

Funktions+Aufbaubeschreibung der Lüftersteuerung-DB0NI

Motivation:

Bisher war in der Tür des Schrankes ein 12x12cm² Ventilator für ~230Volt der immer gelaufen ist eingebaut. Da aber über das Jahr nur an wenigen Tagen, wenn es wirklich warm ist und die Sonne auf das Dach scheint, eine Lüftung des Schrankes sinnvoll erscheint, wurde diese temperaturabhängige Steuerung für einen 12Volt-DC Ventilator entwickelt. Besondere Eigenschaft ist, dass die Drehzahl von 40°C = Mindestdrehzahl bis 60°C = Maximaldrehzahl verändert wird. Die Regelung ist digital (PWM, digital Sensor DS18B20) und dadurch extrem verlustarm.

Für wen ist diese Beschreibung?

Für mich und erfahrene Bastler. Keine 100% klappt Baubeschreibung. Keine Funktionsgarantie. Kein Support.

Besonderer Hinweis:

Der Sensor soll an einer Stelle montiert werden wo die Temperatur am höchsten ist und sich **NICHT** im Luftstrom befindet. Dadurch wird vermieden, dass der Sensor die Außenluft Temperatur anstatt der gewünschten Innenluft Temperatur im Schrank misst.

Hinweis zum verwendeten Sensor mit Schutzkappe:

Die Aussage „Wasserdicht“ ist nicht zutreffend. Ein eintauchen zB. in das Aquarium führt zur Zerstörung.





Sollte das gewünscht sein muss der Schrumpfschlauch entfernt und durch einen mit innen Verklebung ersetzt werden.

Voraussetzungen:

- a) 12Volt DC, alternativ 12Volt Stecker Netzteil mit mindestens 0,5Amp.
- b) 12x12cm² Lüfter mit 12Volt-ca.0,1 ... 0,3Amp. Achtung; nicht jeder Lüfter ist geeignet! Es muss besonders darauf geachtet werden das selbiger bereits bei sehr kleinen Spannungen (<3Volt) anläuft zB. PC-Gehäuse-Lüfter. *Typische Angaben eines Herstellers:* Operating Voltage Range – 4,5~13,2VDC. Starting Voltage – 4,5VDC(@)25°C. Reicht nicht! Wird ein anderer verwendet bleibt er bei Temperaturen zw. 40-50°C stehen.
- c) Sollte es zu Störgeräuschen im RX kommen ist zusätzlich mit geeigneten Ferritkernen die HF an den drei Leitungen zur Elektronik zu verdrosseln.

Erforderliche Hardware:

Stk.	Wert	Bezeichnung	Quelle	Preis ca.	Bild
1 x	ATtiny85	Controller	diverse	1,50€	
1x	8.pol.	IC-Fassung	Diverse, Reichelt	0,25€	
Opt. 1 x	CP2102 STC	USB<->Ser Konverter	https://www.ebay.de/itm/152895017873	2,00€	
Opt. 1 x	Pro- grammer	USBasp https://www.fischl.de/usbasp/	Diverse, ebay	ca. 10€	
1 x	1N5817	Shottky Diode	Diverse, Reichelt	ca. 0,05€	
1 x	1N400x	Netzdiode	Diverse, Reichelt	ca. 0,04€	
2 x	4,7k Ohm	Widerstand	Diverse, Reichelt	ca. 0,08€	
4 x	2,2k Ohm	Widerstand	Diverse, Reichelt	ca. 0,08€	
1 x	100 Ohm	Widerstand	Diverse, Reichelt	ca. 0,08€	
1x	BUZ11	FET-Power	Diverse, Reichelt	ca. 1,00€	
1x	7805	Spg.Stabi-IC	Diverse, Reichelt	ca. 0,30€	
1 x	DS18B20 Sensor	Digitaler Temperatur S.	Diverse, Reichelt	ca. 3,00€	
1 x	12Volt	Lüfter s.o.	Diverse, Reichelt	ca. 10 €	
1 x	geeignetes	Gehäuse	Diverse, Reichelt	ca. 10 €	
20 x	geignete	Lötstützpunkt	Diverse, Reichelt	ca. 1,00€	
1 x	Punkte	LochrasterLpt	Diverse, Reichelt	ca. 10 €	
4 x	rt.ge.gr.am.	LED's 5mm	Diverse, Reichelt	ca. 0,10€	
1 x	82µF/ >15Volt	Elko, RAD o.ä.	Diverse, Reichelt	ca. 0,40€	
1 x	68µF/ >6Volt	Elko, RAD o.ä.	Diverse, Reichelt	ca. 0,40€	
3 x	ca.0,1µF/ >20Volt	Keramik-C, RAD o.ä.	Diverse, Reichelt	ca. 0,10€	
Diversen Kleinkram zum zusammenlöten, Drähte, ...					

