyping oder auch für Kleinserien ist dieses System daher sehr gut geeignet. Individuelle Lösungen entwickelt der Hersteller zusammen mit dem Kunden.

Besonders zu erwähnen ist, dass alle Produkte mit lizenzfreien Open-Source-Programmen entwickelt und vertrieben werden. Für das Leiterkartendesign wird zum Beispiel KiCad eingesetzt, die Entwicklung der Gehäuse wird mit FreeCad realisiert. Selbst das Shop-Backend basiert auf einer Open-Source-Anwendung.



**Die aufgebaute Tischwetterstation** mit transparentem Gehäuse – startklar zum Programmieren (Bild 1)

Ganz dieser Mentalität verpflichtet, stellt Tinkerforge zudem auch alle Dokumente sowie Leiterkartendesigns der Allgemeinheit zur Verfügung. Die Daten und Dokumente stehen bei GitHub zum Download bereit, sodass jeder Interessierte theoretisch auch die Schaltungen der einzelnen Module nachvollziehen und gegebenenfalls auch eigenständig ändern kann. Dies ist aus Sicht des Herstellers sogar erwünscht. Das System soll auf diese Weise dem Kunden den höchstmöglichen Individualisierungsgrad bei komfortabelster Bedienung bieten. Dass dies nicht einfach ist, liegt auf der Hand. Dem Unternehmen aus Schloss Holte-Stukenbrock ist dieser Spagat allerdings sehr gut gelungen.

### Stein auf Stein

Die dotnetpro hat bereits im Jahr 2013 über Tinkerforge berichtet [1]. Inzwischen sind mehr als fünf Jahre vergangen, und da ist es selbstverständlich, dass sich seitdem einige Verbesserungen und Veränderungen angesammelt haben. Welche das sind und wie diese aussehen, erläutern die folgenden Abschnitte.

Das Konzept und die Bezeichnungen der Module haben sich seitdem nicht wesentlich verändert. Die Module sind immer noch in vier Kategorien aufgeteilt.

Die „Bricks“ fungieren weiterhin als Zentrale sowie als Basis für jeden Aufbau. Sie sind etwa vier mal vier Zentimeter groß und lassen sich über Steckverbindungen aufeinander stapeln. Der „Master Brick“ ist immer der unterste „Stein“ eines Stapels und bildet die Grundlage. Er ist für die Kommunikation innerhalb des Aufbaus zuständig und verfügt über insgesamt sieben Anschlussmöglichkeiten: eine Mini USB-Buchse, zwei Stapelstecker und vier Bricklet-Ports.

Über die USB-Buchse lassen sich die Bricks mit dem PC verbinden. An den Bricklet-Ports lassen sich Sensoren, Aktoren oder auch Anzeigen anschließen. Sehr interessant sind die Stapelstecker; sie erlauben es, den Master Brick um weitere Bricks zu erweitern, indem diese einfach übereinandergesteckt werden und sich somit stapeln lassen.

Die restlichen Blöcke, die auf den Master gesteckt werden, sind die Slaves. Ein solcher Brick ist beispielsweise der „Silent Stepper Brick“, der Schrittmotoren steuern kann.

Die Stapelstecker sind zum einen für die Kommunikation innerhalb des Stapels zuständig, zum anderen versorgen sie den Stapel über die USB-Verbindung mit Strom. Sollte die Stromversorgung über USB einmal nicht ausreichen, lässt sich der Master Brick zusätzlich noch mit einer „Step-Down Power Supply“ aufpeppen, an die sich Stromquellen mit bis zu 27 Volt anschließen lassen. Über den Stapelstecker können bis zu 5 Ampere bei 27 Volt weitergeleitet werden.

Bei Motortreibern wie bei dem erwähnten Silent Stepper Brick ist es daher möglich, sie direkt mit einer externen Stromversorgung zu verbinden, sollte die Leistung über die Stapelstecker nicht ausreichen.

