

Das Taupunkt-Lüftungssystem

Lüften ist die beste und billigste Maßnahme gegen feuchte Keller – jedenfalls, wenn man es richtig macht und sich nicht von außen zusätzliche Nässe ins Gewölbe holt. Bei unserem Projekt behält ein Arduino Nano die aktuellen Taupunkte drinnen und draußen rund um die Uhr im Auge und legt durch gezieltes Lüften den Keller trocken.

von Ulrich Schmerold



Vorher: Dem Schimmel in den feuchten Ecken im Keller ist kaum beizukommen.



Nachher: Die Arduino-gesteuerte Taupunkt-Lüftung legt energiesparend den Keller trocken!

Wer kennt das nicht? Feuchte muffige Kellerräume, an den Wänden ekliger und vor allem oft auch gesundheitsschädlicher Schimmel. Ein Besprühen mit Anti-Schimmelspray löst das Problem nur sehr kurzfristig, aber nicht auf Dauer. Eine professionelle Kellersanierung wirkt zwar, sofern sie gut ausgeführt wird und überhaupt möglich ist, ist aber sehr kostspielig – und kann je nach beauftragter „Fachfirma“ auch ohne Erfolg bleiben. Ebenfalls nicht billig ist auf Dauer der Einsatz elektrischer Luftentfeuchter, die viel Energie verbrauchen.

Dabei geht es auch anders. Das Zauberwort dabei lautet: Lüften!

Dies ist jedoch gar nicht so einfach, wie es zunächst klingt. Die meisten Menschen lüften einfach nach Gefühl oder Zeitplan. Dabei werden aber oft gravierende Fehler gemacht, die die Feuchtigkeit im Haus nur noch weiter ansteigen lassen. Richtiges Lüften ist eine Wissenschaft für sich, die aber glücklicherweise einem Arduino beigebracht werden kann.

Das Prinzip

Der Grundsatz ist noch ganz einfach: Die nasse Luft muss raus, die trockene Luft muss rein. Jetzt könnte man meinen, das ist doch ganz einfach: Wenn die *relative Luftfeuchtigkeit* im Keller höher ist als die draußen, so reißen wir die Fenster auf. Dies führt aber leider nicht immer zum Erfolg, denn die *Lufttemperatur* spielt dabei eine ebenso bedeutende Rolle. Je wärmer die Luft ist, desto mehr Feuchtigkeit kann sie nämlich auch enthalten oder aufnehmen. Warme Sommerluft mit einer empfundenen niedrigen *relativen Luftfeuchte* enthält absolut gesehen dann doch viel Wasser, was sich auf Kellertemperatur heruntergekühlt auf einmal in höherer relativer Luftfeuchte niederschlägt, auch wenn das der Intuition widerspricht.

Viel besser ist es daher, bei der Entscheidung über den Lüftzeitpunkt auf den *Taupunkt* zu schauen, denn der spiegelt den absoluten Wassergehalt einer Luftmasse wider.

Die *Taupunkttemperatur* ist die Temperatur einer Oberfläche, bei der sich bei gegebener Lufttemperatur und relativer Luftfeuchte Tau bildet, also Wasser niederschlägt (wie auf dem Fenster zu sehen ist ❶). Das bedeutet in der Praxis: Wenn etwa die Kellerwand diese Temperatur aufweist, so kondensiert Luftfeuchtigkeit an ihr. Dadurch wird die Wand feucht und es bildet sich ein schimmelfreundliches Milieu.

Zum direkten Messen des Taupunktes wird eine Metalloberfläche langsam heruntergekühlt und dabei beobachtet, bei welcher Temperatur sich Wassertröpfchen bilden. Dies ist dann die Taupunkttemperatur. Das ist das genaueste Verfahren, aber für die Automation des Lüftungsvorgangs mit Maker-Mitteln nicht

Kurzinfo

- » Zusammenhang zwischen Temperatur, relativer Luftfeuchtigkeit und Taupunkt
- » Arduino erkennt günstige Zeitpunkte zum Lüften
- » Idee für rückbaubaren Lüftereinbau im Kellerfenster

Checkliste



Zeitaufwand:
ein bis zwei Wochenenden



Kosten:
ab 100 Euro



Programmieren:
Umgang mit der Arduino-IDE



Elektronik:
Löten von bedrahteten Bauteilen



Handwerk:
ggf. Fensterscheibe auswechseln

Werkzeug

- » LötKolben samt Lötzinn
- » Stichsäge oder anderes Werkzeug, je nach Art der Lüftungsöffnung
- » Bohrmaschine und Bohrer
- » übliches Handwerkzeug

Mehr zum Thema

- » Vladimir Poliakov, Luftentfeuchter mit Raspi steuern, Make 6/21, S. 18

Material

- » Gehäuse z. B. 200mm × 120mm × 55mm
- » Arduino Nano mit aktuellem Bootloader!
- » I²C-LC-Display mit 4 Zeilen
- » Ein-Kanal-Relaismodul 5V
- » 2 Sensoren DHT 22 sowie Kabel für deren Anschluss
- » 2 Widerstände 4,7kΩ
- » 2 Widerstände 10kΩ
- » Elko 1000µF oder größer
- » Kondensator 10nF
- » Lochrasterplatine, für den Testaufbau auch Breadboard und diverse Drahtbrücken
- » Schaltnetzteil 5V
- » Lüfter 220V
- » mehrere Anschlussklemmen
- » 1 oder 2 Acrylglasscheiben je nach Fenster, ggf. Kunststoffprofile und Silikon
- » diverses Befestigungsmaterial etwa Schrauben und Dübel

Alles zum Artikel
im Web unter
make-magazin.de/x2yt

praktikabel. Wir wollen mit Sensoren für Temperatur und relative Luftfeuchtigkeit auskommen.

