

# hzrr200

## Dynamische Rücklauftemperaturbegrenzung für Einrohrheizungen.

Version 200

Aufbauanleitung

Kurzbeschreibung

TODO

Änderungen:

Datum	Version	Autor	Beschreibung	Bemerkung
1.7.2020	1.00	P. Loster	erste Zusammenfassung bestehender Beschreibungen, Sensor und Motor Anschluss	
20.7.2020	1.01	P. Loster	Sensor und Motor Test	
23.7.2020	1.02	P. Loster E. Loster	Inbetriebnahme hzrr200 Module	
28.7.2020	1.03	P. Loster E. Loster	Fehler verbessert; Anleitung für Dummies erweitert	
29.7.2020	1.04	E. Loster	Korrekturen, Ergänzungen	
25.8.2020	1.05	E. Loster	Korrekturen	

# INHALT

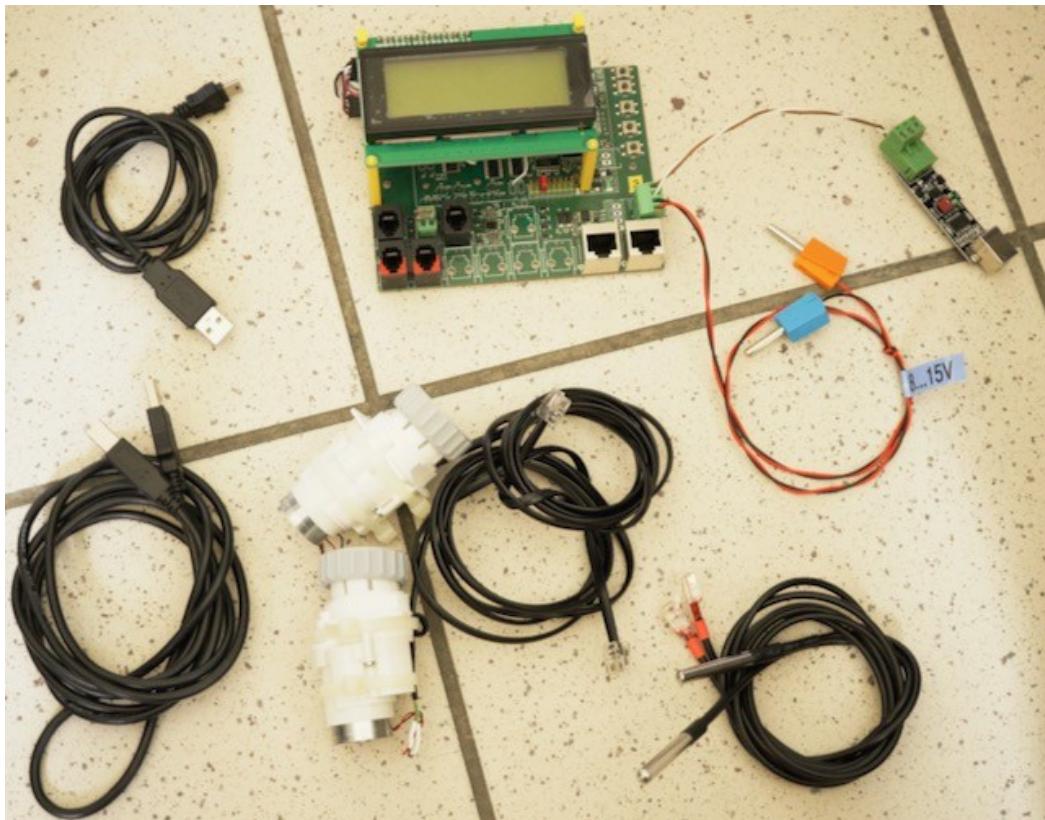
1 Übersicht.....	3
1.1 TODO: Blockschaltbild typ. Gesamtsystem.....	3
1.2 TODO: Kurzbeschreibung der Komponenten.....	3
2 Reglermodul („Modul“).....	3
2.1 Crimpen von Modular Steckverbindern.....	5
2.2 Temperatursensoren DS18B20.....	6
2.2.1 Aufbau und Anschluss.....	6
2.2.2 Eigenschaften.....	6
2.3 Stellmotoren zur Ventilbetätigung.....	6
2.3.1 Anschluss Motor.....	7
2.4 Testgerät für Sensoren und Motoren.....	8
2.5 Aufbau und Inbetriebnahme.....	9
2.5.1 Software zur Inbetriebnahme aufspielen.....	9
2.5.2 Modul aufbauen.....	11
2.5.3 Test Aufbau.....	14
2.5.4 Inbetriebnahme Tests.....	15
2.5.4.1 LCD-Test.....	15
2.5.4.2 Taster-Test (Button-Test).....	16
2.5.4.3 Address- und Temperatursensor-Test.....	16
2.5.4.4 Motor-Test.....	18
2.5.4.5 Kommunikationstest.....	18
2.5.5 Abschluss Inbetriebnahme.....	19
2.6 Anwendungsprogramm Installieren und Prüfen.....	20

# 1 Übersicht

## 1.1 TODO: Blockschaltbild typ. Gesamtsystem

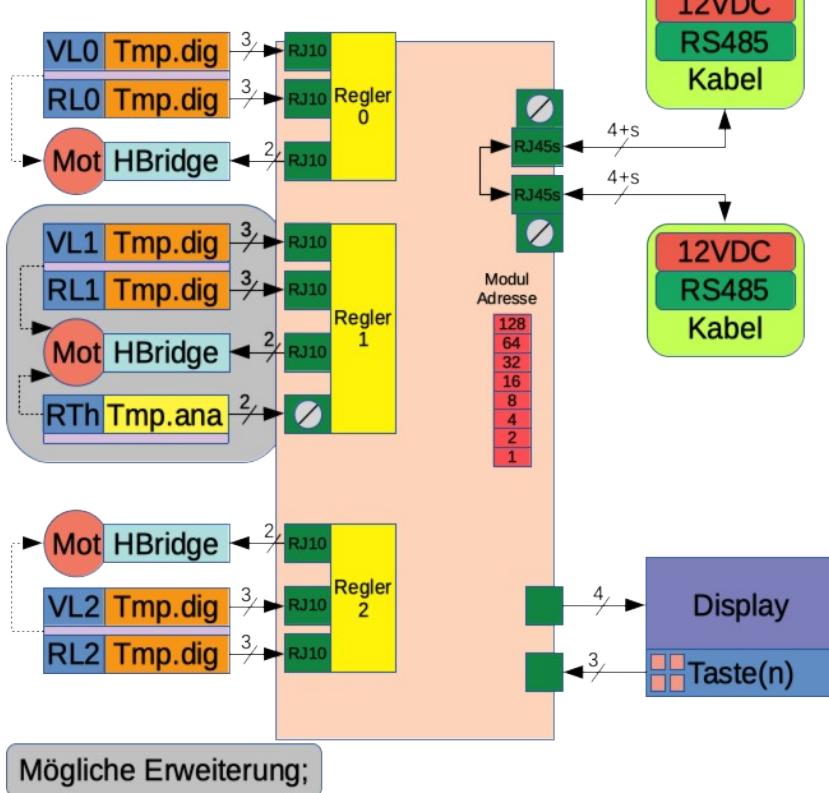
## 1.2 TODO: Kurzbeschreibung der Komponenten