Auch die „Bricklets“ gab es 2013 schon, als der erste Artikel erschien [1]. Diese Module werden nicht auf den Stapel gesteckt, sondern mittels eines Bricklet-Kabels an den Brick angeschlossen. Sie dienen dazu, den Aufbau um Funktionen und Eigenschaften erweitern. Es gibt Bricklets für alle erdenklichen Anwendungsfelder. Früher waren sie mit einem EEPROM bestückt, das Code für den jeweiligen Brick enthielt. Die Bricks übernahmen diesen Code in den eigenen Flashspeicher und führten ihn periodisch aus. Dadurch muss ein Brick nicht jedes einzelne Bricklet kennen, und das ermöglicht die große Vielfalt des Baukastensystems.

Jedoch hat sich gezeigt, dass darunter die Performance des Aufbaus leidet. Denn der Brick muss neben den eigenen Aufgaben auch die Funktionalität des Bricklets verarbeiten. Seit 2017 werden daher die Bricklets mit einem Co-Prozessor ausgestattet. Dieser verarbeitet eigenständig die Funktionali- ►

täten des Bricklets und entlastet damit den angeschlossenen Brick. Außerdem übernimmt er auch die Kommunikation mit diesem. Indem das Bricklet nun selbst die Daten verarbeitet, ließ sich die Leistungsfähigkeit einiger Module deutlich steigern. Mit dem alten EEPROM ließen sich Bricklets wie zum Beispiel das „Thermal Imaging Bricklet“ (Wärmebildkamera) nicht umsetzen.

Der Kommunikation gestaltet sich daher nun wie folgt: Der Sensor auf dem Bricklet übermittelt die Datenpakete direkt an den Co-Prozessor, der in unmittelbarer Nähe auf dem Modul sitzt. Die Kommunikation zwischen Brick und Bricklet läuft über das eigene Tinkerforge-Protokoll über SPI mit maximal 1 000 Nachrichten pro Sekunde. Der Brick spricht USB mit dem Brick-Daemon, der sich auf dem PC befindet (wird vom Hersteller kostenlos angeboten), und dieser wiederum verarbeitet die Datenpakete und stellt sie dem Programm via TCP/IP zur Verfügung.

Wenn weitere Schnittstellen benötigt werden, kann der Stapel mit einer sogenannten „Master Extension“ ergänzt werden. Diese Erweiterung kann den Stapel über Ethernet oder auch WLAN mit dem PC verbinden. Auch RS485 wird angeboten.

Vor etwa vier Jahren hat Tinkerforge einen eigenen SBC auf den Markt gebracht. Der „RED Brick“ ist ein vier mal vier Zentimeter großer SBC und damit einer der kleinsten Single-Board-Computer auf dem Markt. Dieser Brick besitzt einen 1-GHz-Prozessor mit 512 MB DDR3-SDRAM sowie einer SD-Karte als Flash-Speicher. Über eine Micro-HDMI-Buchse lässt sich zudem noch ein Display anschließen. Da es sich hierbei um einen Brick handelt, kann dieser auch einfach unter den Stapel gesteckt werden und ermöglicht damit eine vom PC autarke Lösung. Als Alternative zum PC kann auch ein Raspberry Pi in Frage kommen, um mit dem Baukastensystem eine autonome Lösung aufzubauen.

### Software, UID, Bindings, Getter und Co.

Auf der Softwareseite bietet Tinkerforge nach wie vor den schon erwähnten Brick-Daemon, der via USB die Verbindung

zum Brick herstellt und verarbeitet, sowie den Brick Viewer zum Visualisieren und zum Testen der Bricklets. Außerdem lassen sich mit dem Brick Viewer die UIDs der einzelnen Module ermitteln; die UID ist eine einmalig zugeteilte Zahlen-Namen-Kombination, die das Modul zur Identifizierung besitzt. Anhand der UID lässt sich jedes Modul eindeutig erkennen, selbst wenn der verwendete Bricklet-Port für das Modul gewechselt wurde. Dadurch kann beispielsweise das Bricklet den Brick tauschen, ohne dass das Skript angepasst werden muss.

Für das System werden mittlerweile Bibliotheken für über 17 Programmiersprachen angeboten, und somit dürfte wohl für jeden Anwender die von ihm bevorzugte Sprache dabei sein. Auch bei der Wahl einer IDE kann sich der Programmierer frei entscheiden, da Tinkerforge keine Entwicklungsumgebung vorgibt.

