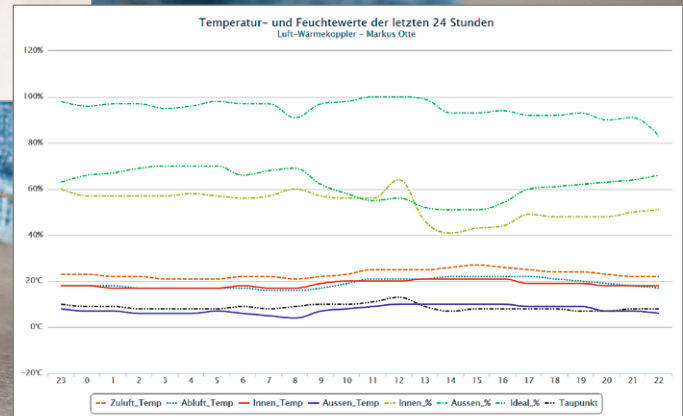
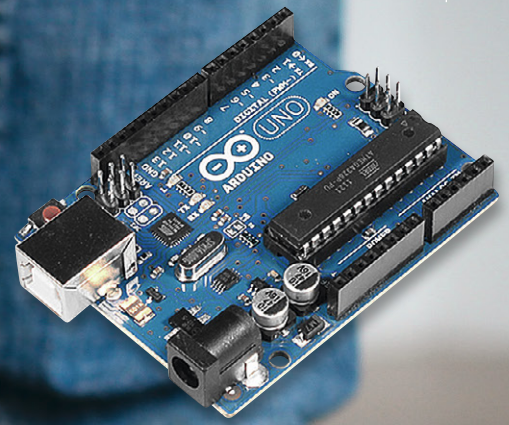




Perfektes Klima mit dem Arduino

Der regelmäßige Luftaustausch im Gebäude ist eine wichtige Klimatisierungsmaßnahme, zumal in gut isolierten Gebäuden. Um dabei nicht allzuviel Heizungsenergie einzubüßen, setzt man Wärmetauscher ein. Diese nutzen die Wärmeenergie der Abluft, um die frische Zuluft vorzuwärmen. Unser Leser Markus Otte hat hierzu eine ausgeklügelte Lösung erarbeitet, die mit einem Arduino gesteuert wird und neben der üblichen reinen Temperaturregelung das gesamte Raumklima, also auch die Luftfeuchte, berücksichtigt.

Vielen Dank!



Markus Otte aus Beverungen

hat für seinen Beitrag zum Leserwettbewerb einen 200-Euro-Warengutschein erhalten!

Gutes Raumklima

Der Winter bringt kühle trockene Luft, der Sommer ist meist warm und feucht. Die Crux an der Sache ist, dass wir den Wohnbereich am liebsten immer gleichbleibend warm bei idealer Luftfeuchte haben möchten.

Dazu kommt dann noch die Luftqualität, welche sich rapide verschlechtert, wenn sich viele Personen über längere Zeit im Raum befinden.

Leider wird häufig zu wenig oder falsch gelüftet, so öffnet man im Sommer vielfach ganztags die Fenster und im Winter bleiben diese faktisch dauernd geschlossen. Um nicht die Wärme aus dem Innenraum durch falsche Lüftung zu verlieren, bietet es sich an, dieses durch Temperatenausgleich der Zu- und Abluft zu bewerkstelligen – eine wichtige Maßnahme gerade auch bei gut isolierten Gebäuden, um nicht nur eine angenehme Klimatisierung zu erreichen, sondern auch Bauschäden durch Schimmelbildung vorzubeugen.

Nun gibt es viele Anbieter von Low-Budget-Luft-Wärme-Tauschern für den Heimbereich im Internet.