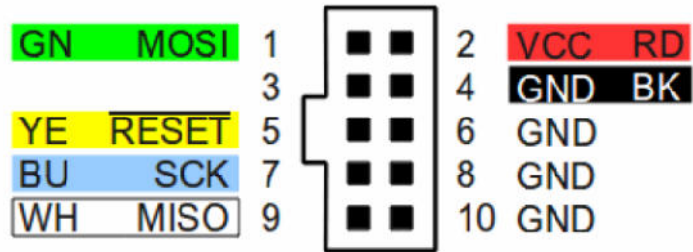
Erforderliche Software:

Das File „Lueftersteuerung-DB0NI-DS18B20.ino“ welches auf den ATtiny85 programmiert werden soll.

Hinweis zur Programmierung des ATtiny85: siehe <https://www.fischl.de/usbsp/>

Anschlüsse:

USBasp-Programmer		ATtiny85
MOSI Pin 1	<->	Pin 5 PB0
RESET Pin 5	<->	Pin 1 PB5
SCK Pin 7	<->	Pin 7 PB2
MISO Pin 9	<->	Pin 6 PB1
VCC Pin 2	<->	Pin 8 VCC
GND Pin 4	<->	Pin 4 GND



Schau mal da:

<https://wolles-elektronikkiste.de/attiny-mit-arduino-code-programmieren>

<https://wolles-elektronikkiste.de/attiny85-84-45-44-25-24-programmieren>

<https://www.kollino.de/arduino/attiny85-programmieren-mit-arduino-uno-oder-usb-programmer/>

und viele andere mehr. Den USBasp gibt es bei vielen Internet Händlern für kleines Geld. Firmware update nicht vergessen. Ebenso den USB-Serial Adapter.

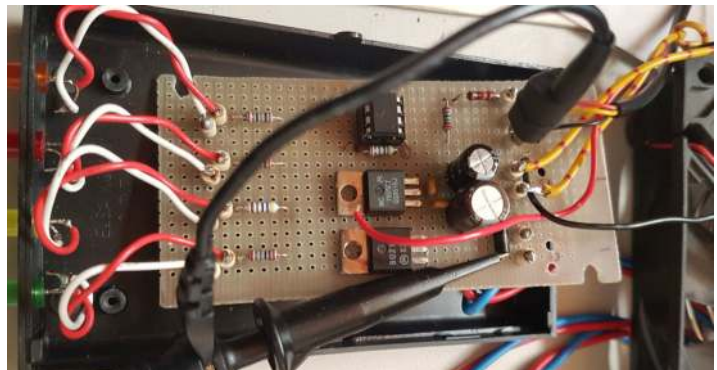
NB: Es gibt auch andere Möglichkeiten, zB. einen Arduino Uno zu verwenden.

<https://www.heise.de/hintergrund/Arduino-Uno-als-In-System-Programmer-2769246.html>

Der mechanische Aufbau in einem Kunststoffgehäuse:

Ist intuitiv erfolgt.

Das Bild kann einen Anhalt ergeben. Auf der einen Seite, links die LED's für die Status Anzeigen. Rechts dann die Anschlüsse für den Sensor, den Lüfter und die 12Volt DC.