Für die Programmierung stellt Tinkerforge entsprechende Bindings bereit. Ein einfacher Methodenaufruf kann auf diese Weise je nach Programmiersprache mit einem Getter und Setter erfolgen; auf diese Weise lassen sich einzelne Datenpakete vom Modul anfragen und zurücksenden. Ein einfacher *GetTemperature()*-Aufruf zum Beispiel ruft die aktuell gemessene Temperatur des Sensors ab und stellt sie für das Programm zur Verfügung.

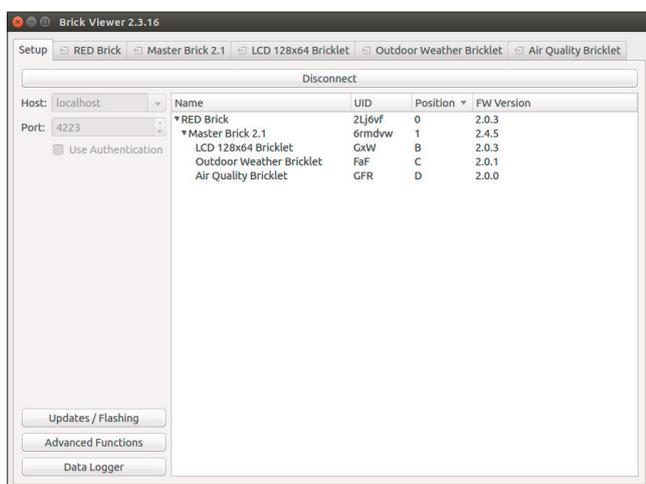
Sobald Daten wiederholt abgerufen werden sollen, gibt es noch die sogenannten Callbacks. Diese können in Abhängigkeit von verschiedenen Parametern konfiguriert werden. Das bedeutet, dass ein solcher Rückruf beispielsweise nur ausgelöst wird, wenn sich die Temperatur verändert hat. Außerdem lässt sich auch eine Zeit einstellen, wie oft und wann frühestens der Abruf erfolgen soll.

Da die Abfragen hinter dem Callback nur in gewissen zeitlichen Abständen ausgeführt werden, lässt sich die Anzahl der Daten, die im System verarbeitet werden, reduzieren und das System dadurch effizienter einsetzen.

Deshalb ist eine Rückruffunktion für eine Wetterstation genau das Richtige. Anstatt das System mit bis zu maximal 1 000 Nachrichten in der Sekunde durch Getter zu überfluten, können die Werte regelmäßig alle 1 000 Millisekunden – also jede Sekunde – abgerufen werden. Damit hat sich die Rate von 1 000 Nachrichten pro Sekunde auf eine Nachricht pro Sekunde reduziert. Für eine meteorologische Messstation ist das mehr als ausreichend.

Hersteller Tinkerforge bietet bereits seit vielen Jahren auch eine quasi fertige Wetterstation an. Diese ist mit einem LCD-Display sowie unterschiedlichen Sensoren ausgestattet. Da seit dem Release dieser Wetterstation geraume Zeit vergangen ist, hat sich das Unternehmen dazu entschlossen, zusätzlich neben der bestehenden Version eine kleinere Tischwetterstation anzubieten.

Diese Variante ist eine komplette Neuentwicklung auf Basis der bis dahin gesammelten Erfahrungen. Die Module stammen weiterhin aus dem modularen Baukastensortiment, jedoch wurde das lasergeschnittene Gehäuse speziell für den Zweck entwickelt. Außerdem reiht es sich optisch in die Designsprache des Herstellers ein und ähnelt stark dem IoT-Gehäuse, das bereits angeboten wird.



Der Brick Viewer listet die verbundenen Bricklets/Module auf und zeigt ihre IDs an (Bild 2)