Taupunkt-Formeln

Glücklicherweise gibt es einige Formeln, die aus diesen beiden Messwerten den Taupunkt näherungsweise berechnen – exakt geht das allerdings nicht, da noch andere variable Größen mitspielen, etwa der aktuelle Luftdruck. Wir verwenden hier eine Taupunktformel ❷, deren Ergebnisse für unsere Zwecke hinreichend genau sind, die mit Messwerten für Temperatur und Luftfeuchte als Input auskommt und die sich gut in einzelne Teile zerlegen lässt, um daraus eine übersichtliche Arduino-Funktion zu erzeugen.

Mit dieser Funktion wird laufend aus einer Temperatur und einer relativen Luftfeuchtigkeit die näherungsweise Taupunkttemperatur berechnet. Dieser Vorgang findet mit zwei Sensoren einmal im Keller und einmal draußen

statt. Die beiden Taupunkttemperaturen werden dann miteinander verglichen. Hat die Luft in unserem Keller den niedrigeren Taupunkt, wird sie mit einem Lüfter nach draußen befördert und Außenluft mit dem niedrigeren Taupunkt strömt nach. Der Effekt: Der Keller wird nach und nach getrocknet.



② Taupunkt näherungsweise berechnen

Die Physik hinter dem Taupunkt ist zu kompliziert, um sie hier kurz zu erklären, aber das Lüftungssystem funktioniert glücklicherweise auch ohne tieferes Verständnis des Nutzers. Zur groben Orientierung: Im Kern besteht die hier verwendete Formel aus der sogenannten *Magnus-Formel*, die aus der Temperatur t (in °C) näherungsweise den *Sättigungsdampfdruck* in Hektopascal (hPa) errechnet. Multipliziert man diesen mit der relativen Luftfeuchtigkeit ϑ (in %), erhält man den *Dampfdruck*. Der wird jetzt sowohl im Nenner als auch im Zähler des großen Bruchs logarithmiert und durch die Konstante 6,1078 geteilt und anschließend einmal mit b multipliziert (im Zähler) und einmal von a subtrahiert (im Nenner), um die *Taupunkttemperatur* t_g (in °C) zu bekommen. Diese Schritte sind im Arduino-Code in der Funktion `float taupunkt(float t, float r)` praktisch eins zu eins umgesetzt.

Bei den beiden Parametern a und b gibt es eine Fallunterscheidung: Für Temperaturen über null Grad Celsius gilt:

$$a = 7,5; b = 237,3$$

$$t_g = \frac{b \cdot \log_{10} \left(\frac{6,1078 \cdot 10^{\frac{(a \cdot t)}{(b+t)}} \cdot \frac{\vartheta}{100\%}}{6,1078} \right)}{a - \log_{10} \left(\frac{6,1078 \cdot 10^{\frac{(a \cdot t)}{(b+t)}} \cdot \frac{\vartheta}{100\%}}{6,1078} \right)}$$

Sättigungsdampfdruck (Magnus-Formel) Dampfdruck

Bei Temperaturen unter dem Gefrierpunkt rechnet man stattdessen mit:

$$a = 7,6; b = 240,7$$

Etwas mehr Hintergrund zur Formel findet man online über den Link in der Kurzinfo.

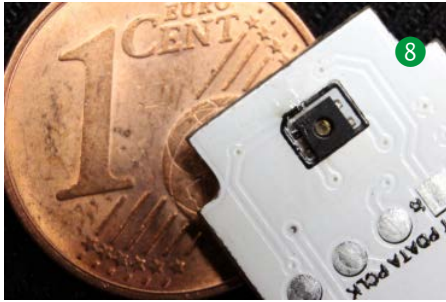
Alternativ lässt sich der Taupunkt mit hinreichender Genauigkeit auch aus einer Tabelle ③ herauslesen. Wer also keine Automatisierung des Lüftungsvorgangs umsetzen kann oder möchte, kann den richtigen Zeitpunkt für das Lüften auch wie folgt herausfinden: Mit Thermometer und Hygrometer Temperatur und relative Luftfeuchte innen und außen

messen und jeweils mit der Tabelle den Taupunkt ermitteln. Dann vom inneren Taupunkt den äußeren Taupunkt abziehen. Ist das Ergebnis positiv, kann gelüftet werden; je positiver, desto besser. Für die Tabelle wurde übrigens nicht die Formel ② verwendet, ich habe sie vor Jahren Wert für Wert mit Hilfe eines Online-Rechners erstellt.