Die elektrische Verschaltung:

* Anschließen der 12Volt Versorgung. Falls verpolt passiert nichts. Die Stromaufnahme der Elektronik beträgt nur wenige mA so dass der Lüfter Ausschlag gebend ist.

* Den Lüfter bitte auch richtig gepolt anschließen.

* Der digitale Sensor DS18B20 ist bereits fertig verdrahtet und kann an einem geeignetem Platz (NICHT im Luftstrom, an einem Hotspot) montiert werden.

Funktionsablauf:

Nach dem Anlegen der Versorgungsspannung von 12 Volt leuchten zunächst als Funktionskontrolle die vier LED's zwei mal kurz auf. Die grüne LED zeigt das vorhanden sein der 12Volt an. Danach wird der Lüfter mit maximaler Drehzahl

gestartet und intern eine hohe Temperatur simuliert. Durch das integrative Verhalten der temperaturabhängigen Steuerung des Lüfters beginnt nun eine langsame Verringerung der Drehzahl bis zum Erreichen der Mindestdrehzahl. Wenn die Nachlaufzeit abgelaufen ist stoppt der Lüfter, es sei denn die Temperatur ist hoch genug um den Lüfter im geregelten Zustand mit der dazu passenden Drehzahl am Laufen zu halten. Die jeweiligen Temperaturbereiche werden durch die drei gelben, roten und amber LED's angezeigt.

Im normalen Betrieb wird beim Überschreiten der unteren Temperaturgrenze (40°C) der Lüfter zunächst mit maximaler Drehzahl gestartet damit er sicher anläuft um nach ca. drei Sekunden mit der Drehzahl von 22% ... 100% der sich ändernden Temperatur zu folgen. Die gelbe LED leuchtet. Beim Überschreiten der oberen Temperaturgrenze (60°C) wird die maximale Drehzahl beibehalten und bei weiterem Anstieg eine zunächst langsam und dann immer schneller blinkende rote Warn-LED angesteuert. Eine sonstige Warnausgabe ist denkbar. Wird bei einem weiteren Anstieg der Temperatur die NOT-AUS-Grenze erreicht, wird die amber LED angesteuert. Eine sonstige NOT-AUS-Ausgabe ist denkbar. Sinkt die Temperatur wird die Lüfter-Drehzahl entsprechend verringert bis sie nach dem Unterschreiten der unteren Grenze nach der Nachlaufzeit auf null gestellt wird. Die gelbe LED erlischt.

Sollte sich im normalen Betrieb die Temperatur längere Zeit im aktiven Bereich konstant halten wird als Folge auch die Drehzahl des Lüfters konstant sein. Das kann zur Folge haben dass es beim Lüfter zu einem vorzeitigen Lagerschaden kommen kann. Um dem vorzubeugen wird die Drehzahl automatisch zufällig um einen geringen Betrag verringert bzw. erhöht. Es wird dabei beachtet dass die Drehzahl immer im gewünschten Bereich bleibt. Das gilt auch für die maximale bzw. minimale Drehzahl.

Sollte es erforderlich sein, zB. bei einer regelmäßigen Wartung, zu prüfen ob der Ventilator einwandfrei läuft, ist die RESET-Funktion (Taster) herausgeführt.

Nach dem Tasten wird der Ablauf wie nach dem Einschalten ausgeführt.

Sensor Überwachung: Ist der Sensor defekt oder nicht angeschlossen ist eine Regelung nicht möglich. In der Folge wird der Lüfter auf maximale Drehzahl eingestellt und die rote LED blinkt jede Sekunde zwei mal.

Kann realisiert werden:

Zweiter Sensor? Ist möglich wenn ein zweiter Temperatur – Hotspot existiert.

Nur aktiv wenn der TX läuft?

Intervall Betrieb? Soll der Lüfter regelmäßig die Luft im Schrank austauschen?

Luftstromüberwachung? Sollte, aus welchem Grund auch immer (zB. Filter dicht), es nicht zu einer ausreichenden Luftförderung kommen ist es möglich selbigen zu überwachen.