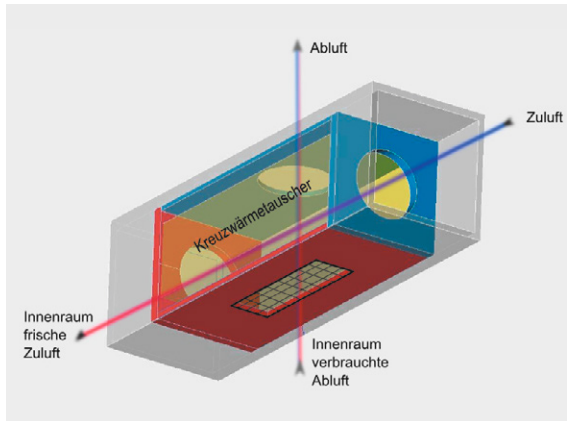


Bild 1: Das Arbeitsprinzip eines Kreuzwärmetauschers

Einige setzen auf Kreuzwärmetauscher (Bild 1), welche über 12-V-PC-Lüfter eine Umwälzung ermöglichen, andere auf Keramikeinsätze in der Wand, die Wärme beim Luftaustritt speichern und beim Ansaugen wieder abgeben sollen. Alle diese Geräte haben das grundlegende Problem, dass die Luftfeuchte nicht berücksichtigt wird. Dabei ist diese eine sehr wichtige Komponente des Raumklimas, schließlich ist die richtige Balance zwischen Raumtemperatur und Raumluftfeuchte der Schlüssel zum angenehmen, gesunden und der Bausubstanz nicht schadenenden Klima. Zur Steuerung des Wärmetauschers über die beiden Klimakomponenten bietet sich heute natürlich ein Mikrocontroller an.

Nun habe ich mir einen Arduino zum Experimentieren gekauft, genaugenommen ein Starterpaket mit einem Arduino Duemilanove und 16 KB Flash-Memory. Leider erwies sich das im Lauf des Projekts als zu schmalbrüstig und der Umstieg auf einen Arduino UNO R3 mit 32 KB Flash-Speicher verbesserte die Möglichkeit, ein umfangreicheres Programm zu schreiben.

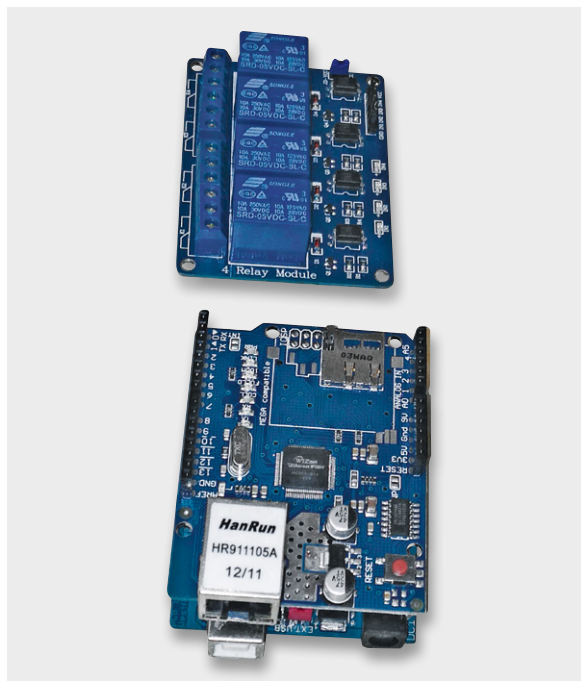


Bild 2: Neben einem Arduino-Board kommen ein Ethernet-Shield und eine Relaiskarte zum Einsatz.

Dazu kam dann noch ein passendes Ethernet-Shield. Bei beiden Bauteilen empfiehlt sich der Kauf des jeweiligen Originals, da auch meine Erfahrung war, dass die günstigen Nachbauten nicht unbedingt für den Dauerbetrieb geeignet sind. Als weiteres notwendiges Bauteil kam ein 4-Kanal-Relais-Modul dazu, welches passend für die Spannungen am Arduino ausgelegt ist. Hiermit lassen sich die beiden 230-V-Lüfter des Wärmetauschers direkt betreiben. Bild 2 zeigt die beiden Module.

Während der Internetrecherche stieß ich auf eine Website [1], welche viele Arduino-Skripte und kleine Schaltungen im Temperatur-/Feuchtebereich anbietet. Dort ist auch die Datenübertragung zu einem Webserver beschrieben. Um die gemessenen Daten auf einen Webserver schreiben zu können, muss auf jeden Fall Zugriff auf eine MySQL-Datenbank möglich sein. Am einfachsten ist dies über Webspace mit MySQL-Support oder der Betrieb eines eigenen Servers zu Hause (z. B. mit Hilfe eines Raspberry Pi oder NAS). Die Schwierigkeit besteht hier allerdings, dass man über einen DynDNS-Server im Internet eine Umleitung auf die eigene dynamische externe IP-Adresse herstellen muss.

Die Lüfter

Zunächst einmal stellt sich die Frage: Wie schnell tausche ich die Luft aus bzw. wie laut darf das Ganze im Wohnbereich sein, um nicht störend zu wirken?

Lufttauscher einiger Hersteller verwenden häufig 12-V-Lüfter, die sich zum einen elektronisch einfach ansteuern bzw. regeln lassen und zum anderen in der Anschaffung recht günstig sind. Nachteil ist die geringe Strömungsleistung, was den Luftaustausch und damit die Geräuschbelästigungsdauer deutlich verlängert. Ich habe mich deshalb für die 230-V-Variante entschieden, die in meinem Fall nicht geregelt wird, was aber theoretisch mit etwas technischem Aufwand möglich wäre. Nach langer Internetrecherche fand ich eine besonders leise Variante, welche zudem noch eine lange Herstellergarantie bietet.