# 2 Reglermodul („Modul“)



*Bild 1: Reglermodul oben Mitte (Prototyp); USB-Kabel oben links zum Programmieren; RS485-USB Adapter und 12V Anschluss oben rechts; USBA-B Kabel f.RS485 Adapter unten links; zwei Stellmotoren unten Mitte und zwei Temperatursensoren unten rechts*

## HZRR-200 Steuermodul



### Arbeitspakete:

- Hardware:
- +Schaltung entwerfen
  - +PCB 1 entwerfen, fertigen, testen
  - +Korrektur
  - PCB 2 fertigen, testen
  - Gehäuse Einbau, Muster
  - Stückliste, Lieferanten, Doku
- Software programmieren:
- +Treiber f. alle Anschlüsse, C++ Klasse(n)
  - Rücklauf (RL) Regelung
  - Raumthermostat (RT) Regelung
  - Anzeige definieren (LCD, 4x20)
  - Display + Taste progr.; Menü
  - Status + Kommandos definieren
  - RS485 Dialog definieren und progr.
  - Speicherung
  - Timing festlegen
  - Prozesse und Ablauf festlegen
  - Programmteile zusammenführen
  - Watchdog progr. und Test
  - Tests, Korrekturen
- Dokumentation
- i Schalt- und Bestückungsplan
  - i Stückliste (BOM)
  - i Aufbauanleitung
  - i Software Struktogramm, Abl.Diagr.
  - i Quellcode dokumentieren
  - e Installationsanleitung
  - e Anschlussbeschreibung
- => Definition und Programmierung der Zentrale

HZRR-200\_Block04.odg  
© Peter Loster, 2020-03-21

Bild 2: Blockschaltbild Reglermodul

Ein Mikrocontroller im Reglermodul (kurz: Modul) enthält bis zu drei Regler Nr. 0,1 und 2. Regler 1 kann auch als Heizkörperthermostat arbeiten. An das Modul sind eine LCD-Anzeige und Tasten angeschlossen. Die Moduladresse wird über Steckbrücken eingestellt. Die Versorgungsspannung von 12V sowie die Datenbusleitung für eine RS485 Schnittstelle werden über RJ45 Modular Stecker und geschirmte min. CAT5E Leitungen hergestellt.

1. Zwei Temperatursensoren, 'Onewire' DS18B20 über RJ10 (4P4) Modular Stecker für Vorlauf (VL) und Rücklauf (RL) Temperatur für jeden Regler Nr. 0, 1, 2 . Anschluss über drei Adern.
  2. Ein 3V Stellmotor für jeden Regler zum Verstellen des Ventils; zwei Adern.
  3. 4 Zeilen zu je 20 Zeichen LCD-Anzeige
  4. 5 Tasten und eine Reset-Taste
  5. 2 RJ45 Stecker zur Stromversorgung mit 12V (2x +12V, 3xGND) und für den RS-485 Bus (2 Pole) sowie die Abschirmung der Leitung. Alternativ können diese Leitungen auch über Steckklemmen angeschlossen werden, von denen eine bestückt ist.
- ACHTUNG:** Es können maximal 30 Module in einem RS485 Netz betrieben werden.
6. **Alternativ kann Regler 1 als Heizkörper Regler mit einem Raumsensor „RAS PT“ der Fa. „Technische Alternative“ betrieben werden.** Die Onewire Temperatursensoren entfallen dann.

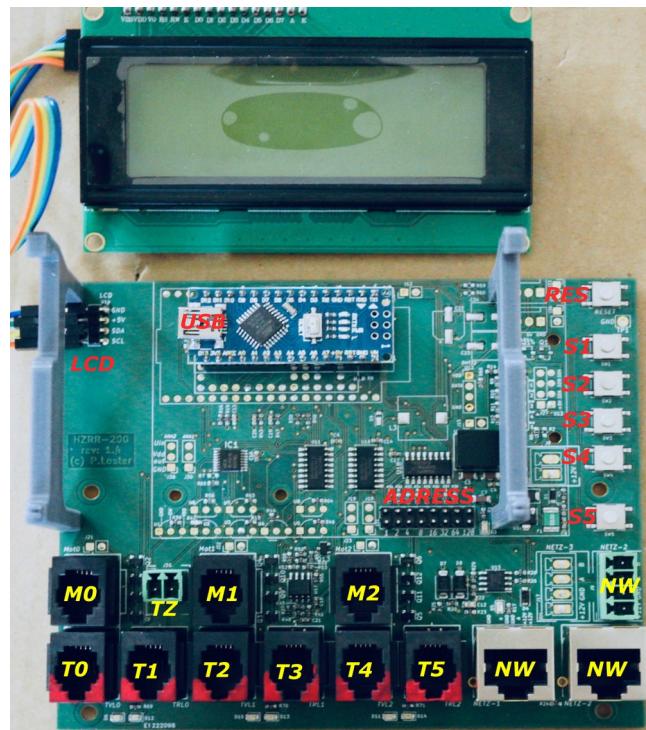


Bild 3: Anschlüsse hzrr200 Regler Modul: LCD, USB, M)otor 0...2, T)emperatur 0...5, wobei T1, T3, T5 blau markiert Rücklauf Temperatur sind; TZimmer, N)etzW)erk, S)chalter, Adressen