Debug-Serieller Ausgang:

Am PIN 5 des ATtiny85 werden mit 38,4 KBaudot Informationen über die Temperatur und den Lüfter ausgegeben.

1) =NotausPin →

weil die Temperatur viel zu hoch ist würde eine Abschaltung erfolgen.

2) Sensor-> 200°C/10 →

Am Sensor Ort werden 20°C gemessen.

3) Blink → Die integrierte Regel-Temperatur liegt über 60°C.

4) FanAn → Der Lüfter sollte laufen.

5) FanSpeed: 191 → Die PWM steuert mit 191 von 255 an.

6) 760°C/10 =Regel-Temperatur → Dieser Wert bestimmt die PWM.

7) NachL → Ist die Regel-Temperatur unter 40°C läuft der Lüfter noch 30sec.

Wie gelangt das Programm, daß als SourceCode vorliegt in den ATtiny85? :

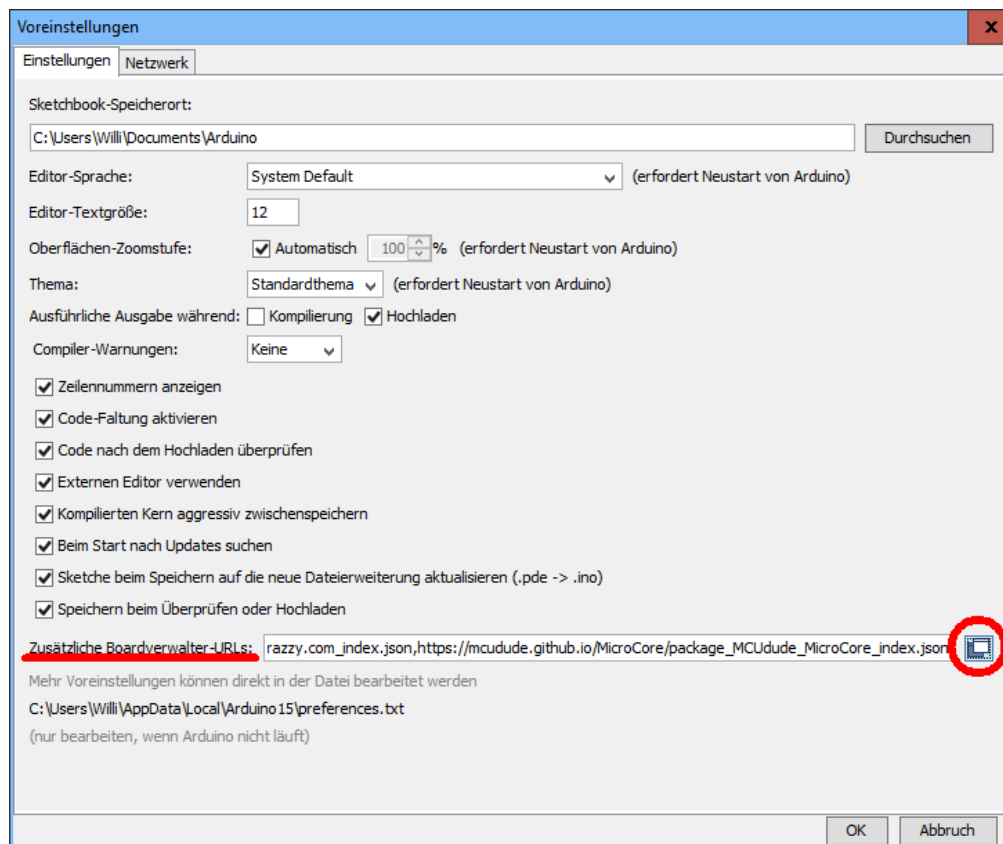
Voraussetzungen:

Ein halbwegs aktueller PC möglichst mit Windows10/11.

