

Drahtlose Sensordatenerfassung und -verarbeitung mit Linux

Axel Wachtler, Jörg Wunsch, Matthias Vorwerk

4. April 2011

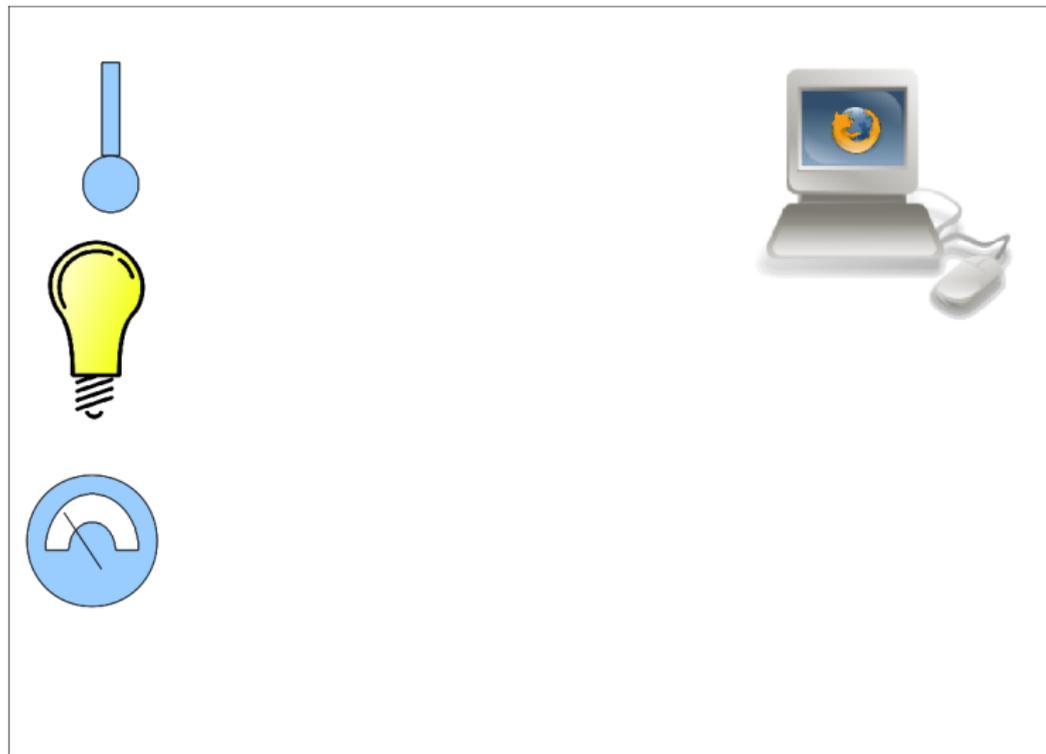
Motivation

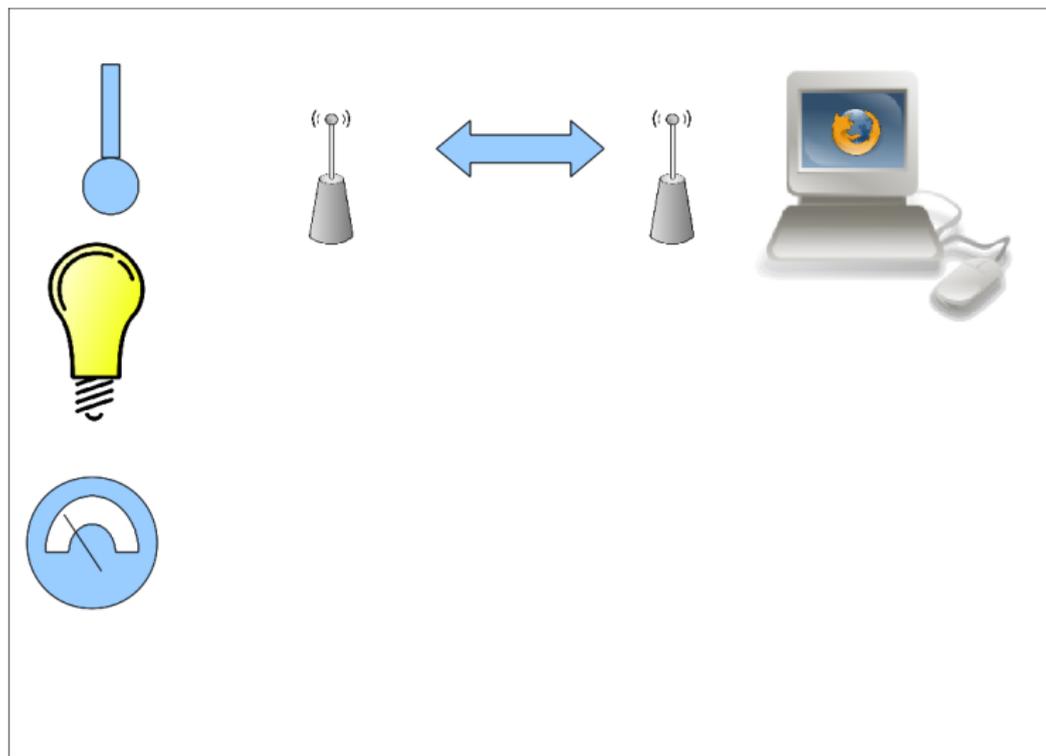
- Umweltgrößen (Temperatur, Luftfeuchte, Druck und Licht) **beeinflussen technische Prozesse und Vorgänge als Stellgrößen**
- Durch die dezentrale Erfassung an (vielen) unterschiedlichen Orten kann der ortsabhängige Verlauf von Parametern bestimmt werden.
- Die **drahtgebundene** Verteilung von Sensoren verursacht einen erhöhten Infrastruktur-Aufwand.
- Die Datenrate einzelner Sensoren ist gering. Messwerte fallen in der Regel im Sekunden- oder Minutentakt an.