## 2.1 Crimpen von Modular Steckverbindern



Bild 4: Crimpzange für 6Px, 8Px und 4Px Modular Stecker (RJ-)



Bild 5: Crimpen eines RJ10 (4P4) Steckers

## 2.2 Temperatursensoren DS18B20

### 2.2.1 Aufbau und Anschluss



Bild 8: Sensor wie geliefert



Bild 7: Sensor fertig;  
rot markiert

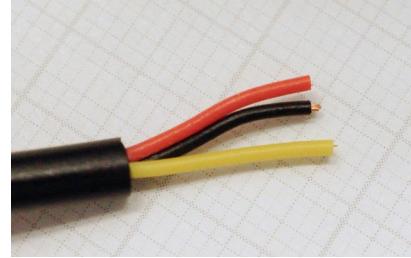


Bild 6: Vorbereitung der  
Kabelenden

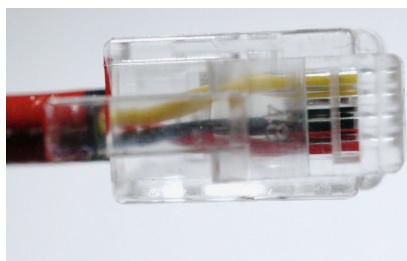


Bild 10: Sensorstecker; Lasche oben



Bild 9: Sensorstecker,  
Lasche oben

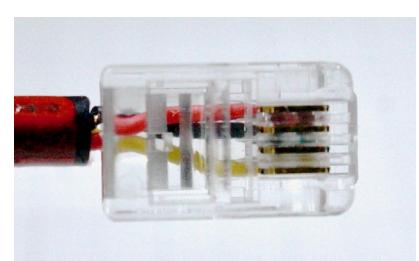


Bild 11: Sensor Stecker,  
Kontaktseite

- Kabel wie in Bild 6 ca. 12mm abisolieren und Kabel nach Farbe etwas aufspreitzen
  - In den RJ10 Stecker wie in Bild 11 Kabel einführen; Pin 3 bleibt frei; dann mit Spezialzange für Modular 4P4 Stecker crimpfen
- ACHTUNG: Verpolen der Anschlüsse zerstört den Sensor !!!**
- Sensor an Testschaltung anschließen und Temperatur prüfen.
  - Stecker oder auch Kabel in Steckernähe mit rotem Lack markieren, wenn Test erfolgreich war, siehe auch Bilder;
- dies ist nötig um die Stecker von den Motorsteckern zu unterscheiden!**

### 2.2.2 Eigenschaften

Siehe Datenblatt DS18B20. Auf wenige zehntel °C genaue Sensoren mit digitaler Messwertübermittlung. Die Leitungslänge verfälscht daher die Messwerte nicht. Um eine Eigenerwärmung der Sensoren durch den Betrieb gering zu halten (vor allem bei Messungen in Luft) werden die Sensoren nur während der Messung kurzfristig mit Strom versorgt.

## 2.3 Stellmotoren zur Ventilbetätigung

Die Stellmotoren stammen aus elektronischen Heizungsthermostaten mit 2x1,5V Versorgungsspannung. 2 Adern eines 4-poligen Spezialkabels für Modular Stecker (RJ10) 4P4 werden an den Motor angelötet und auf der anderen Seite an den Stecker gecrimpt.

### 2.3.1 Anschluss Motor

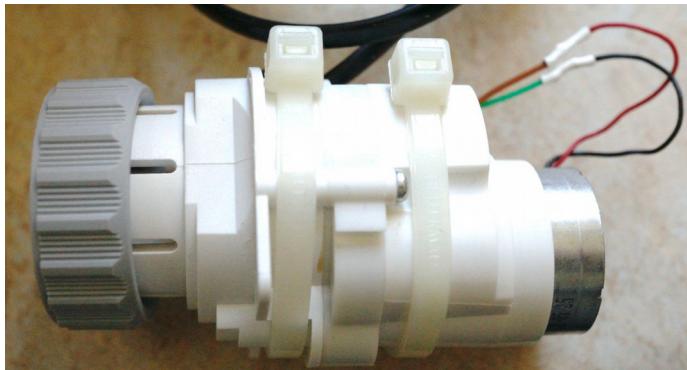


Bild 12: Stellmotor für 'Heimann' Ventil

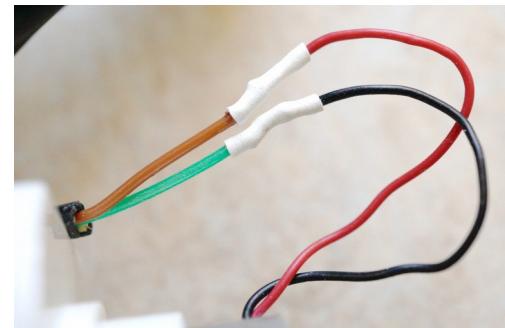


Bild 13: Anschluss rot-braun und grün-schwarz an Motor

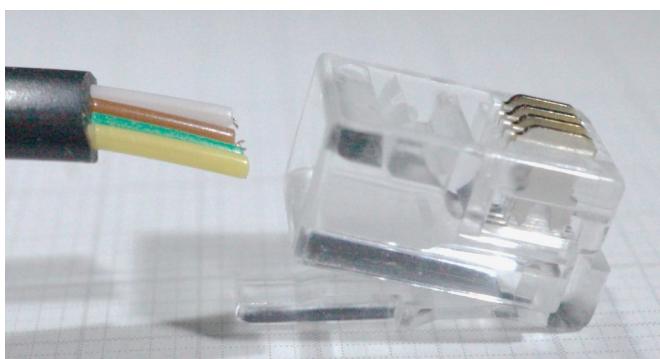


Bild 14: Motorstecker: Kabel in Modular 4P4 Stecker einführen - POLUNG !!!