An der Tabelle sieht man auch leicht, dass man sich beim Lüften nur schlecht auf seine Intuition verlassen kann. Angenommen, drinnen misst man bei 20°C eine relative Luftfeuchte von 60%, dann ergibt sich eine Taupunkttemperatur von 12°C. Sind es draußen 15°C mit 75% Luftfeuchtigkeit, ergibt sich als Taupunkttemperatur 10,6°C – wer jetzt lüftet, bringt

③ Taupunkttafel

| Temperatur | relative Luftfeuchte | | | | | | | | | | | | | |
|------------|----------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 30% | 35% | 40% | 45% | 50% | 55% | 60% | 65% | 70% | 75% | 80% | 85% | 90% | 95% |
| 30°C | 10,5 | 12,9 | 14,9 | 16,8 | 18,4 | 20 | 21,4 | 22,7 | 23,9 | 25,1 | 26,2 | 27,2 | 28,2 | 29,1 |
| 29°C | 9,7 | 12 | 14 | 15,9 | 17,5 | 19 | 20,4 | 21,7 | 23 | 24,1 | 25,2 | 26,2 | 27,2 | 28,1 |
| 28°C | 8,8 | 11,1 | 13,1 | 15 | 16,6 | 18,1 | 19,5 | 20,8 | 22 | 23,2 | 24,2 | 25,2 | 26,2 | 27,1 |
| 27°C | 8 | 10,2 | 12,2 | 14,1 | 15,7 | 17,2 | 18,6 | 19,9 | 21,1 | 22,2 | 23,3 | 24,3 | 25,2 | 26,1 |
| 26°C | 7,1 | 9,4 | 11,4 | 13,2 | 14,8 | 16,3 | 17,6 | 18,9 | 20,1 | 21,2 | 22,3 | 23,3 | 24,2 | 25,1 |
| 25°C | 6,2 | 8,5 | 10,5 | 12,2 | 13,9 | 15,3 | 16,7 | 18 | 19,1 | 20,3 | 21,3 | 22,3 | 23,2 | 24,1 |
| 24°C | 5,4 | 7,6 | 9,6 | 11,3 | 12,9 | 14,4 | 15,8 | 17 | 18,2 | 19,3 | 20,3 | 21,3 | 22,3 | 23,1 |
| 23°C | 4,5 | 6,7 | 8,7 | 10,4 | 12 | 13,5 | 14,8 | 16,1 | 17,2 | 18,3 | 19,4 | 20,3 | 21,3 | 22,2 |
| 22°C | 3,6 | 5,9 | 7,8 | 9,5 | 11,1 | 12,5 | 13,9 | 15,1 | 16,3 | 17,4 | 18,4 | 19,4 | 20,3 | 21,2 |
| 21°C | 2,8 | 5 | 6,9 | 8,6 | 10,2 | 11,6 | 12,9 | 14,2 | 15,3 | 16,4 | 17,4 | 18,4 | 19,3 | 20,2 |
| 20°C | 1,9 | 4,1 | 6 | 7,7 | 9,3 | 10,7 | 12 | 13,2 | 14,4 | 15,4 | 16,4 | 17,4 | 18,3 | 19,2 |
| 19°C | 1 | 3,2 | 5,1 | 6,8 | 8,3 | 9,8 | 11,1 | 12,3 | 13,4 | 14,5 | 15,5 | 16,4 | 17,3 | 18,2 |
| 18°C | 0,2 | 2,3 | 4,2 | 5,9 | 7,4 | 8,8 | 10,1 | 11,3 | 12,5 | 13,5 | 14,5 | 15,5 | 16,3 | 17,2 |
| 17°C | -0,6 | 1,4 | 3,3 | 5 | 6,5 | 7,9 | 9,2 | 10,4 | 11,5 | 12,5 | 13,5 | 14,5 | 15,3 | 16,2 |
| 16°C | -1,4 | -0,5 | 2,4 | 4,1 | 5,6 | 7 | 8,2 | 9,4 | 10,5 | 11,6 | 12,6 | 13,5 | 14,4 | 15,2 |
| 15°C | -2,2 | -0,3 | 1,5 | 3,2 | 4,7 | 6,1 | 7,3 | 8,5 | 9,6 | 10,6 | 11,6 | 12,5 | 13,4 | 14,2 |
| 14°C | -2,9 | -1 | 0,6 | 2,3 | 3,7 | 5,1 | 6,4 | 7,5 | 8,6 | 9,6 | 10,6 | 11,5 | 12,4 | 13,2 |
| 13°C | -3,7 | -1,9 | 0,1 | 1,3 | 2,8 | 4,2 | 5,5 | 6,6 | 7,7 | 8,7 | 9,6 | 10,5 | 11,4 | 12,2 |
| 12°C | -4,5 | -2,6 | 1 | 0,4 | 1,9 | 3,2 | 4,5 | 5,7 | 6,7 | 7,7 | 8,7 | 9,6 | 10,4 | 11,2 |
| 11°C | -5,2 | -3,4 | 1,8 | -0,4 | 1 | 2,3 | 3,5 | 4,7 | 5,8 | 6,7 | 7,7 | 8,6 | 9,4 | 10,2 |
| 10°C | -6 | -4,2 | 2,6 | -1,2 | 0,1 | 1,4 | 2,6 | 3,7 | 4,8 | 5,8 | 6,7 | 7,6 | 8,4 | 9,2 |



trockenere Luft nach drinnen. Im Sommer hingegen, etwa bei 25°C und 50% Luftfeuchtigkeit im Freien, liegt der Taupunkt dort tatsächlich bei 13,9°C. Wenn man jetzt lüftet, weil sich die Luft so schön warm anfühlt, holt man sich tatsächlich zusätzliche Feuchtigkeit in den Keller ...