## Listing 1: Das Programm für die Tischwetterstation

```

1: using System;
2: using Tinkerforge;
3:
4: class ExampleWeatherStation
5: {
6:     private static string HOST = "localhost";
7:     private static int PORT = 4223;
8:     private static string UID_AIR_QUALITY = "GFR";
9:     private static string UID_LCD = "GxW";
10:    private static string UID_OUTDOOR_WEATHER =
        "FaF";
11:    private static BrickletAirQuality air_quality;
12:    private static BrickletLCD128x64 lcd;
13:    private static BrickletOutdoorWeather
        outdoor_weather;
14:
15:    // Callback function for station data callback
16:    static void StationDataCB(
        BrickletOutdoorWeather sender,
        byte identifier, short temperature,
        byte humidity, long windSpeed,
        long gustSpeed, long rain,
        byte windDirection, bool batteryLow)
17:    {
18:        lcd.WriteLine(4, 0, String.Format(
            "Temp OUT: {0,6:##.00} {1}C",
            temperature/10.0, (char)0xF8));
19:    }
20:
21:    // Callback function for all values callback
22:    static void AllValuesCB(BrickletAirQuality
        sender, int iaqIndex,
23:        byte iaqIndexAccuracy, int temperature,
        int humidity, int airPressure)
24:    {
25:        lcd.WriteLine(2, 0, String.Format(
            "IAQ:: {0,6:###}", iaqIndex));
        // 0xF8 == ° on LCD 128x64 charset
27:        lcd.WriteLine(3, 0, String.Format(
            "Temp IN: {0,6:##.00} {1}C",
            temperature/100.0, (char)0xF8));
28:        lcd.WriteLine(5, 0, String.Format(
            "Humidity: {0,6:##.00} %RH",
            humidity/100.0));
29:        lcd.WriteLine(6, 0, String.Format(
            "Air Pres: {0,6:###.0} mbar",
            airPressure/100.0));
30:    }
31:
32:    static void Main()
33:    {
34:        IPConnection ipcon = new IPConnection();
        // Create IP connection
35:        air_quality = new BrickletAirQuality(
            UID_AIR_QUALITY, ipcon);
        // Create device object
36:        outdoor_weather =
            new BrickletOutdoorWeather(
            UID_OUTDOOR_WEATHER, ipcon);
        // Create device object
37:        lcd = new BrickletLCD128x64(
            UID_LCD, ipcon);
38:
39:        ipcon.Connect(HOST, PORT);
40:        // T us ist ipcon must be connected
41:
42:        lcd.ClearDisplay();
43:        lcd.WriteLine(0, 0, "TF Weather Station");
44:
45:        // Register all values callback to function
            AllValuesCB
46:        air_quality.AllValuesCallback +=
            AllValuesCB;
47:        // Set period for all values callback to
            1s (1000ms)
48:        air_quality.
            SetAllValuesCallbackConfiguration(
            1000, false);
49:
50:        // Enable station data callbacks
        outdoor_weather.
51:            SetStationCallbackConfiguration(true);
52:        // Register station data callback to
            function StationDataCB
53:        outdoor_weather.StationDataCallback +=
            StationDataCB;
54:        Console.WriteLine("Press enter to exit");
55:        Console.ReadLine();
56:        ipcon.Disconnect();
57:    }
58: }

```

### Die Tischwetterstation programmieren

Um die Tischwetterstation zu programmieren, ist als Erstes festzulegen, welche Daten benötigt werden und was die Wetterstation ansonsten beherrschen soll. Folgende Eigenschaften und Funktionen werden festgelegt: Die Wetterstation soll

- die Temperatur, die Luftfeuchtigkeit, die Luftqualität und den Luftdruck messen,
- Wetterdaten von den Außenwetterstationen erfassen und lesen,
- alles übersichtlich auf dem Display anzeigen, ▶

- sich kontinuierlich aktualisieren,
- autark laufen.

Dies sind somit die Ausgangssituation und das Ziel. Aufgrund dieser Spezifikationen kann die Tischwetterstation, wie sie bei Tinkerforge angeboten wird, genutzt werden, da sie bereits einen großen Teil der Voraussetzungen erfüllt. Als zusätzliche Module landen im Warenkorb dann noch der RED Brick mit passender SD-Karte (enthält das RED-Brick-Image), das „Outdoor Weather Bricklet“ sowie zum Komplettieren eine Außenwetterstation.