Bild 15: Motorstecker: 6mm Mantel entfernen

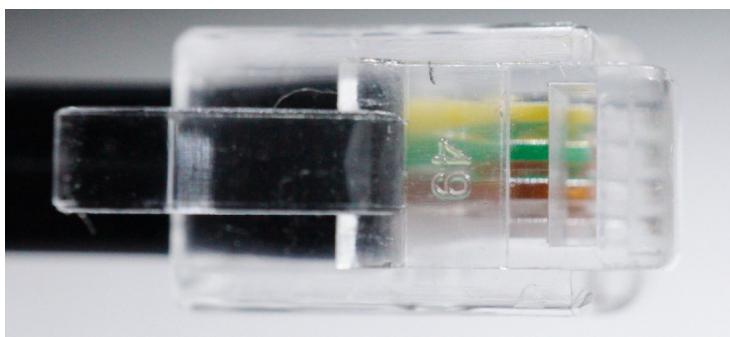


Bild 16: fertig gecrimpter Stecker



Bild 17: Seitenansicht, Feder unten - gelb sichtbar

**WICHTIG: Auf richtige Polung achten !!! Andernfalls laufen die Ventile falsch rum**

## 2.4 Testgerät für Sensoren und Motoren

Ein Reglermodul wurde zum Test der Funktion von Temperatursensoren und Motoren nach deren Montage mit einer speziellen Test-Software versehen.



*Bild 18: Stromversorgung über 5V USB Kabel am Mikrocontroller des Sensormoduls; Temperatursensoren können unten links an den Modular Steckern (4P4) rot gekennzeichnet angeschlossen werden, darüber über 4P4 Stecker schwarz die Motoranschlüsse und dazwischen der Anschluss für den Raumtemperatur Fühler.*

Motor&Sensor test  
+ FW:0.14.2020-07-19  
HW:0.1+ 2020-03-31

*Bild 19: Start Anzeige*

Sensortest01 5=Ende  
U=25.1 R=25.5 Z=24.3  
1/2=M0 3/4=M1 Auf/Zu  
Imot=0.0

*Bild 20: Erste Anzeige nach dem Start;  
V)orlauf- R)ücklauf- Z)immertemperatur in  
°C; Taste1: Motor0 Auf ... Taste4: Motor1 Zu;  
Taste 5 Neu starten*

Sensortest01 5=Ende  
U=25.2 R=25.2 Z=24.4  
1/2=M0 3/4=M1 Auf/Zu  
Mot0 AUF Imot=15.30

*Bild 22: Taste1 bei angeschlossenem Motor  
gedrückt: Motor0 Auf, Strom ist 15.3mA  
(typischer Wert, kann ca. 10 bis 40mA sein)*

Sensortest01 5=Ende  
U=25.2 R=25.0 Z=24.9  
1/2=M0 3/4=M1 Auf/Zu  
Mot0 AUF Imot=98.6

*Bild 21: Taste2 bei angeschlossenem Motor  
gedrückt: Motor0 Zu auf Anschlag, Strom ist  
98.6mA (typischer Wert, kann zwischen 80  
und 120mA sein)*

Sensor test 01 5=Ende  
U=25.1 R=24.6 Z=24.4  
1/2=M0 3/4=M1 Auf/Zu  
! MOTOR KURZSCHLUSS!

*Bild 23: Bei einer Inbetriebnahme kann ein Kurzschluss vorkommen. Der Motor wird sofort ausgeschaltet und für 2sec dieser Zustand angezeigt: Taste sofort auslassen, Motor abstrecken und Fehler suchen!!!*

INITIATING  
RELOAD  
OF  
HZRR200

Bild 24: Nach Drücken der Taste 5 wird kurz dies angezeigt und das Testgerät startet neu.

## 2.5 Aufbau und Inbetriebnahme

## 2.5.1 Software zur Inbetriebnahme aufspielen

Die Software befindet sich in obigem Verzeichnis (iMac3 von pl), kann aber auf einem beliebigen Computer mit Linux/OSX/Win mit installierter Arduino IDE (Vers >= 1.8) mit den nötigen Zusatzmodulen geladen und auf dem Arduino Nano installiert werden.

pl/Documents/\_/\_projekte/hzrr200/HZRR\_200/Software/arduino/hzrr-200/hzrr200\_10\_Inbetrieb/hzrr200\_10\_Inbetrieb.ino

**ACHTUNG:** Das Programm besteht aus mehreren Dateien. Sie alle werden in der Leiste über dem Text-Editor angezeigt. **DIESE DATEIEN NICHT VERÄNDERN** sonst funktioniert das Test-Programm nicht mehr und muss wieder repariert oder von einer Sicherung geholt werden.

**WICHTIG: Zum Hochladen des Programms ist es unwichtig, welche der Dateien angewählt wurde, Drücken des „Hochladen“ Pfeils lädt immer das gesamte Programm.**



Bild 25: Arduino Nano an USB-B-mini Stecker zum Programmieren der Inbetriebnahmesoftware

```
hzrr200_10_Inbetrieb - Hzrr200brd.cpp | Arduino 1.8.10
hzrr200_10_Inbetrieb Hzrr200app.cpp Hzrr200app.h Hzrr200brd.cpp Hzrr200brd.h hzrr200doc2.h

651 // ****
652 // Inbetriebnahme Test
653 // ****
654
655 void Hzrr200brd::test_inbetrieb( byte wahl ) {
656   byte but=0, butOld=0;
657   char s[41];
658   int i;
659   float f;
660
661   // *** Version
662   Serial.print(wahl);
663   Serial.print(" ");
664   Serial.print(FIRMWARE_NAME);
665   Serial.print(" ");
666   Serial.print(FIRMWARE_VERSION);
667   Serial.print(" ");
668   Serial.print(FIRMWARE_DATE);
669   Serial.print(" ");
670   Serial.print(HARDWARE_VERSION);
671   Serial.print(" ");
672   Serial.print(HARDWARE_DATE);
673   Serial.println();
674
675
676
677

Hochladen abgeschlossen.

avrduude done. Thank you.

780
```

Bild 26: Arduino IDE mit geladenem Programm zur Inbetriebnahme ('Sketch')