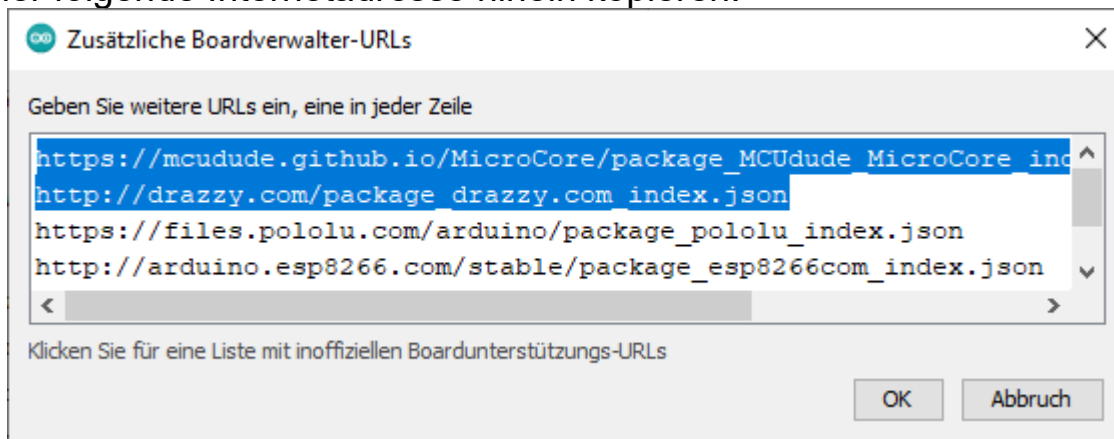
Die ARDUINO – IDE: zB. Version >=1.8.19 von hier:

(<https://www.arduino.cc/en/main/software>) auf dem PC installieren.

Dann die ARDUINO – IDE starten und auf Datei →Voreinstellungen: Klicke rechts auf das Symbol



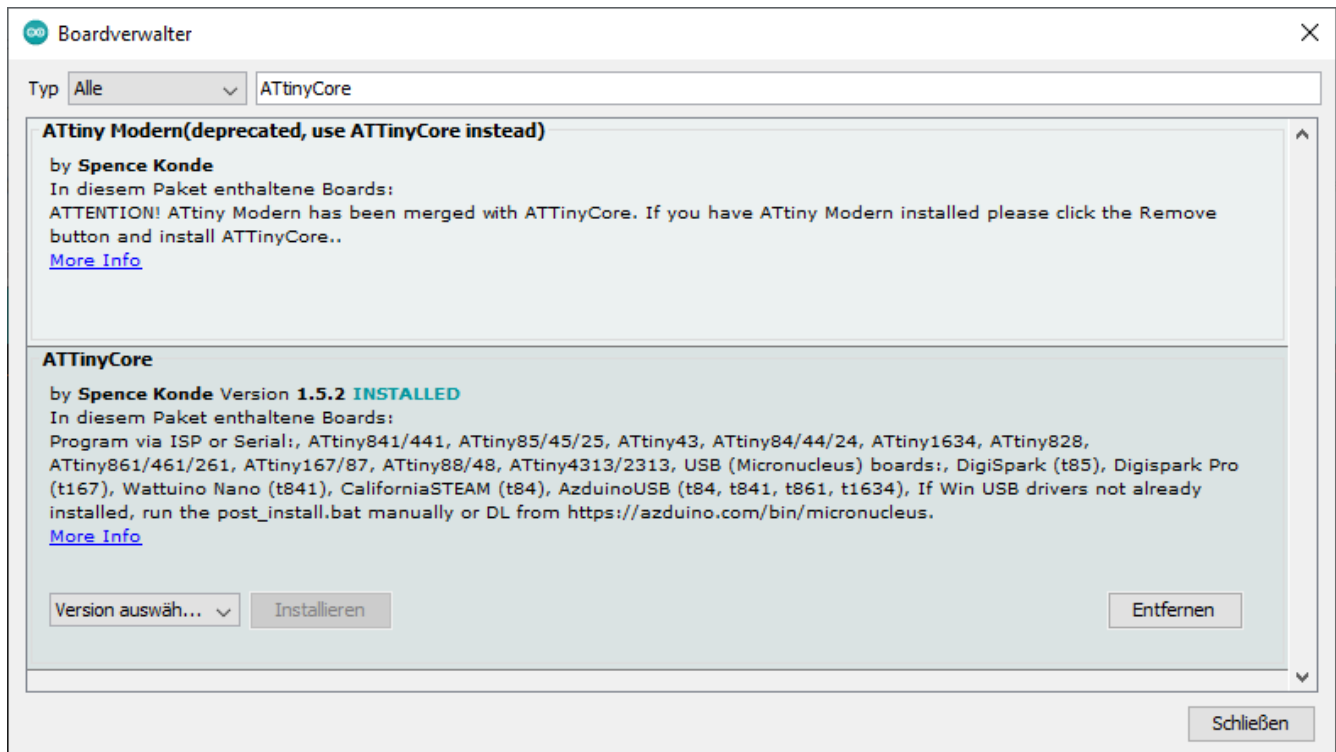
Und hier folgende Internetadresse hinein kopieren.