### Verkabeln, schrauben und stecken

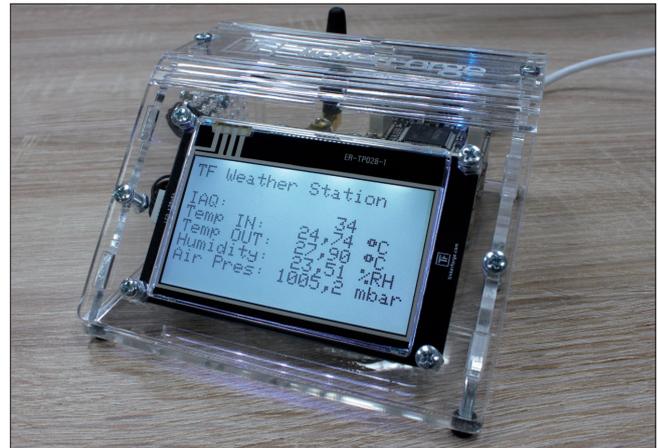
Als Erstes wird das Plexiglasgehäuse von der Folie befreit und im Anschluss das „Air Quality Bricklet“ an eine der Seitenwände geschraubt. Dafür sollten Distanzbolzen verwendet werden, da der Aufbau im Inneren Wärme produziert und damit die Temperaturmessung des Bricklets beeinflussen würde.

Als Nächstes wird der Master Brick auf den RED Brick gesteckt und auf dem Unterboden montiert. Das LCD 128x64 Bricklet wird an die Innenseite des Deckels geschraubt. Da die Tischwetterstation für die Verwendung im Haus ausgelegt ist, wird das Outdoor Weather Bricklet eingebaut und angeschlossen, um die Wetterstation zusätzlich mit Außenwetterdaten füttern zu können. Das Bricklet ist mit einem 433-MHz-Empfänger ausgestattet und damit in der Lage, Daten von bis zu 255 Außenwetterstationen zu empfangen.

Jetzt noch die 7p-10p-Bricklet-Kabel in die Module einstecken und mit dem Master Brick verbinden, dann noch den Deckel aufsetzen und festschrauben. Als letzten Punkt kleben Sie noch die GummifüÙe unter den Boden – fertig. Der Zusammenbau ist im Normalfall in weniger als 15 Minuten erledigt (Bild 1).

### Das eigene Programm schreiben

Als Erstes sind der Brick-Daemon, der Brick Viewer sowie die C#-Bindings herunterzuladen und zu installieren. Im Anschluss daran wird der Aufbau mit einem USB-Kabel mit dem



Alle Werte im Blick – die eigene Wetterstation ist fertig (Bild 4)

PC verbunden und der Brick Viewer gestartet. Nach einem Klick auf *Connect* erscheinen die einzelnen Module im Fenster (Bild 2).

Rechts neben den Namen der Module stehen die UIDs, die Position des Bricks im Stapel oder der Steckplatz des Bricklets an dem jeweiligen Brick und in der letzten Spalte die Version der Firmware. Die UID ist im weiteren Verlauf für die Programmierung sehr wichtig.

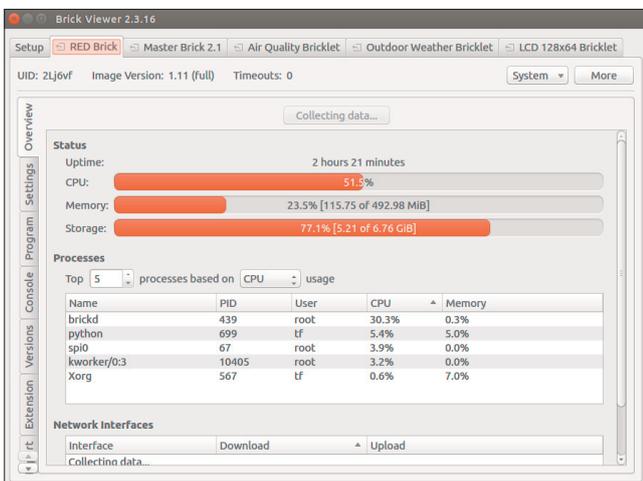
Das fertige Programm für die Tischwetterstation ist in Listing 1 zu sehen. In den ersten Zeilen definiert der Code den Host und Port, ordnet die UIDs zu und importiert die Bibliotheken (Zeilen 6 bis 13). Dank des Brick Viewers konnten die UIDs festgestellt und zur eindeutigen Zuordnung in das Skript übertragen werden. Weiter unten werden die Verbindungen zu den Bricklets (Zeilen 35 bis 37) sowie zum Brick Daemon (Zeile 39) hergestellt; die Variable *air\_quality* beispielsweise wird dabei mit dem Air Quality Bricklet verknüpft und repräsentiert diese Komponente.