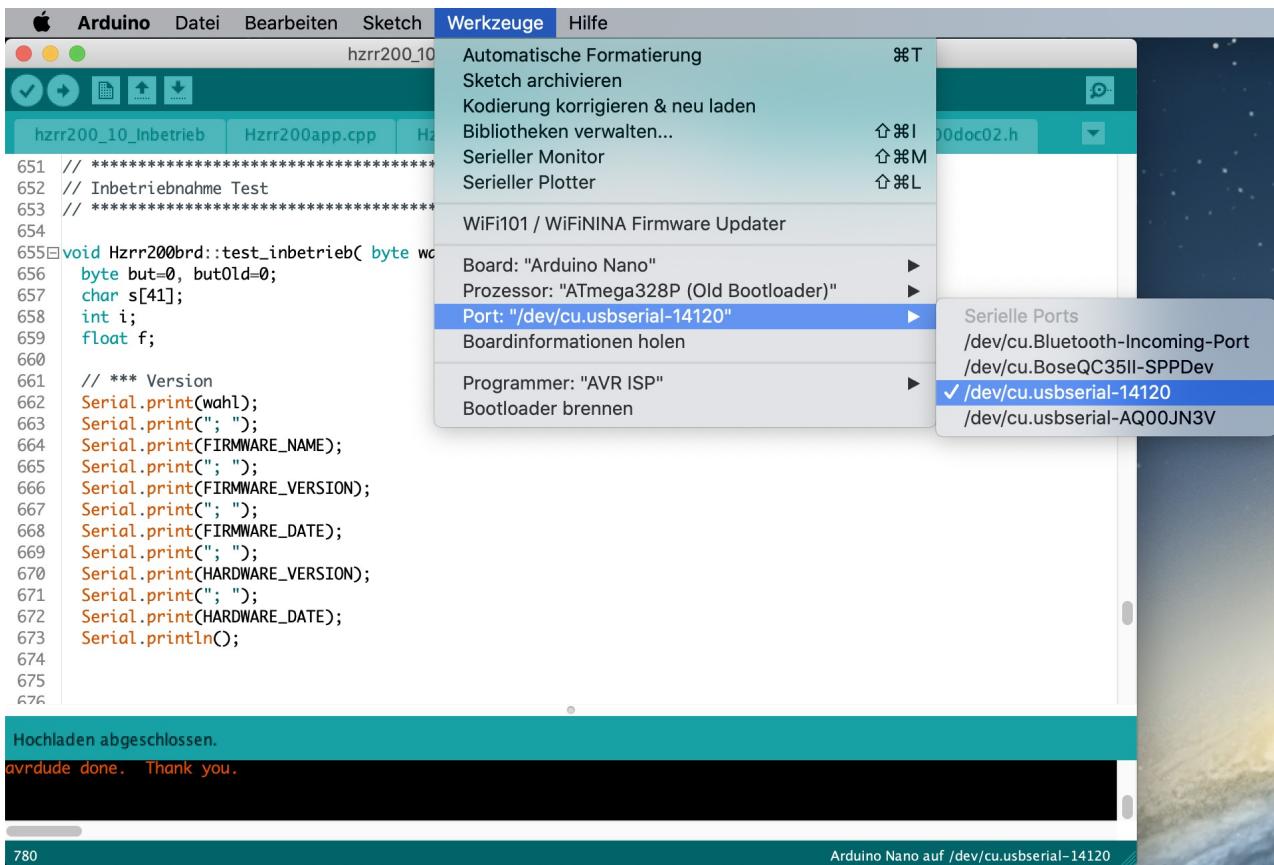


Bild 27: Einstellungen an der Arduino IDE

Unter Werkzeuge wird eingestellt:

- Board: Arduino Nano
- Prozessor: ATmega328P (Old Bootloader)
- Port: Häckchen neben der passenden seriellen USB Schnittstelle.  
Hinweis: der lässt sich finden indem man den Arduino absteckt, in obiges Menü geht und sich die Port-Liste merkt. Dann wieder anstecken und nach etwa 10 sec die Portliste neu öffnen und anzeigen lassen. An den neuen Port ein Häckchen setzen.
- Ganz unten rechts wird zur schnellen Orientierung eine Kurzform der Einstellungen angezeigt,  
Oben in der Fensterüberschrift wird der Name des Programms und die Version der Arduino IDE angezeigt.
- Drücken der runden Schaltfläche oben links mit dem Pfeil nach rechts startet das Programmieren des Arduino Nano. Dabei wird das Programm ggf. vorher neu übersetzt.  

- Nach erfolgreichem Hochladen wird unten in der Zeile über dem schwarz hinterlegten Fenster der Text „Hochladen abgeschlossen“ angezeigt.
- Die Programmierung kann auch im Eingebauten Zustand bei fertigem Modul erfolgen.

## 2.5.2 Modul aufbauen

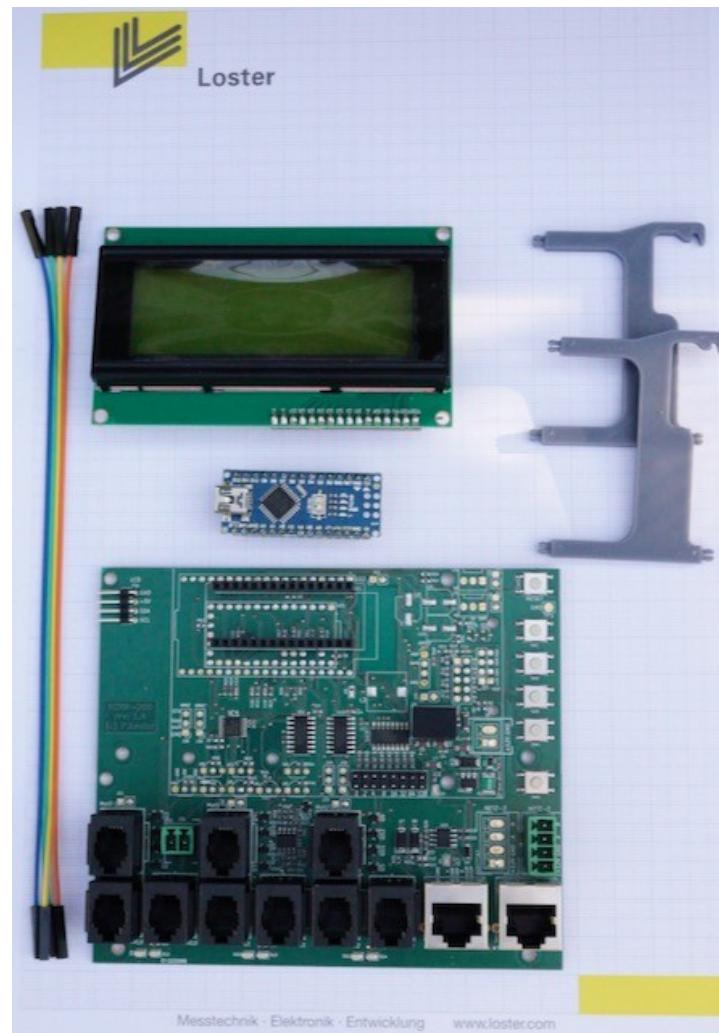


Bild 28: Komponenten zum Aufbau des Moduls

- Oben links: LCD mit montierter I2C Platine auf der Rückseite
- Oben rechts: Zwei Halter für LCD-Anzeige
- Mitte: Arduino Nano, vorzugsweise mit Inbetriebnahme Programm installiert
- Unten: Modul Platine
- linker Rand: 4-poliges Buchse-Buchse Kabel zum Anschluss der LCD-Anzeige an die Modul Platine.