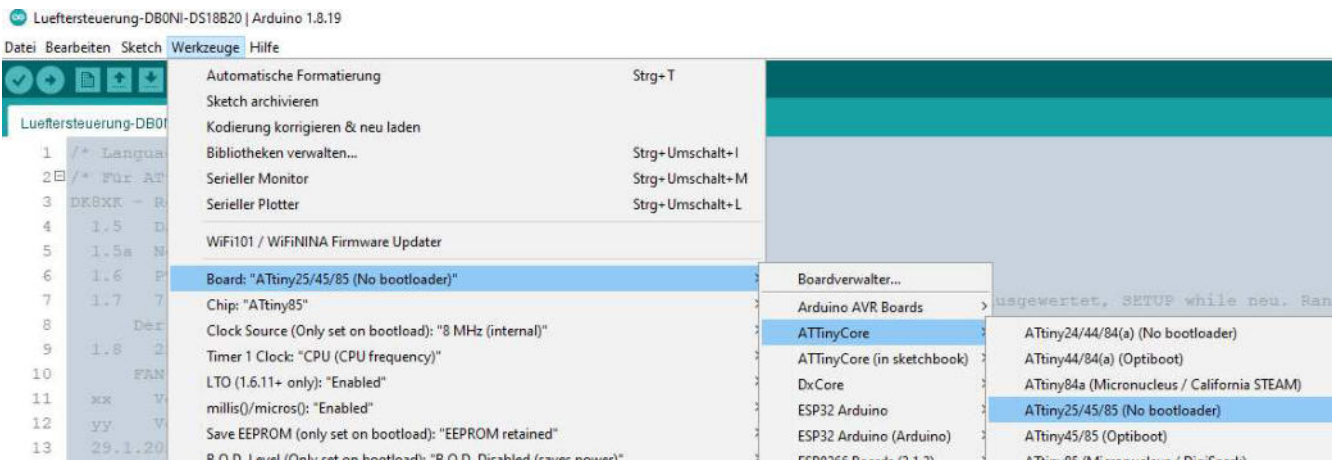
siehe auch hier:

<https://github.com/arduino/Arduino/wiki/Unofficial-list-of-3rd-party-boards-support-urls>