Idealerweise würde man nun die Zu-/Ablüfter außerhalb des Gebäudes installieren, was der Senkung der Lärmbelastung im Innenbereich zugute kommt. Leider fand ich keinen Lüfter für meine Spezifikationen, der auch Außentemperaturen bis -20 °C auf Dauer aushält. Deshalb habe ich die Lüfter im Innenraum der Lufttauscherbox montiert.

Der Wärmetauscher

Man findet im Internet einige Hersteller von Kreuzwärmetauschern für Eigenentwicklungen. Leider waren meine Möglichkeiten sehr beschränkt, was den Einbauort betrifft, aber ich erinnerte mich an meine Jugend, als man einem Turbo-Diesel-Fahrzeug mithilfe eines Ladeluftkühlers zusätzliche Lebensenergie einhauchen konnte.

Das Prinzip ist einfach, die angesaugte Luft, welche der Turbolader aufheizte, wurde vom Fahrtwind wieder abgekühlt, um eine bessere Verbrennungsleistung zu erreichen. Es findet also ein effektiver Luft-Wärme-Austausch statt. Ein passender Ladeluftkühler (Bild 3) war



Bild 3: Als Wärmetauscher wird ein Ladeluftkühler eingesetzt.

nun schnell gefunden, passende Anschlussschläuche gibt es natürlich auch in allen Bauformen und Längen dazu.

Das Gehäuse

Holz ist das Material, welches sich im Heimbereich ziemlich einfach verarbeiten lässt. Es bringt viele Eigenschaften mit, welche bei meinem Projekt von Vorteil waren. Gute Wärmedämmung und Schallabsorption, wenn man ein weiches Holz wie Kiefer, Fichte oder Birke verwendet. Leider ist die mechanische Verformung bei Feuchtigkeit und hoher Wärme ein unerwünschter Nebeneffekt. Deshalb wurde das verwendete Holz entsprechend vorbehandelt, um das Einziehen von Feuchtigkeit ins Material zu verhindern. Zusätzlich sind die Strömungsbereiche im Innenraum zur Schall- und Wärmeisolierung mit Schaumstoff und aluminiumbeschichteten Matten gedämmt (Bild 4). Diese Matten findet man im Handel, um Heizkörper im Wohnbereich gegen die Außenwand zu isolieren.

Der Aufbau

Aufgrund baulicher Gegebenheiten mussten leider das Zuluft- und das Abluftrohr ziemlich dicht beieinander durch die Wand geführt werden (Bild 5), was zur Folge haben könnte, dass sich draußen die Abluft mit der Zuluft vermischt. Die Lüftungsgitter an der Außenwand wurden deshalb versetzt montiert, um etwas Abhilfe zu schaffen.

Ich denke, dass trotz des baulichen Mankos genügend frischer Sauerstoff angesaugt wird.

Der Abluftventilator ist am hinteren Ende im Gehäuseboden befestigt (Bild 6) und direkt mit einem Abluftrohr nach draußen geführt. Dieser Lüfter zieht die warme Luft durch die Lamellen des Ladeluftkühlers, hier findet der Wärmeaustausch mit der kühlen Außenluft statt, welche den Kühler im Inneren passiert.

Das Ansaugen der frischen Außenluft bewerkstelligt der zweite Lüfter, welcher am anderen Ende in der Ansaugkammer montiert ist. Diese steht in Verbindung mit dem Ausgang des Ladeluftkühlers (Bild 7). Der Eingang des Kühlers führt direkt über einen Silikonschlauch (siehe Bild 3) nach draußen.

In der Nähe der Lüfter befinden sich jeweils 1-Wire-Temperatursensoren vom Typ DS18B22, welche man bequem parallel an einem Eingang des Arduino anschließen kann. Allerdings bedeutet das, dem Sensor eine eindeutige Adresse zu geben, welche man entweder zuvor per Arduino-Programm auslesen muss oder vom Hersteller bescheinigt bekommt.

Die Innen- und Außentemperatur wird durch jeweils einen kombinierten Temperatur-Feuchte-Sensor vom Typ DHT22 (Bild 8) gemessen. Diese sind einmal außen im Ansauggitter und innen im Innenraum platziert.

Leider ist der Innenraumsensor in meinem Fall etwas niedriger platziert als der Sensor des Zuluftventilators, was daher keinen direkten Temperaturvergleich zulässt. Denn gerade in Deckennähe sieht man deutlich den Temperaturanstieg, welcher auf wenigen Zentimetern vertikal festzustellen ist.