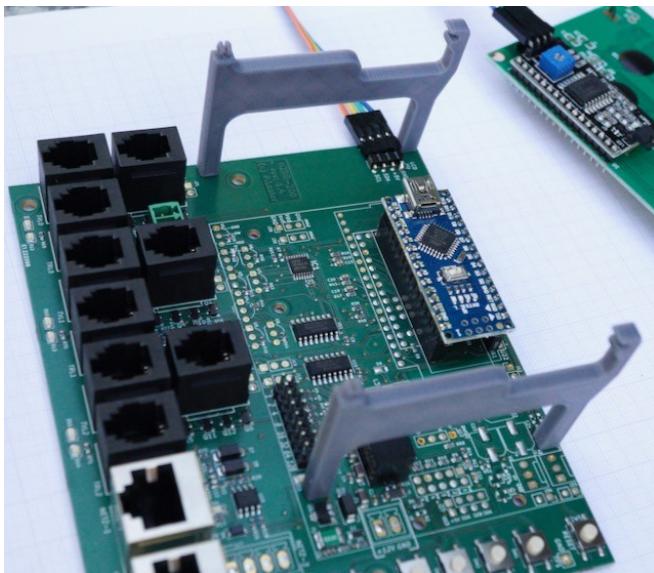


*Bild 29: Arduino Nano auf Modul Platine aufgesteckt; LCD über 4-poligem Kabel verbunden.*



*Bild 30: LCD von der Unterseite*

- Arduino Nano wird wie im Bild aufgesteckt. Richtung beachten! Nicht versetzt aufstecken.
- 4-poliges Kabel wie im Bild an die Stecker anschließen. POLUNG BEACHTEN !!!  
Bemerkung: Die Kabel haben jedesmal andere Farben, da sie von einem 20-poligen Flachkabelbund abgetrennt werden.



*Bild 31: Zwei Abstandshalter in die dafür vorgesehenen Löcher drücken*



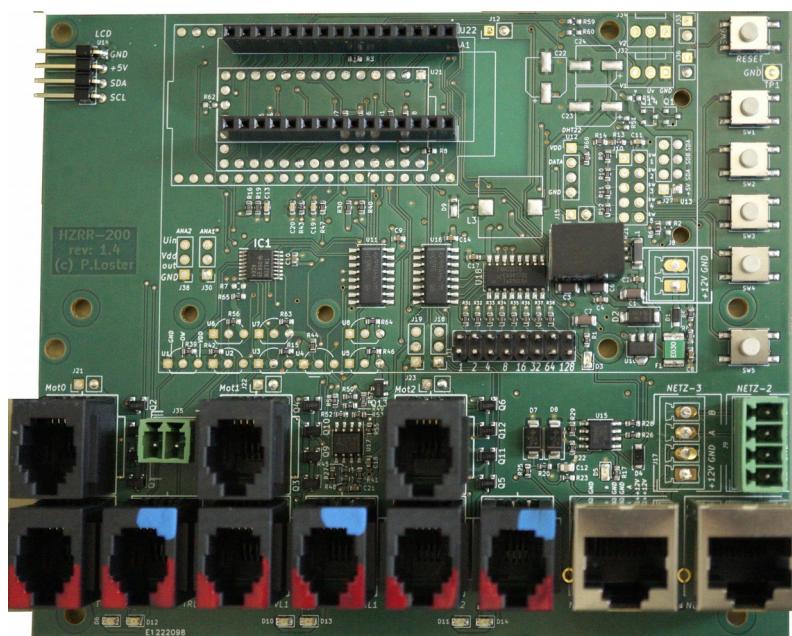
*Bild 32: Montage des LCDs an den Abstandhaltern*



*Bild 33: Kennzeichnen der Temperatur-Sensor Buchsen mit roter Farbe. Roter Nagellack mit etwas zusätzlichem weißem Pigment hat sich bewährt.*

**ACHTUNG: vorsichtig auftragen dass kein Lack in das innere der Buchse fließt !!!**

Rücklauf-Temperatur Buchsen **blau** kennzeichnen:



### 2.5.3 Test Aufbau

Das Modul wird mit 12VDC versorgt, über den selben Stecker wird auch das serielle RS485 Netzwerk angeschlossen. Diese Leitungen sind auch 1:1 mit den RJ45 Netzwerksteckern verbunden.

**ACHTUNG: Es handelt sich NICHT UM EINE ETHERNET (Computer Netzwerk) Verbindung!! auch wenn die Modular 8P8 Buchsen mit Schirm gleich sind.**

Die Module können auch über die RJ45 8P8 verkabelt werden.



Bild 36: USB nach RS485 Konverter  
(A/+:weiß B/-:braun)

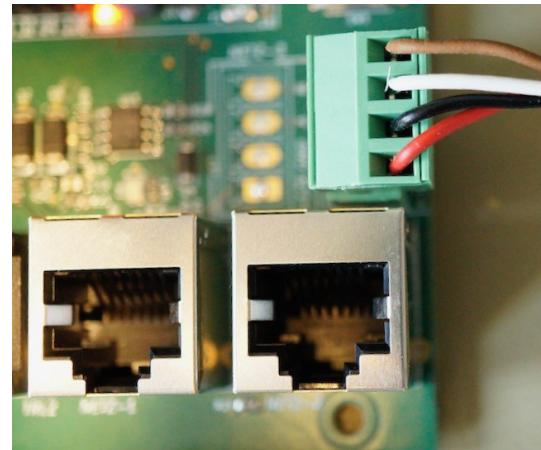


Bild 35: Anschluss 12V Stromversorgung  
(+ :rot - :schwarz) und RS485  
Datenverbindung (A/+:weiß B/-:braun)

Die Netzwerkverbindung wird über einen RJ45 - USB Adapter hergestellt.