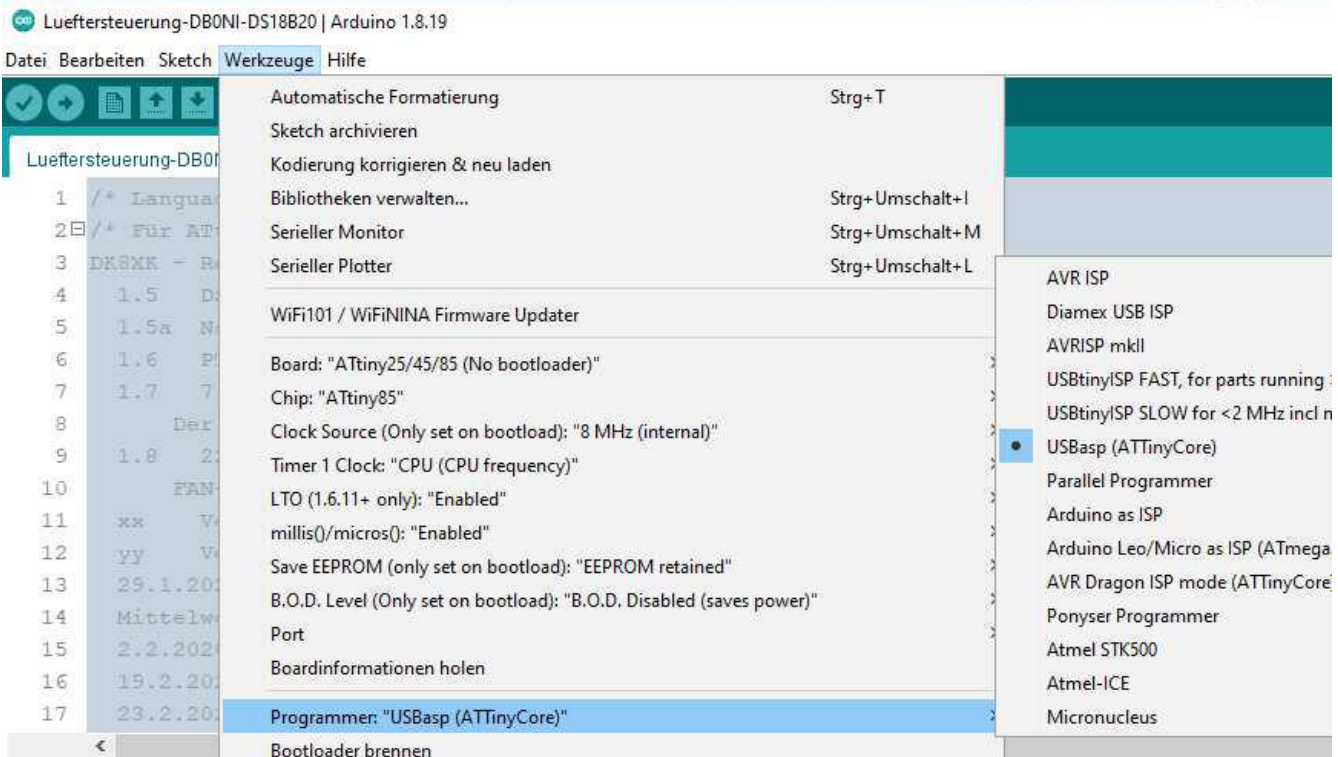
Dann in der Arduino-IDE auf "Werkzeuge" → Boards:"xxxx" → Boardverwalter ATtinyCore im Suchfeld eintippen und den **AttinyCore** by Spencer Konde installieren. Wenn fertig schließen.



So sieht es aus wenn die Boards ATTiny bereits installiert sind.