Bild 4: Die Schall- und Wärmeisolierung des Wärmetauschers



Bild 5: Ab- und Zuluftöffnung liegen dicht nebeneinander in der Außenwand und sind deshalb versetzt montiert. Zusätzlich sind die Zu- und Abluftrohre in der Wand mit leichtem Gefälle eingebaut, um entstehendes Kondensat nach außen abzuleiten. Außerdem wurden in den Luftkanälen Edelstahlrückschlagklappen installiert, um die Wärmedämmung zu verbessern.



Bild 6: Der Abluftventilator ist am hinteren Ende im Gehäuseboden befestigt.

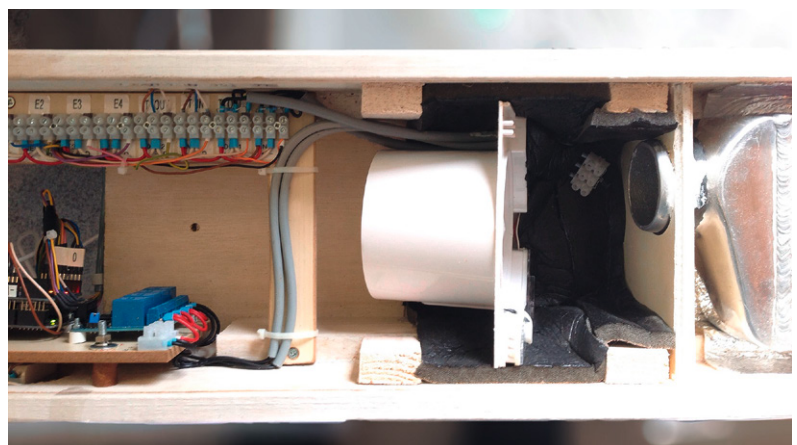


Bild 7: Die Montage des Ansauglüfters in einer eigenen Kammer

Abgeglichen wurden alle Sensoren durch einen 24-Stunden-Trockentest. Das heißt, die fertige Box wurde einen Tag lang waagrecht in einen Raum mit gleichmäßiger Raumtemperatur gelegt und Messdaten aufgezeichnet. Daraus wurde ein Mittelwert gebildet und alle vier Sensoren entsprechend angepasst. Die Feuchtwerte sind ebenfalls gemittelt und zusätzlich noch mit einem externen Feuchtemesser abgeglichen.

Zum Umschalten zwischen Hand- und Automatikbetrieb wurden zwei Schalter ins Gehäuse eingepasst (Bild 9).

Der erste Schalter schaltet den Automatikbetrieb für beide Lüfter ein, der zweite ermöglicht es, beide Lüfter dauerhaft einzuschalten. Beide in Stellung „AUS“ bedeutet auch: beide Lüfter aus bzw. nur reine Datenaufzeichnung der Messwerte.

Die Bildfolge in Bild 10 zeigt noch einmal den Einbau des Wärmetauschers in den Fenstersturz und Installationsdetails der Steuerung, während Bild 11 die Gesamtschaltung zeigt.

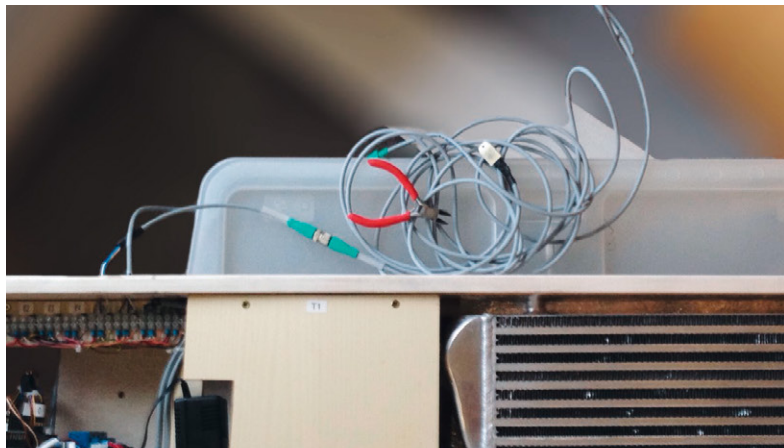


Bild 8: Die extern zu montierenden Sensoren



Bild 9: Über zwei Schalter werden die Betriebsarten eingestellt.



Bild 10: Die Montage des Wärmetauschers und der Steuerungskomponenten

Wenden wir uns nun dem Steuerungsprogramm zu.