Die Sensoren

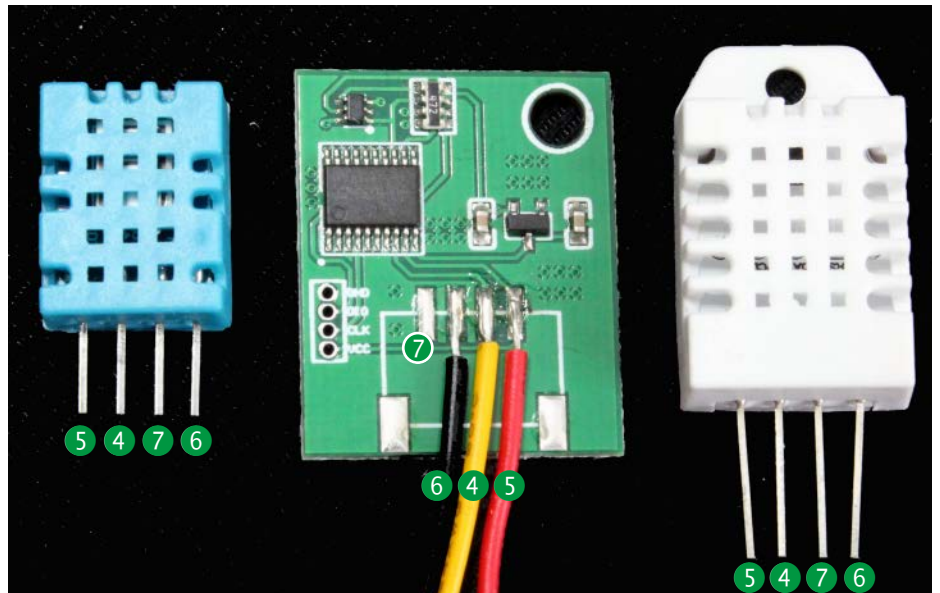
Zum Messen von Temperatur und Luftfeuchtigkeit gibt es viele Kombisensoren auf dem Markt. Nach dem Testen mehrerer verschiedener Sensoren fiel die Wahl auf den Sensor DHT22. Er arbeitet sehr stabil, die Messwerte sind hinreichend exakt, er ist günstig in der Anschaffung und es gibt eine Arduino-Bibliothek für ihn.

Die DHT-Sensoren verschicken ihre Daten seriell über einen *One-Wire-Bus*, der – wie der Name schon sagt – mit einer Datenleitung auskommt ④. Weiter benötigt der Sensor eine Versorgungsspannung (VCC) von 3,3V – 6V ⑤ und einen Masseanschluss (GND ⑥). Der verbleibende vierte Anschluss ⑦ ist ungenutzt.

Wie klein der eigentliche Temperatur- und Feuchtigkeitssensor ist, lässt sich an einem zerlegten DHT22-Modell (gut im Vergleich zu einem Cent erkennen ⑧).

Als Außensensor hatten wir anfangs einen DHT21 im passenden Außengehäuse erworben. Leider gab der Sensor jedoch schon nach kurzer Zeit den Geist auf. Aus diesem Grund montierten wir ganz im Sinne des *Upcyclings* in das funktionale Gehäuse des DHT21 einen DHT22-Sensor, der nun störungsfrei arbeitet ⑨. Für den Fall, dass wir den Sensor erneut austauschen müssen, erfolgte die Montage gleich mit einer 4-poligen Schraubklemme. Bisher war dies aber nicht nötig.

Die maximale Kabellänge für den DHT22 ist aus dem Datenblatt nicht herauszulesen – die hängt nicht nur vom Sensor, sondern auch vom Board ab. Nach der Auswertung vieler verschiedener Quellen gehen wir aber davon aus, dass 10 Meter problemlos möglich sind. Allerdings sollte dann das Kabel etwas hochwertiger sein (z. B. *Twisted-Pair*-Kabel mit Schirm oder CAT6-Kabel) und die beiden Pull-up-Widerstände der Sensoren sollten verkleinert werden (1,2k bis 4,7k). Andere berichten jedoch auch, dass ein 50-Meter-Kabel mit