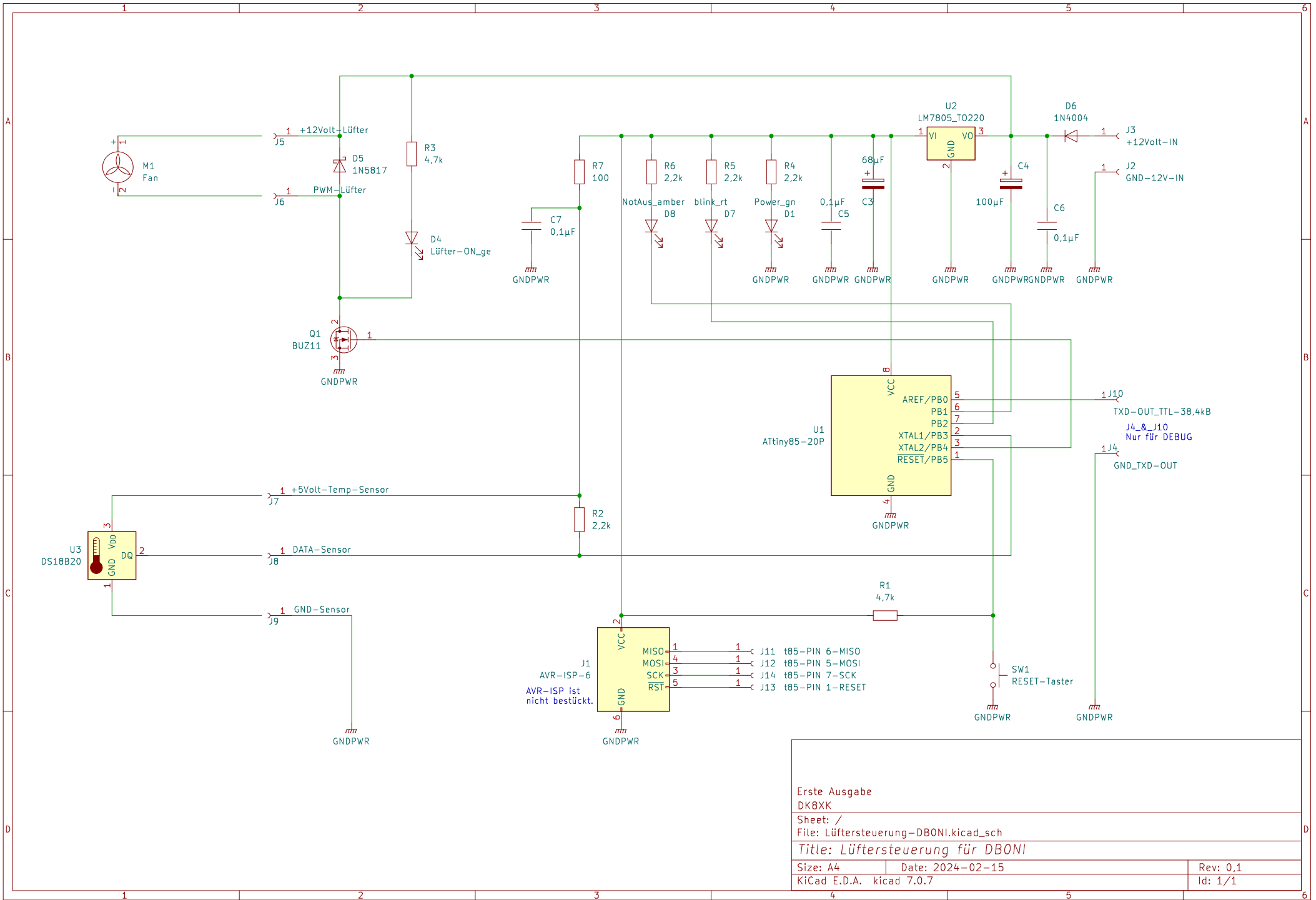
Dann wieder → „**Werkzeuge**“ → **Board** → **ATtiny25/45/85(no bootloader)** wählen.
 Dann noch den USBasp als Programmierer anwählen.
Werkzeuge → **Programmierer** → **USBasp**.



Sind die individuellen Werte eingetragen kann der Sketch „Lueftersteuerung-DB0NI-DS18B20.ino“ in die ARDUINO-IDE geladen werden und zum testen → **Sketch** → „**Überprüfen/Kompilieren**“ oder **Strg+R** testweise kompiliert werden. War die Kompilierung erfolgreich bleibt noch den Attiny85 wie oben beschrieben anzuschließen.

Nun können wir versuchen den Sketch hochzuladen → **Sketch** → „**Hochladen mit Programmier**“ oder **<STRG+Umschalt+U>**.

Die Schaltung der Lüftersteuerung:



Erste Ausgabe
DK8XK

Sheet: /
File: Lüftersteuerung-DBONI.kicad_sch

Title: Lüftersteuerung für DBONI

Size: A4	Date: 2024-02-15	Rev: 0.1
KiCad E.D.A. kicad 7.0.7		Id: 1/1