Der Arduino-Sketch

Das Steuerungsprogramm (aus Platzgründen hier nicht abgebildet, steht zum Download bereit im Onlinebereich des ELVjournal) folgt diesem prinzipiellen Ablauf:

1. Initialisierung
2. Einlesen der Sensoren
3. Uhrzeitabgleich mit Internetzeitserver
4. Berechnungen und Automatische Lüftersteuerung
5. Daten zum Webserver schreiben

Das Setup

Zunächst ist es wichtig, die richtigen Libraries aus dem Web zu laden und diese zu installieren, damit man nicht unnötig Nerven bei der Fehlersuche verschwendet. Man benötigt hierzu die Libraries SPI, Ethernet, DallasTemperature, Onewire, dht, Wire, EthernetUdp und Time.

Im Internet findet man die unterschiedlichsten Versionen und muss diese auch passend zur Version der Arduino-Software installieren.

Ich nutze die Arduino-Software Version 1.5.4 sowohl auf Windows 7 als auch auf Mac OS X zusammen mit Java 7.

Die #include-Anweisungen binden zunächst alle genutzten Libraries ein. Danach füttern wir den C-Präprozessor mit allen festen Werten, um Speicher zu sparen.

Gerade beim Speicher kann man, wie erwähnt, je nach Arduino-Board Probleme bekommen, wenn das Programm dann doch etwas umfangreicher wird.

Die Pins bekommen sowohl einen festen Wert als auch Werte für die Taupunkt-Berechnung und die Offsetwerte für den Abgleich der Temperatursensoren.

„Onewire“ initialisiert die einzelnen ds1820-Temperatursensoren, wobei jedem Sensor eine Device-Address zugewiesen wird.

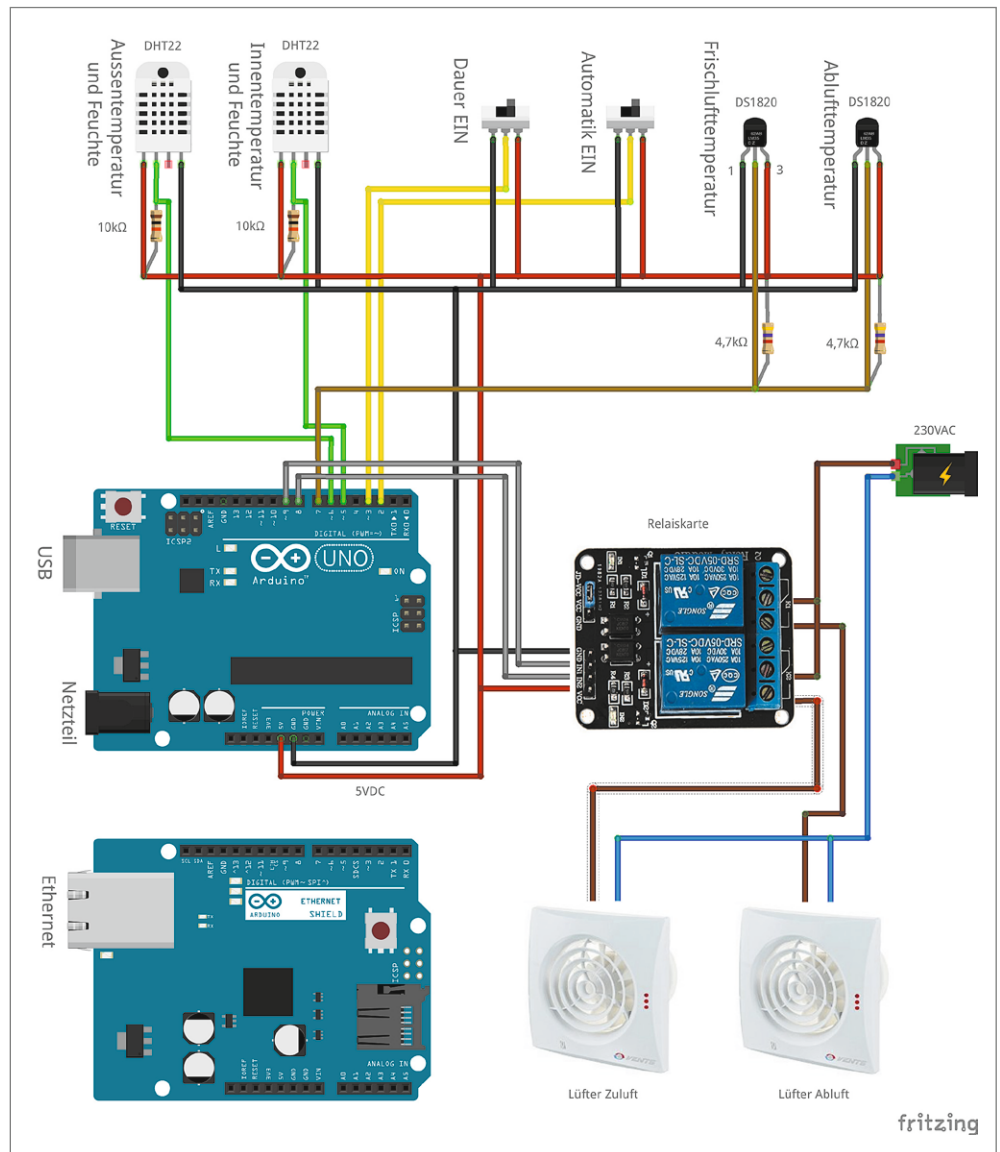
Diese Adressen ermittelt man am besten mit dem Arduino-Sketch von [2].