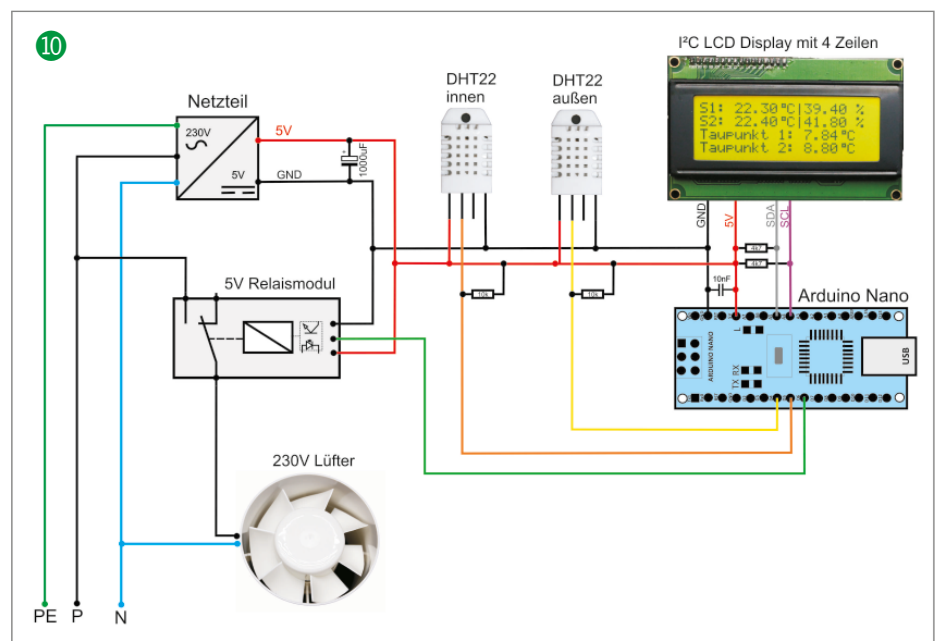


Verschiedene DHT-Sensoren von links nach rechts: DHT11, DHT21, DHT22.

einem DHT22 funktioniert. Getestet haben wir das allerdings nicht.

Die Schaltung

Wie sich aus dem Schaltplan ⑩ gut erkennen lässt, hält sich der Schaltungsaufwand bei diesem Projekt in Grenzen. Die beiden 4,7k-Widerstände sind als Pull-up-Widerstände für das LCD-Display verbaut. Die beiden 10k-Widerstände sind Pull-ups für die Datenleitungen der DHT-Sensoren. Bei unseren Tests funktionierten die Sensoren zwar auch ohne diese, vollständigshalber haben wir sie jedoch – wie im Datenblatt empfohlen – eingebaut. Der Arduino hat auch interne Pull-ups, das macht sich im Test bemerkbar. Bei längeren





Leitungen und auch Störungen sind externe Widerstände aber robuster.

Die beiden Kondensatoren (1000µf und 10nf) dienen zur Spannungsstabilisation und zum Filtern von EMV-Störungen. Während des

Testbetriebs zeigten sich nämlich unregelmäßig Systemabstürze, jedenfalls solange der Lüfter angeschlossen war – ohne ihn blieben die Abstürze aus. Es sieht somit danach aus, dass der Lüfter unregelmäßig Störungen zum Arduino sendet, obwohl durch das Relaismodul eine galvanische Trennung besteht.

Um sicherzugehen, zerlegten wir den Lüfter und untersuchten ihn ¹¹. Entstörglieder enthielt dieser ebenso wenig wie einen Anschluss für ein Schutzkontaktkabel (aber klar, das Gehäuse ist ja auch aus Kunststoff). Neben den Kondensatoren fängt am Ende auch die Software Störungen ab, dazu gleich mehr.

Zum schnellen Testen lässt sich die Schaltung problemlos auf einem Breadboard aufbauen ¹². Hier können dann auch noch Korrekturwerte für die Sensoren ermittelt und

in die Software eingetragen werden. Dazu habe ich beim Testaufbau die angezeigten Messwerte der Sensoren mit einem genauen Thermometer und Hygrometer verglichen. Wichtiger als möglichst korrekte absolute Sensormesswerte ist allerdings, dass beide Sensoren in der gleichen Umgebung auch identische Werte anzeigen, da das Lüftungssystem später nur auf das Verhältnis zwischen drinnen und draußen schaut.