Wie vorab definiert, soll das Display alle gemessenen Werte anzeigen und diese zudem automatisch aktualisieren. Für dieses Projekt werden daher die bereits erwähnten Rückrufe verwendet.

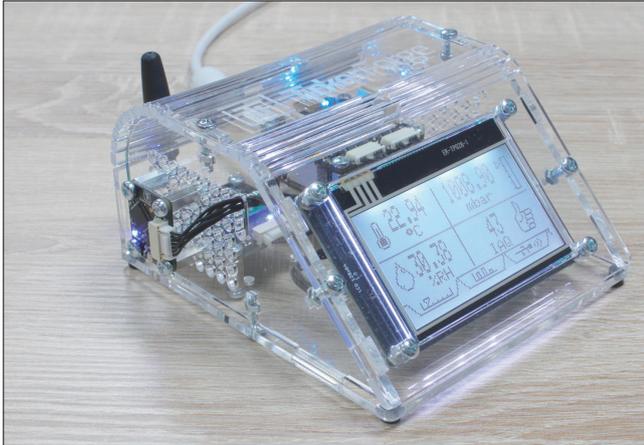
Zunächst wird ein Callback für die Werte des Air Quality Bricklet registriert (Zeile 46). Eine Übersicht über alle verfügbaren Callbacks ist der Dokumentation auf der Internetseite des Herstellers [2] zu entnehmen. Der hier verwendete Rückruf kann alle Daten, die mit dem Air Quality Bricklet gemessen werden können, abrufen und bereitstellen. Mit dem Setter (Zeile 48) werden die Konfigurationen für den *AllValues-Callback* festgelegt. Der Callback wird alle 1000 Millisekunden ausgelöst, unabhängig davon, ob sich der Wert seit der letzten Messung verändert hat.

Die Rückruffunktion für das Outdoor Weather Bricklet (siehe die Zeilen 50 bis 53) ist ähnlich aufgebaut, und wird in Zeile 51 aktiviert.

Innerhalb der Rückrufe werden nun die Messdaten des jeweiligen Bricklets bereitgestellt. Diese werden mit *lcd.WriteLine()* in Zeile 18 auf dem Display dargestellt. Der erste Parameterwert (4) steht für die Zeile und der zweite (0) für die Spalte, in der der Text auf dem Display angezeigt werden soll.



Die Benutzeroberfläche des RED Brick im Brick-Viewer (Bild 3)



Das neue API für die Tinkerforge-Wetterstation bietet eine überarbeitete Oberfläche (Bild 5)

Für die Spalten können Werte von 0 bis 21 und für die Zeilen von 0 bis 7 eingesetzt werden. Die Kombination 0, 0 (Zeile 43) ist die oberste Ausgabezeile und die erste Spalte links.

Zur Wiedergabe der Werte der Außenwetterstation ist sicherzustellen, dass sich mindestens eine funktionsfähige Außenwetterstation im Empfangsumkreis des 433-MHz-Empfängers befindet.

### Testen und speichern

Bevor das Programm auf dem RED Brick gespeichert wird, sollte es zunächst getestet werden. Dafür wird der Aufbau per USB mit dem PC verbunden und das Programm ausgeführt. Sobald alle Fehler behoben sind, wird der RED Brick im Brick-Viewer über den passenden Reiter ausgewählt und im Anschluss auf den Karteireiter *Program* geklickt (Bild 3).

Über *New* wird dann ein Dialogfenster geöffnet, mit dessen Hilfe das neue Programm auf den RED Brick gespeichert werden kann. Bevor das Programm hochgeladen wird, muss als Programmiersprache noch *C#* eingestellt werden. Bei der vorletzten Ansicht sollte noch *Always* ausgewählt werden, damit das Programm im Anschluss durchgängig läuft. Danach wird das Programm auf den RED Brick hochgeladen. Die Software startet automatisch und gibt die Messwerte fein säuberlich untereinander auf dem Display aus, wo sie sich alle 1000 Millisekunden auf den aktuellen Stand bringen, wie es mit der Callback-Konfiguration in Zeile 48 von Listing 1 angegeben ist.