- Übersicht Funkmodule

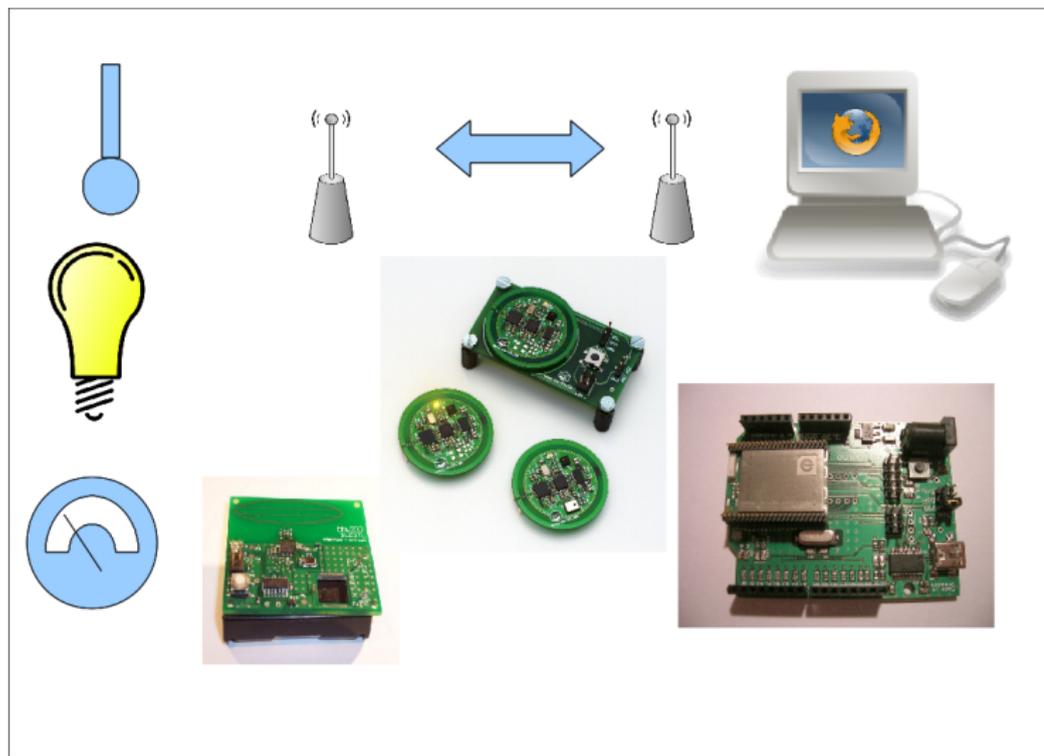
- Übersicht Funkmodule
- Vergleich verschiedener Sensoren

- Übersicht Funkmodule
- Vergleich verschiedener Sensoren
- Verarbeitung der Sensordaten





Drahtlose Sensornetzwerke



- IEEE 802.11.x (WiFi)

- **IEEE 802.11.x** (WiFi)
 - 54 ... 300 Mbit/s
 - Netzwerkinterface, FileTransfer, Audio, Video