Auf dem PC wird zusätzlich zur Arduino IDE ein serielles Terminal geöffnet. Als Parameter werden das USB-Interface, 115200 Baud, 8 Bit, no parity, 2 Stopbits eingestellt.

**WICHTIG: Es sind nun ZWEI USB-Serielle Schnittstellen mit dem Modul verbunden, und es laufen ZWEI verschiedene Serielle Terminal-Programme in jeweils einem eigenen Fenster !!!**

1. Über die USB-Schnittstelle des Arduino Nano, darstellbar im seriellen Terminal der Arduino IDE (Start über die Schaltfläche rechts oben in der Arduino IDE), z.B. in Bild Bild 41 oder Bild 42 und folgende.
2. Über den USB-RS485 Konverter Bild 36 am RS485 Bus; darstellbar über das in Bild Bild 37 und Bild 38 beschriebene Serielle Terminal



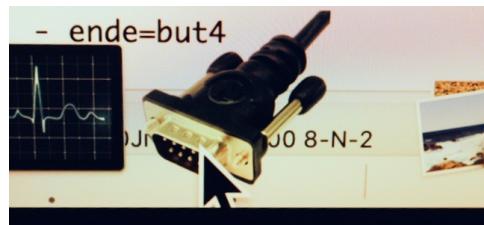


Bild 37: Serielles Terminal öffnen, hier CoolTerm (Mac), alternativ GtTerm (Linux)

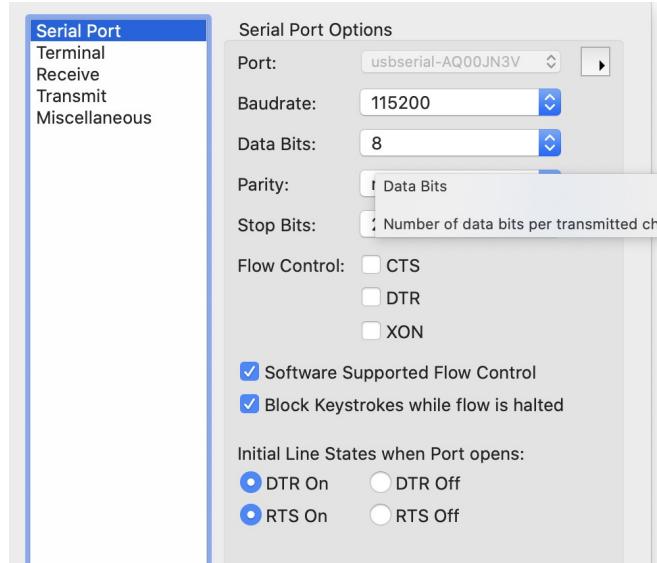


Bild 38: Einstellungen für das serielle Terminal

## 2.5.4 Inbetriebnahme Tests

### 2.5.4.1 LCD-Test

Nach Start des Programms wird ein Startbildschirm angezeigt, DANN nach einigen Sekunden wird der LCD-Test angezeigt:

```
12345678901234567890
12345678901234567890
12345678901234567890
12345678901234567890
```

Bild 39: LCD-Test

#### 2.5.4.2 Taster-Test (Button-Test)



Bild 40: Tasten: But5 Loch But4 But3 But2 But1 --- RESET

Gleichzeitig wird auf dem Arduino Terminal (Start oben rechts in der Arduino IDE) folgendes angezeigt:

Bild 41: Anzeige nach Start des Test-Programms auf dem seriellen Terminal der Arduino IDE

Drückt man die Tasten (Buttons) 1 bis 4 so erscheint die zugehörige Nummer, beim Loslassen wird wieder 0 angezeigt. Nach Drücken der Taste 5 wird zum nächsten Test weitergeschaltet.

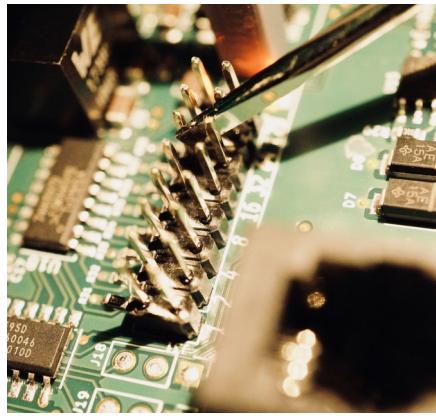
#### 2.5.4.3 Address- und Temperatursensor-Test

Auf dem seriellen Terminal der Arduino IDE erscheint jede Sekunde eine neue Anzeige der Form:

```
Temp4 degC=-127.00; Temp5 degC=-127.00; Temp6 degC=-127.00; Temp7 degC=-127.00dC; TZ=25.7
Moduladresse = 0x20 = 32
Temp0 degC=27.56; Temp1 degC=-127.00; Temp2 degC=-127.00; Temp3 degC=-127.00; ->But4
Temp4 degC=-127.00; Temp5 degC=-127.00; Temp6 degC=-127.00; Temp7 degC=-127.00dC; TZ=25.7
Moduladresse = 0x20 = 32
Temp0 degC=27.56; Temp1 degC=-127.00; Temp2 degC=-127.00; Temp3 degC=-127.00; ->But4
Temp4 degC=-127.00; Temp5 degC=-127.00; Temp6 degC=-127.00; Temp7 degC=-127.00dC; TZ=25.7
```

Bild 42: Ausgabe auf dem Seriellem Terminal der Arduino IDE

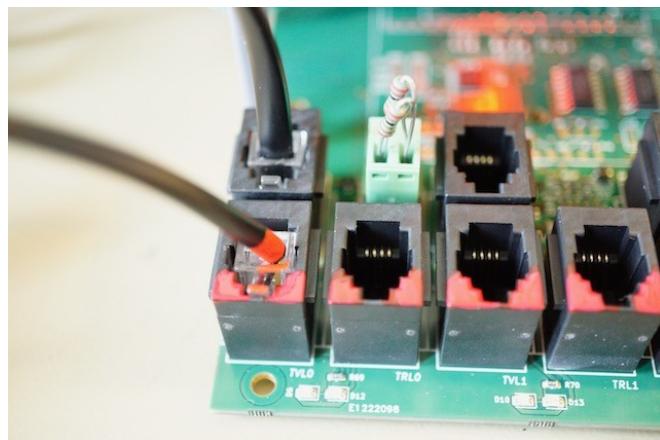
Der Reihe nach werden die gegenüberliegenden Pins der Adress-Stifteiste gebrückt, z.B. mit einem blanken Schraubenzieher. **Pins nicht verbiegen!**



*Bild 43: Überbrücken der Adress-Stifte mit einem blanken, kleinen Schraubendreher*

Auf der Ausgabe des Seriellen Monitors der Arduino IDE kann dann die vom Modul erkannte Adresse abgelesen werden: Je nachdem welche Stifte gebrückt wurden:

„Moduladresse = 1“ oder 2, 4, 8, 16, 32, 64 oder 128.