Danach folgen die Ethernet-Einstellungen. Die MAC-Adresse sollte auf dem Ethernet-Shield aufgedruckt sein. Eine feste IP-Adresse holt man sich aus seinem Heim-Netzwerkrouter.

Die IP-Adresse des Time-Servers von der RWTH Aachen lautet 134.130.4.17, wobei man natürlich auch jeden anderen Timeserver verwenden kann. Je nachdem, welcher am nächsten liegt bzw. die höchste Zuverlässigkeit bietet.



Bild 11: Der Schaltplan für die Steuerung des Wärmetauschers. Erstellt mit Fritzing.



Da die Zeitübertragung über UDP-Transport läuft, geben wir den lokalen Port 8888 an. Die NTP_Packet_Size beträgt 48 Byte. Danach starten wir die Ethernet-UDP-Instanz und setzen alle Zeitvariablen auf null.

Nun kommt der Ethernet-Client an die Reihe, wo alle erforderlichen Adressen und Passwörter festgelegt werden. Hierbei sollte möglichst vermieden werden, das Passwort per Klartext zu übertragen, sondern dieses für die Übertragung zu verschlüsseln.

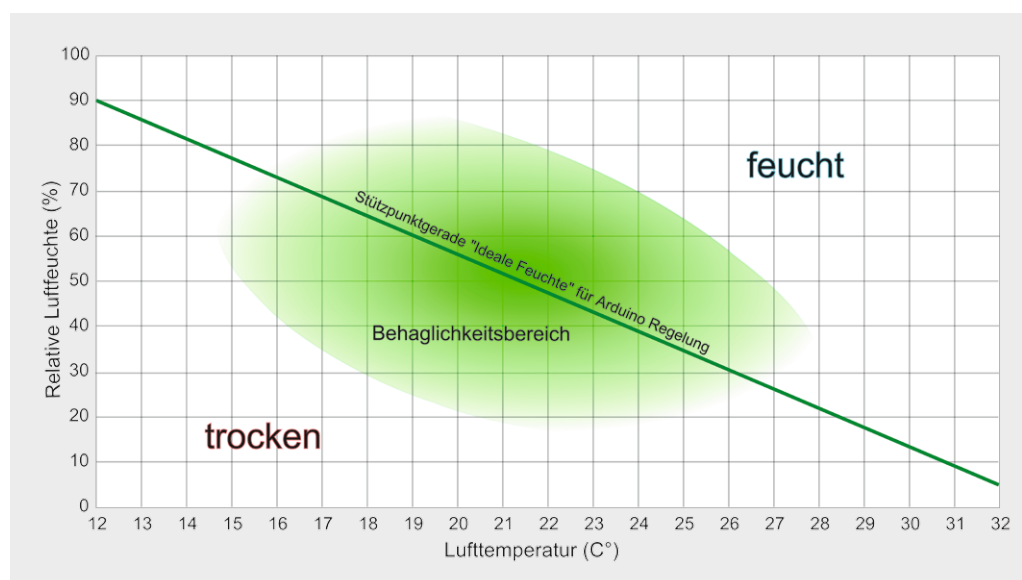


Bild 12: Die grüne Linie zeigt den optimalen Wertebereich für gutes Raumklima.