- **IEEE 802.11.x** (WiFi)
 - 54 ... 300 Mbit/s
 - Netzwerkinterface, FileTransfer, Audio, Video
- **IEEE 802.15.1** (Bluetooth)

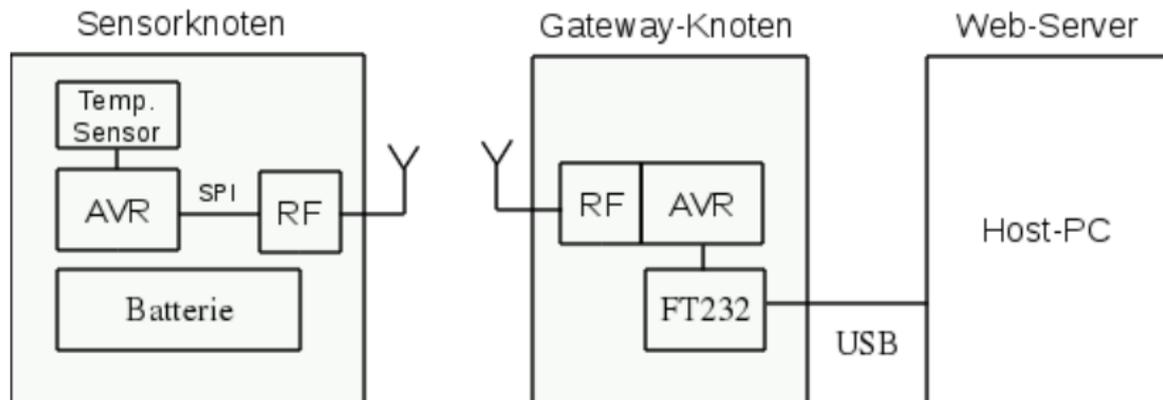
- **IEEE 802.11.x** (WiFi)
 - 54 ... 300 Mbit/s
 - Netzwerkinterface, FileTransfer, Audio, Video
- **IEEE 802.15.1** (Bluetooth)
 - 732,2 kbit/s ... 2,1 MBit/s
 - Nahbereichskommunikation, Sprachübertragung

- **IEEE 802.11.x** (WiFi)
 - 54 ... 300 Mbit/s
 - Netzwerkinterface, FileTransfer, Audio, Video
- **IEEE 802.15.1** (Bluetooth)
 - 732,2 kbit/s ... 2,1 MBit/s
 - Nahbereichskommunikation, Sprachübertragung
- **IEEE 802.15.4** (ZigBee)

- **IEEE 802.11.x** (WiFi)
 - 54 ... 300 Mbit/s
 - Netzwerkinterface, FileTransfer, Audio, Video
- **IEEE 802.15.1** (Bluetooth)
 - 732,2 kbit/s ... 2,1 MBit/s
 - Nahbereichskommunikation, Sprachübertragung
- **IEEE 802.15.4** (ZigBee)
 - 20 bis 250 kbit/s
 - hauptsächlich für Sensoranwendungen geeignet:

- **IEEE 802.11.x** (WiFi)
 - 54 ... 300 Mbit/s
 - Netzwerkinterface, FileTransfer, Audio, Video
- **IEEE 802.15.1** (Bluetooth)
 - 732,2 kbit/s ... 2,1 MBit/s
 - Nahbereichskommunikation, Sprachübertragung
- **IEEE 802.15.4** (ZigBee)
 - 20 bis 250 kbit/s
 - hauptsächlich für Sensoranwendungen geeignet:
 - geringe Datenrate
 - energiesparend
 - niedrige Systemkosten

Drahtloses Datenlogger System



• Sensorknoten

- autark, (Batterie, Energy-Harvesting)
- sendet kontinuierlich die Werte der Sensoren zum Gateway

• Gateway

- Ausgabe der Sensordaten über serielles Interface
- permanente Stromversorgung (Netzteil, USB)

• Webserver

- liest und speichert die Daten vom Gatewayknoten
- Auswertung der Daten mit speziellen Anwendungen.