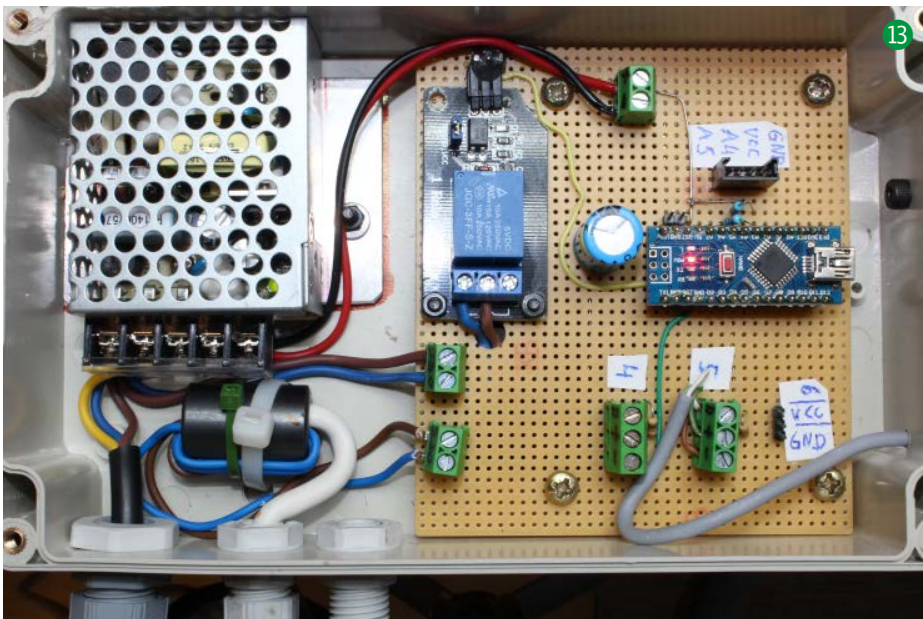
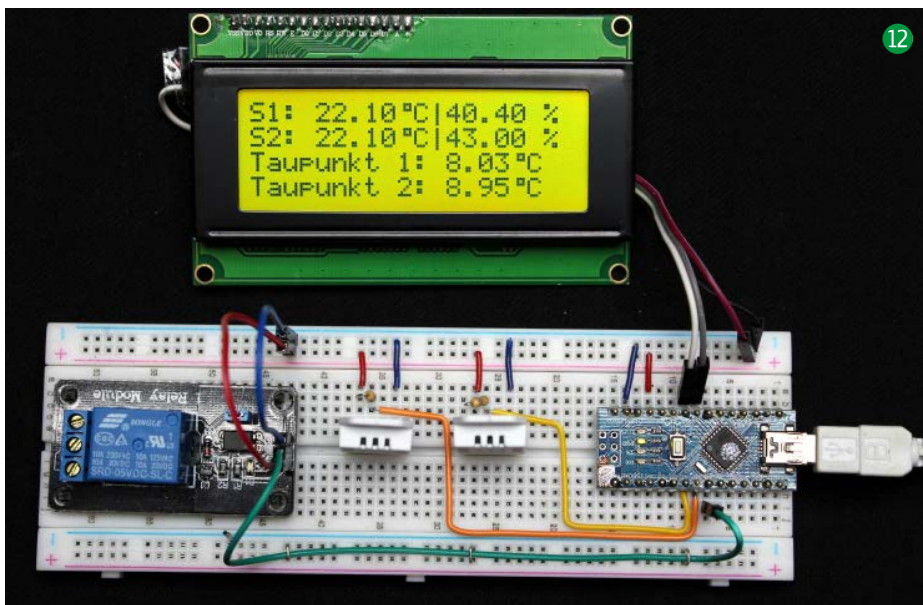
In der Endversion finden alle Komponenten in einem Gehäuse mit den Maßen 200mm x 120mm x 57mm Platz ¹³. Auf der Lochrasterplatine lassen sich hauptsächlich viele Anschlussklemmen erkennen (für 230V Netzanschluss, 230V Lüfteranschluss, je eine Leiste für die Sensoren und eine Steckerleiste für das LCD-Display). Ganz links befindet sich das 5V-Schaltnetzteil. Darunter ist ein Ferritkern zu erkennen, um den wir die Leitungen des Lüfters zur weiteren EMV-Entstörung geführt haben.

Links oberhalb des Arduino ist eine 2-polige Steckerleiste für einen (optionalen) Reset-Taster zu erkennen (nicht im Schaltplan eingezeichnet). Rechts unten befindet sich eine 3-polige Steckerleiste für einen (ebenfalls optionalen) Schalter mit drei Stellungen: *Ein*, *Auto* und *Aus*. Die Beschaltung kann man auf dem Bild ablesen. Die Außenseite des Gehäuses zeigt Bild ¹⁴.

Die Software

Im Programm werden folgende Bibliotheken verwendet: *DHT sensor library* von Adafruit, *LiquidCrystal I2C* von Marco Schwartz und *Sodaq_wdt*. Falls nicht schon für andere Projekte geschehen, müssen diese Bibliotheken in der Arduino IDE unter *Werkzeuge/Bibliotheken verwalten* hinzugefügt werden; wie das geht, beschreibt ein Online-Artikel (siehe Link in der Kurzinfor).

Zu Beginn des Quelltextes werden viele Konstanten deklariert ¹⁵. Hier werden z. B. die Ports eingestellt, an denen sich die Sensoren befinden und welche Sensortypen verbaut wurden. Dann werden die Korrekturwerte der Sensoren festgelegt, falls diese nicht von Haus aus exakte Werte ausgeben. Darauf folgen Parameter zum Schaltverhalten der Steuerung, z.B. ab welchem Taupunktunterschied



der Lüfter angesteuert wird oder wie groß der Abstand von Ein- und Ausschaltpunkt sein soll, damit der Lüfter nicht dauernd an- und gleich wieder abgeschaltet wird.