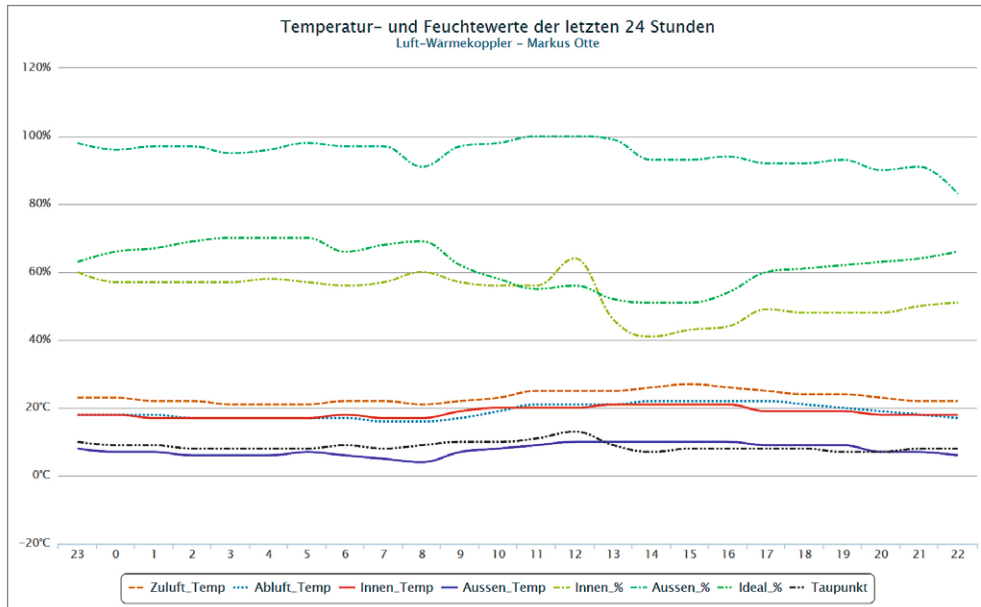


Bild 13: Die Datenaufzeichnung erfolgt über einen Webserver.

Anschließend werden alle für das Programm notwendigen Variablen deklariert.
Nun schalten wir die Pins per Software gegen Masse, initialisieren alle Sensoren und den Ethernet-Client.

Hauptprogramm

Um zeitgesteuert verschiedene Programmschritte erledigen zu können, führen wir im Hauptprogramm vor jedem Programmblock einfach einen Vergleich per „IF THEN“-Abfrage auf Minute und Sekunde und/oder Stunde durch.

Dieses entlastet zum einen den Prozessor, weil nicht in jedem Durchlauf alles ausgeführt wird, und zum anderen wird man im Ablauf flexibler.

Natürlich könnte man auch die Arduino-Zeit „millis()“ nutzen, was ich in meinem speziellen Fall nicht getan habe.

Nach 10 Sekunden startet das Auslesen der DS1820- und DHT22-Sensoren.

Der Taupunkt wird dann näherungsweise per Formel ermittelt:

dht1_t = Temperatur des DHT22-Sensors,
dht1_h = Feuchte des DHT22-Sensors,
aTp = 17.271; ermittelt aus dem Datenblatt
bTp = 237.7; ermittelt aus dem Datenblatt

$$\text{taupunktTmp} = (\text{aTp} * \text{dht1_t}) / (\text{bTp} + \text{dht1_t}) + \log(\text{dht1_h} / 100)$$

$$\text{taupunkt} = (\text{bTp} * \text{taupunktTmp}) / (\text{aTp} - \text{taupunktTmp})$$

Diese Formel entdeckte ich ebenfalls durch Internetrecherche [5].

Um die ideale Wohlfühlfeuchte ebenfalls näherungsweise zu ermitteln, habe ich mir ein Diagramm aus dem Heizungs-/Klimabereich besorgt und eine Gerade durch den Behaglichkeitsbereich gezogen (Bild 12). Dort ermittelte ich den Wert dH von -4,2857 %/Grad, beginnend ab 12° Innentemperatur und 90 % Luftfeuchte.

Nach 20 Sekunden und wiederholt alle 20 Minuten findet die Abfrage des NTP-Zeitserver und das Setzen der aktuellen Uhrzeit im Programm statt. Der Webserver wird somit nach jeder Stunde mit neuen Temperatur- und Feuchtedaten versorgt. Bild 13 zeigt eine entsprechende Datenaufzeichnung.

Zum Automatikbetrieb muss der erste Schalter eingeschaltet sein. Als Regelalgorithmus schalten, ausgehend von der idealen Feuchte, um 10 % zu hohe bzw. zu niedrige Luftfeuchten im Innenraum beide Lüfter ein.



Weiterhin findet alle paar Stunden eine Zwangslüftung statt. Dies kann natürlich je nach Raumnutzung flexibel angepasst werden. Manuell könnte man die Lüfter mit dem zweiten Schalter in Rotation versetzen.

Unterprogramme

Nun folgen die Unterprogramme Webserver, Datensenden (Webserver/Serielle Schnittstelle), DS1820 auslesen, DHT22 auslesen, Automatik (Lüfter schalten), Feuchterege- lung, Zeitzähler zur einfachen Simulation der Uhrzeit.