*Bild 44: Motor 0, Temperatur 0 und Zimmertemperatur Simulation eingesteckt*

Als nächstes wird in jede der 6 Modular 4P4 Buchsen (auch RJ10 bezeichnet) ein Temperatursensor gesteckt. Es genügt ein Sensor der jewils in eine Buchse gesteckt wird. Auf dem Seriellen Monitor der Arduino IDE wird dann die Temperatur angezeigt, in etwa die Umgebungstemperatur des Sensors, beginnend mit Temp0 bis Temp5. Temperatur 6 und 7 haben keine Stecker.



*Bild 45: Raum-Thermometer  
Simulation mit 1100 Ohm Widerstand*

Zuletzt wird der Eingang der Raumtemperatur geprüft.

Anstecken des Raumthermostaten von Technische Alternative oder als Ersatz einen Widerstand von 1100 Ohm ermöglicht das Prüfen der Anzeige „Tz=25.7“ - gemeint ist auch hier °C .

Mit der Taste4 (But4) geht es zur nächsten Anzeige:

#### **2.5.4.4      Motor-Test**

```
19:42:38.010 -> ; Temp4 degC=-127.00; Temp5 degC=-127.00; Temp6 degC=-127.00; Temp7 degC=-127.00dC; TZ=25.7
19:42:39.086 -> ; Moduladresse = 0x0 = 0
19:42:39.194 -> ; Temp0 degC=26.37; Temp1 degC=-127.00; Temp2 degC=-127.00; Temp3 degC=-127.00; ->But4
19:42:39.194 -> ; Temp4 degC=-127.00; Temp5 degC=-127.00; Temp6 degC=-127.00; Temp7 degC=-127.00dC; TZ=25.7
19:42:40.274 -> ; Moduladresse = 0x0 = 0
19:42:40.379 -> ; Temp0 degC=26.37; Temp1 degC=-127.00; Temp2 degC=-127.00; Temp3 degC=-127.00; ->But4
19:42:40.379 -> ; Temp4 degC=-127.00; Temp5 degC=-127.00; Temp6 degC=-127.00; Temp7 degC=-127.00dC; TZ=25.7
19:42:41.456 -> Motor Test
19:42:41.456 -> Taste 1:nächster; 2:vorh.Motor /// 3:Auf; 4:Zu; Taste 5 Ende
19:42:41.456 -> Motor0 Zu
19:42:41.456 -> iMot=2.09mA;
19:42:42.465 -> iMot=9.75mA;
```

*Bild 46: Taste4 (But4) wechselt zum Motor-Test*

```
19:38:53.191 -> iMot=1.05mA;
19:38:54.192 -> iMot=0.70mA;
19:38:55.203 -> iMot=0.00mA;
19:38:56.202 -> iMot=0.00mA;
19:38:57.206 -> iMot=0.00mA;
19:38:58.217 -> iMot=0.00mA;
19:38:59.220 -> iMot=0.00mA;
19:39:00.192 -> iMot=0.00mA;
19:39:00.407 -> Motor0
19:39:01.202 -> iMot=0.00mA;
19:39:02.207 -> iMot=0.00mA;
19:39:03.211 -> Motor0 Auf
19:39:03.211 -> iMot=2.09mA;
19:39:04.220 -> iMot=12.54mA;
19:39:05.216 -> iMot=14.98mA;
```

*Bild 47: Ausgabe beim Motortest*

- Taste1: nächster Motor Nr. 0, 1, 2
- Taste2: vorheriger Motor
- Taste3: Ventil auf
- Taste4: Ventil zu

- Taste5: Ende

Mit den Tasten kann der Motor gewählt und in beide Richtungen bewegt werden. Der Test ist mit einem Motor in allen drei Motor-Ausgängen durchzuführen.

Mit Taste 5 kommt man zum nächsten Test:

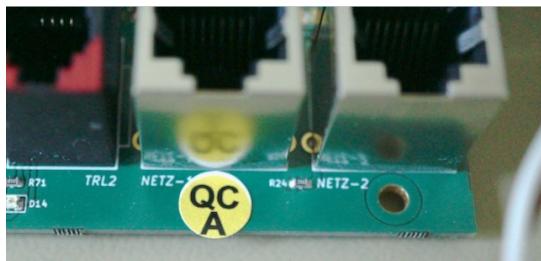
## 2.5.4.5 Kommunikationstest

Bild 48 zeigt oben die Ausgabe der Seriellen Terminals der Arduino IDE (STA1) und unten jenes das mit dem RS486-USB Wandler (STUW) verbunden ist.

- alle paar Sekunden schreibt das Regler Modul auf das STAI  
**'tx: '**  
und sendet dann den Text  
**'hello! - ende=but4'**  
über die serielle Schnittstelle und aktiviertem RS485  
Sender-Treiber  
and den angeschlossenen RS495 - USB Adapter.
  - Auf dem STAI erscheint der komplette Text.
  - Auf dem zweiten STUW erscheint nur der Text  
**'hello! - ende=but4'**
  - Schreiben von Zeichen auf dem STUW unten wird  
dort nicht angezeigt, aber die Zeichen werden über  
RS485 an das STAI gesendet und dort jeweils als z.B.  
**'rx=a'**  
angezeigt.

## 2.5.5 Abschluss Inbetriebnahme

Aufkleber zur Kennzeichnung der Inbetriebnahme anbringen.



*Bild 49: Kennzeichen der erfolgreichen Inbetriebnahme*

Bild 48: serial Terminals

## 2.6 Anwendungsprogramm Installieren und Prüfen

TODO