Im eigentlichen Programm fällt nun in Zeile 40 ein Aufruf der Watchdog-Funktion auf:

```
wdt_enable(WDTO_8S);
```

Diesen Watchdog haben wir eingeführt, da sich auch mit erhöhten EMV-Maßnahmen für den Lüfter die Systemhänger nicht komplett auf null reduzieren ließen – so alle ein bis vier Tage stürzt die Steuerung ab, wenn der Lüfter dranhängt (ohne Lüfter allerdings überhaupt nicht, am Code liegt es also nicht). Die Watchdog-Funktion aus der Bibliothek *avr/wdt.h* startet deshalb den Arduino neu, falls nicht spätestens nach 8 Sekunden ein Reset-Signal gesendet wurde, was in der Zeile 161 passiert:

```
wdt_reset();
```

Achtung! Für die Watchdog-Funktion benötigt der Arduino **zwingend** die neue Bootloader-Version *optiboot*. In die ältere Version (*ATmegaBOOT*) hatte sich ein Fehler eingeschlichen, der beim Auslösen des Watchdogs den Arduino in eine Endlosschleife schickte. Es folgten endlose Starts und Neustarts, die sich nur schwer in den Griff kriegen ließen. Bei den China-Klonen des Arduino Nano haben wir bisher nur den alten Bootloader gefunden. Den neuen Bootloader bringen meist nur originale Arduino Nanos mit, die mit rund 17 Euro aber mehr kosten als die Klone. Dennoch können Sie auch so ein Board für dieses Projekt einsetzen; Sie müssen nur den neuen Bootloader aufspielen. Dafür finden Sie eine Anleitung bei den Links zum Artikel (URL in der Kurzfinfo). Dort lesen Sie auch, wie Sie im Zweifelsfall vorab herausbekommen, ob ein Board bereits den neuen Bootloader enthält.

Der Quelltext der Arduino-Steuerung ist bisher absichtlich einfach gehalten. Jedoch kann es durchaus sinnvoll sein, etwa eine Funktion für Datenlogging hinzuzufügen. Die Temperaturen, Luftfeuchtigkeiten und Taupunkte könnten dann auf einer SD-Speicherkarte gespeichert werden. Auch die Laufzeiten des Lüfters wären sicher bei einer Auswertung interessant.

15 Konstanten im Code

```
#define RELAIPIIN 6 // Anschluss des Lüfter-Relais
#define DHTPIN_1 5 // Datenleitung für den DHT-Sensor 1 (innen)
#define DHTPIN_2 4 // Datenleitung für den DHT-Sensor 2 (außen)

#define RELAIS_EIN LOW
#define RELAIS_AUS HIGH
bool rel;

#define DHTTYPE_1 DHT22 // DHT 22
#define DHTTYPE_2 DHT22 // DHT 22

// ***** Korrekturwerte der einzelnen Sensorwerte *****
#define Korrektur_t_1 -3 // Korrekturwert Innensensor Temperatur
#define Korrektur_t_2 -4 // Korrekturwert Außensensor Temperatur
#define Korrektur_h_1 0 // Korrekturwert Innensensor Luftfeuchtigkeit
#define Korrektur_h_2 0 // Korrekturwert Außensensor Luftfeuchtigkeit
//*****

#define SCHALTmin 5.0 // minimaler Taupunktunterschied, bei dem das
// Relais schaltet
#define HYSTERESE 1.0 // Abstand von Ein- und Ausschaltpunkt
#define TEMP1_min 10.0 // Minimale Innentemperatur, bei der die Lüftung
// aktiviert wird
#define TEMP2_min -10.0 // Minimale Außentemperatur, bei der die Lüftung
// aktiviert wird

DHT dht1(DHTPIN_1, DHTTYPE_1); //Der Innensensor wird ab jetzt mit dht1
// angesprochen
DHT dht2(DHTPIN_2, DHTTYPE_2); //Der Außensensor wird ab jetzt mit dht2
// angesprochen
```

Der Ausweg

Nun bleibt noch das Problem: Wie bekommen wir die Luft nach draußen? Einen Mauerdurchbruch traut sich nicht jeder zu und in einer Mietwohnung kann das auch richtig Ärger geben. Da auch wir nicht unsere Vollwärmeschutzfassade beschädigen wollten, haben wir einen anderen Ausweg gesucht – und gefunden.

Warum nicht in eine Glasscheibe des Kellerfensters ein Loch schneiden? Da aber auch dies nicht reversibel ist, haben wir kurzerhand die originale Scheibe gegen eine selbst angefertigte Acrylglasscheibe ausgetauscht. Solche Scheiben kann man etwa im Internet kaufen und oft auch gleich in den passenden Maßen zuschneiden lassen. In das Acrylglas lässt sich nun problemlos mit der Stichsäge eine Öffnung für den Lüfter sägen **16**.

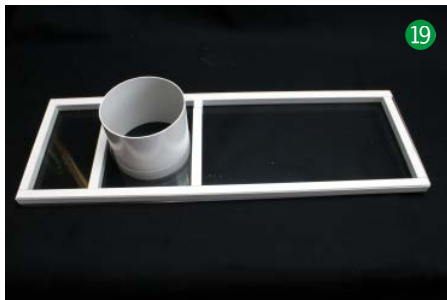
Wird statt einer Einfachverglasung eine Isolierglasscheibe ersetzt, verwendet man ein-

fach zwei Acrylglasscheiben **17**, die mit Kunststoffprofilen auf den passenden Abstand gebracht werden. Diese Konstruktion wird mit transparentem Silikon zusammengeklebt **18**. Beim Zusammenkleben sollte allerdings peinlich auf Sauberkeit geachtet werden – sind die Scheiben erst einmal verklebt, können Verunreinigungen oder Kunststoffspäne aus dem Zwischenraum nicht mehr entfernt werden ...