- Es gibt eine Vielzahl an Funkmodulen von unterschiedliche Boardherstellern.
- Die *μ*racoli-Library unterstützt derzeit über 30 Boards und Module
 - A.N. Solutions
 - Atmel
 - Colorado Micro Devices
 - Dietzsch und Thiele Partnerschaftsgesellschaft
 - dresden elektronik ingenieurtechnik gmbh
 - In-Circuit GmbH
 - Logos Electromechanical LLC
 - Meshnetics

- **tiny230** - DIY Projekt von **Jörg Wunsch**
 - ATtiny44/84 mit AT86RF230/231,
 - Sensoren: KTY13, LED als Lichtsensor
- **RadioFaro*** - DIY Projekt von **Daniel Thiele**
 - ATmega128RFA1 Module von Dresden Elektronik
 - Arduino Formfaktor
- **Muse231** - Prototyp, **Ingenieurbüro Dietzsch und Thiele**
 - ATmega88PA + AT86RF231,
 - Sensoren SHT21, MMA7455, Mikrofon, Lichtsensor-LED
- **ZigBit Adapter Boards**
 - ATmega1281 + AT86RF230
 - Sensoren über Breakoutboard angeschlossen



Hardware

- 1 x Gateway Knoten mit USB/RS232, 1...N x Sensorknoten
- 1 x Programmer, optional Hardware-Debugger
 - AVR Dragon → ISP Programmer + JTAG Debugger
 - AVR ISP, USBprog, PonyProg → ISP Programmer
 - JTAG ICE mkII → JTAG Debugger

Software

- avr-gcc + binutils → Compiler und Linker
- avr-libc → Standard C Library für AVR
- avrdude → Programmier Tool
- avr-gdb + avarice → Debugger und JTAG Bridge
- μ racoli → AVR Radio Library

Library zur einfachen Inbetriebnahme von
IEEE-802.15.4-Transceiver-Knoten.



- URL: <http://www.uracoli.de>

Library zur einfachen Inbetriebnahme von IEEE-802.15.4-Transceiver-Knoten.



- URL: <http://www.uracoli.de>
- derzeit über 40 Plattformen (AVR + Atmel-Transceiver) unterstützt

Library zur einfachen Inbetriebnahme von IEEE-802.15.4-Transceiver-Knoten.



- URL: <http://www.uracoli.de>
- derzeit über 40 Plattformen (AVR + Atmel-Transceiver) unterstützt
- High und Low Level Transceiver-Funktionen

Library zur einfachen Inbetriebnahme von IEEE-802.15.4-Transceiver-Knoten.



- URL: <http://www.uracoli.de>
- derzeit über 40 Plattformen (AVR + Atmel-Transceiver) unterstützt
- High und Low Level Transceiver-Funktionen
- Timer, Sensor und IO-Funktionen sind optional

Library zur einfachen Inbetriebnahme von IEEE-802.15.4-Transceiver-Knoten.



- URL: <http://www.uracoli.de>
- derzeit über 40 Plattformen (AVR + Atmel-Transceiver) unterstützt
- High und Low Level Transceiver-Funktionen
- Timer, Sensor und IO-Funktionen sind optional
- viele Beispiele, Anwendungen: Wireless UART, Sniffer, Arduino Support

Library zur einfachen Inbetriebnahme von IEEE-802.15.4-Transceiver-Knoten.



- URL: <http://www.uracoli.de>
- derzeit über 40 Plattformen (AVR + Atmel-Transceiver) unterstützt
- High und Low Level Transceiver-Funktionen
- Timer, Sensor und IO-Funktionen sind optional
- viele Beispiele, Anwendungen: Wireless UART, Sniffer, Arduino Support
- modified BSD Lizenz

Auswahl von Temperatursensoren

Auswahl von Temperatursensoren

- große Vielfalt an analogen und digitalen Sensoren

Auswahl von Temperatursensoren

- große Vielfalt an analogen und digitalen Sensoren
- Vergleich von:
 - LM75 (National Semiconductor, I²C)
 - DS18B20 (Maxim/Dallas, "OneWire")
 - SHT-21 (Sensirion (CH), I²C); Temperatur und rel. Luftfeuchte
 - TMP141 (TI/Burr Brown, "SensorPath")
 - KTY13-6 (Infineon)
 - ATmega128RFA1 (Atmel), interner Sensor

Auswahl von Temperatursensoren

- große Vielfalt an analogen und digitalen Sensoren
- Vergleich von:
 - LM75 (National Semiconductor, I²C)
 - DS18B20 (Maxim/Dallas, "OneWire")
 - SHT-21 (Sensirion (CH), I²C); Temperatur und rel. Luftfeuchte
 - TMP141 (TI/Burr Brown, "SensorPath")
 - KTY13-6 (Infineon)
 - ATmega128RFA1 (Atmel), interner Sensor
- Preis zwischen EUR 0,50 und EUR 20

Hardware

Hardware

- Montage der Sensoren auf Board
- SHT-21 2x, TMP141 3x, alle anderen 1x
- Abfrage über Zigbit-Funkmodul
- Weiterleitung der Daten über Funk
- Batteriebetrieb mit 2 x LR03