Damit die Sekunden der Uhr auch im rich- tigen Takt laufen, was durch unterschiedli- che Zykluszeiten ohne Nutzung der millis()- Funktion schwierig ist, findet ein Abgleich durch eine Pause von 800 ms (delay(800)) in jedem Zyklus statt. Das bedeutet, ein Zyklus hat eine Gesamtdauer von einer Se- kunde. Dies kann je nach Programmgröße oder Aufrufen externer Geräte einen weite- ren Abgleich notwendig machen.

Um die interne Arduino-Uhrzeit halb- wegs reell zu gestalten, wird alle 20 Minu- ten ein neuer Zeitabgleich mit dem NTP- Server durchgeführt. Natürlich könnte man, um dies zu perfektionieren, einen DCF77- Uhrzeitempfänger einbauen.

Weitere Informationen, auch zu weiteren Projekten, finden sich auf meiner Home- page [6].

ELV



Weitere Infos:

- [1] www.fluuux.de
- [2] fluuux.de/2012/09/arduino-adressen-aller-ds1820-ermitteln
- [3] www.amazon.de/emall-supply-Computer-Zubehör
- [4] www.amazon.de/Badlüfter-SKS24-Badezimmer-Zubehör
- [5] www.macherzin.net
- [6] www.markusotte.de

www.minicomputer.elv.de

Verwendete Produkte/Bauteile:

Arduino Uno R3

Ethernet-Shield für Arduino UNO

4-Kanal-5-V-Relais-Modul für Arduino TTL-Logik:

Hochstrom-Relais: AC 250 V 10 A, DC 30 V 10 A

Stromverbrauch pro Kanal: 15–20 mA

Anbieter: [3]

Temperatur-/Feuchtesensor DHT22 (Alternativ AM2302):

Versorgungsspannung: 3,3–6 V

Messstrom: 1,5 mA

Stand-by-Strom: 50 µA

Luftfeuchtigkeit: von 0 bis 100 % (relative Luftfeuchtigkeit)

Temperaturbereich: -40 bis +80 °C

Genauigkeit: ±2 % rF, ±0,5 °C

Temperatursensor DS1820:

Versorgungsspannung: 3,3–5 V

Temperaturbereich: -55 bis +125 °C

Genauigkeit: ±0,5 °C bei -10 bis +85 °C

Ventilator 100 Quiet V1 (Nachfolger ist V2)

Nennweite: 100 mm

Spannung: 230 V

Lautstärke: max. 25 dB(A)

Verbrauch: 7,5 W

Luftleistung: 97 m³/h

Umdrehungen: 2165 rpm

Schutzart: IP45

Anbieter: [4]

2 Einbauschalter Typ z. B. Gira/Siemens/Busch Jäger etc.

Machen Sie mit!

Jede veröffentlichte Anwendung wird mit einem Warengutschein in Höhe von 200 Euro belohnt.



Wir wollen es wissen – Ihre Anwendungen und Applikationen!

Welche eigenen kreativen Anwendungen und Applikationen haben Sie mit den ELV-Haustechnik-Systemen, aber auch mit anderen Produkten und Bausätzen realisiert – ob mit Standard-Bausteinen oder eingebunden in eigene Applikationen? Alles, was nicht gegen Gesetze oder z. B. VDE-Vorschriften verstößt, ist interessant. Denn viele Applikationen verhelfen sicher anderen zum Aha-Erlebnis und zur eigenen Lösung.

Schreiben Sie uns, fotografieren Sie Ihre Applikation, berichten Sie von Ihren Erfahrungen und Lösungen. Die interessantesten Anwendungen werden redaktionell bearbeitet und im ELVjournal mit Nennung des Namens vorgestellt.



Per E-Mail
leserwettbewerb@elv.de



Per Post
ELV Elektronik AG, Leserwettbewerb, 26787 Leer

Die Auswahl der Veröffentlichungen wird allein durch die ELV-Redaktion ausschließlich nach Originalität, praktischem Nutzen und realisierter bzw. dokumentierter Ausführung vorgenommen, es besteht kein Anspruch auf Veröffentlichung, auch bei themengleichen Lösungen. **Der Rechtsweg ist ausgeschlossen.** Für Ansprüche Dritter, Beschädigung und Verlust der Einsendungen wird keine Haftung übernommen. Alle Rechte an Fotos, Unter- lagen usw. müssen beim Einsender liegen. Die eingesandten Unterlagen und Aufnahmen verbleiben bei der ELV Elektronik AG und können von dieser für Veröffentlichungen und zu Werbezwecken genutzt werden.