Zur Aufnahme des Rohrlüfters haben wir ein Lüftungsrohr passend abgesägt, durch die Öffnungen in den Acrylglasscheiben geführt und mit Silikon eingeklebt **19**. Vorsicht: Platzieren Sie das Rohr nicht zu nahe an der Außenkante! Anderenfalls lässt sich später das Fenster nicht mehr vollständig öffnen, da das Rohr stört.

Bild **20** zeigt eine fertige „Isolierglas“-Scheibe mit Lüfterdurchbruch und Lüftungsgitter. Die Isolationswirkung ist übrigens nicht einmal schlecht: Bei Messungen mit dem





Wichtig: Schornsteinfeger fragen

Vor Einbau eines elektrischen Lüfters/Entlüfters sollte man unbedingt den Schornsteinfeger fragen, ob das im konkreten Fall zulässig ist. Falls etwa eine Gasheizung im Keller ist, kann der Lüfter deren Abgasstrom stören, insbesondere, falls er

die Luft aktiv nach draußen befördert und die Außenluft passiv einströmt. Dann erhält man nämlich einen Unterdruck im Keller und die Luft kann auch durch den Schornstein der Gasheizung (inklusive deren Abgase) einströmen ...



Infrarotthermometer war die Acrylglasscheibe nur um durchschnittlich 1,4°C kälter als die echte Isolierglasscheibe.

Wer den Eigenbau scheut: Klimaanlagebauer haben zum Teil unter dem Stichwort „Abluftscheibe“ ähnliche Konstruktionen halbfertig im Angebot oder fertigen sie auf Maß. Das kann dann aber schnell mehr kosten als der Rest des Systems ...

Die Art und Weise, wie Ihre bisherigen Scheiben entfernt werden, hängt natürlich davon ab, welche Art von Fenstern Sie besitzen. Einen Weg sie auszubauen gibt es auf jeden Fall – irgendwie muss die Scheibe ja auch mal hineingekommen sein. Hier zeigen wir, wie die verbreiteten Fenster mit Glashalteleisten zerlegt werden. Zum Auswechseln der Scheibe werden alle vier Profileisten auf der Innenseite mit einem Spachtel vorsichtig herausgehoben (21) (22) (bei Holzfenstern sind sie oft mit Nägeln fixiert).

Anschließend kann schon die Originalscheibe entfernt und die Acrylscheibe eingesetzt werden. Nun noch die vorher entfernten Profile wieder einsetzen und mit einem Kunststoffhammer vorsichtig festklopfen (23). Fertig!

Die fertige Steuerung haben wir direkt neben einem Kellerfenster montiert (24). Der Innensensor befindet sich rechts am Gehäuse. Der Außensensor hängt draußen regengeschützt unter der Abschlussleiste des Vollwärmeschutzes (25). Der Rohrlüfter wurde so montiert, dass er die Luft von innen nach außen fördert.

Besitzt man ein recht dichtes Haus oder Keller, kann es nötig sein, an der gegenüberliegenden Kellerseite einen Luftzugang zu installieren. Der Lüfter kann natürlich nur richtig arbeiten, wenn irgendwo Luft nachströmt. Im Zweifelsfall dafür wieder eine Acrylglasscheibe mit Öffnung anfertigen – dieses Mal dann ohne Lüfter, nur mit Lüftungsgitter.

Die Erfahrungen

In den ersten Tagen nach Inbetriebnahme der Taupunkt-Lüftungssteuerung lief unser Lüfter pausenlos. Sobald die Feuchtigkeit im Raum abgenommen hatte, gingen die Laufzeiten nach und nach zurück. Die schon bestehenden Schimmelflecken müssen selbstverständlich noch einmal konventionell bekämpft werden, z. B. mit einem chlorhaltigen Schimmelspray und warmen Spülwasser und einer Bürste. Durch die sinkende Luftfeuchtigkeit und die abnehmende Feuchtigkeit im Mauerwerk sollten in Zukunft jedoch keine Flecken mehr auftreten.

In meinem Siedlerhaus von 1948 war schon nach wenigen Tagen eine Verringerung der Luftfeuchtigkeit deutlich messbar. Der Trocknungseffekt beschränkt sich aber nicht nur auf den Keller. Ich merke ihn auch noch deutlich im Erdgeschoss und im ersten Stock. In allen Räumen sank die relative Luftfeuchtigkeit bis in einen gesunden Bereich (40% – 60%) ab. Der Erfolg der Taupunkt-Lüftungssteuerung ist also durchdringend.

Wenn Sie die Lüftungssteuerung nachbauen: Bitte seien Sie nicht frustriert, wenn die Feuchtigkeit im Raum auch mal ansteigt. Wird der Lüfter aufgrund der Datenlage nicht eingeschaltet (etwa, weil die Luft draußen feuchter ist als drinnen), so diffundiert dennoch Feuchtigkeit aus den Wänden in den Raum und die relative Luftfeuchtigkeit dort steigt wieder an. Sobald der äußere Taupunkt erneut tief genug sinkt, wird der Lüfter seiner Aufgabe aber wieder nachkommen. Und Sie müssen sich keine Gedanken mehr um den richtigen Zeitpunkt zum Lüften machen. —pek