Sollte das Programm nun nicht automatisch ausgeführt werden oder ein Fehler auftreten, dann sollten Sie zuerst den RED Brick aktualisieren. Die Nutzeroberfläche des RED Brick enthält ein Log-File; es kann bei der Fehlersuche sehr hilfreich sein.

### Fix und fertig

Das Projekt „Individuelle Wetterstation“ ist geglückt. Die Selbstbau-Station erfüllt nun alle Wünsche und Erwartungen, die vorab definiert wurden (Bild 4). Für Nutzer, denen die gewählten Eigenschaften noch nicht genügen, bietet Tinkerforge eine große Auswahl unterschiedlicher Module, um die

Wetterstation mit weiteren Sensoren auszustatten. Darunter befindet sich zum Beispiel das „Particulate Matter Bricklet“, das Feinstaub in der Luft misst. Dies kann in Großstädten sehr empfehlenswert sein. Mit einem „Sound Pressure Level Bricklet“ kann die Tischwetterstation zum Beispiel am Arbeitsplatz die Belastung durch Lärm ermitteln. Auch ein „UV Light Bricklet“ oder „Ambient Light Bricklet“ kann eine wertvolle Ergänzung zur bestehenden Tischwetterstation sein. Wenn der RED Brick mit einem WLAN-Stick ausgestattet oder durch eine WIFI Master Extension 2.0 ausgetauscht wird, lässt sich die Wetterstation auch einfach über das WLAN steuern und auswerten.

Seit November 2018 bietet Tinkerforge ein neues API für das LCD 128x64 Bricklet an. Es enthält ein eigenes GUI (Bild 5), das einfach in das Programm eingebunden werden kann. Mit dem API lässt sich eine Schaltfläche erzeugen, die dank der Touch-Funktion des Bricklets betätigt werden kann. Dabei bleibt es dem Entwickler überlassen, wo die Schaltfläche dargestellt wird, wie groß sie ist und welchen Text sie enthalten soll. Besonders spannend ist auch der Graph. Er bietet eine einfache Möglichkeit, die Werte, die vom Air Quality Bricklet oder dem Outdoor Weather Bricklet bereitgestellt werden, in einer Grafik darzustellen. Ähnlich wie bei einer Tabellenkalkulation lassen sich Tabs anzeigen, die zwischen verschiedenen Ansichten wechseln können. Mit wenigen Codezeilen mehr können auf dem LCD auch Grafiken dargestellt werden.

Wenn man alles ausreizt, lässt sich eine Wetterstation aufbauen und programmieren, die mit ihren Funktionen weit über den handelsüblichen Produkten liegt.

### Fazit

Mit der passenden Hardware kann sich jeder Hobby-Meteorologe mit Tinkerforge seine eigene Wetterstation zusammenbauen und dank dieser Anleitung auch noch eigenständig programmieren. Die Messstation ist zudem eine sehr hilfreiche Ergänzung für alle Nutzer, die sich beruflich oder auch privat mit dem Thema Internet of Things auseinandersetzen wollen. Als Einstieg in die Welt der Meteorologie oder als Ergänzung für bestehende Systeme ist diese Wetterstation bestens geeignet. ■

[1] Marcin Kawalerowicz, Bernhard Kron, *Die Bastlerschmiede, Bastelanleitung für eine Continuous-Integration-Ampel*, dotnetpro 12/2013, Seite 70 ff., [www.dotnetpro.de/A1312TinkerForge](http://www.dotnetpro.de/A1312TinkerForge)

[2] *Tinkerforge-Doku*, [www.tinkerforge.com/de/doc](http://www.tinkerforge.com/de/doc)



#### Nicolai Jakob

ist seit Februar 2018 bei Tinkerforge für Einkauf, Vertrieb und Marketing zuständig. Während verschiedener Tätigkeiten in Vertrieb und Marketing studierte er nebenberuflich Betriebswirt mit der Fachrichtung Absatzwirtschaft.

[nicolai@tinkerforge.com](mailto:nicolai@tinkerforge.com)