Hardware

- Montage der Sensoren auf Board
- SHT-21 2x, TMP141 3x, alle anderen 1x
- Abfrage über Zigbit-Funkmodul
- Weiterleitung der Daten über Funk
- Batteriebetrieb mit 2 x LR03



Umgebung

Hardware

- Montage der Sensoren auf Board
- SHT-21 2x, TMP141 3x, alle anderen 1x
- Abfrage über Zigbit-Funkmodul
- Weiterleitung der Daten über Funk
- Batteriebetrieb mit 2 x LR03

Umgebung

- Langzeittests in ruhiger Umgebung



Hardware

- Montage der Sensoren auf Board
- SHT-21 2x, TMP141 3x, alle anderen 1x
- Abfrage über Zigbit-Funkmodul
- Weiterleitung der Daten über Funk
- Batteriebetrieb mit 2 x LR03

Umgebung

- Langzeittests in ruhiger Umgebung
- Vergleich mit Laborthermometer



Hardware

- Montage der Sensoren auf Board
- SHT-21 2x, TMP141 3x, alle anderen 1x
- Abfrage über Zigbit-Funkmodul
- Weiterleitung der Daten über Funk
- Batteriebetrieb mit 2 x LR03

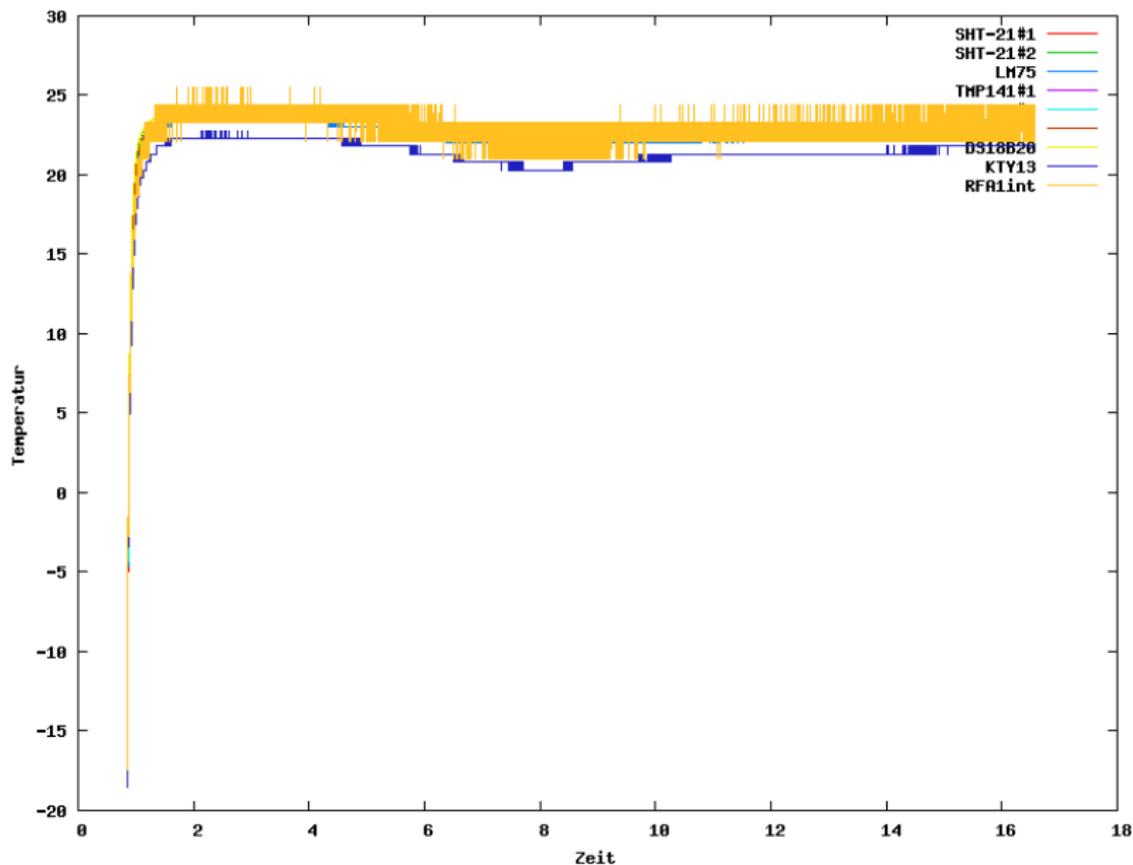


Umgebung

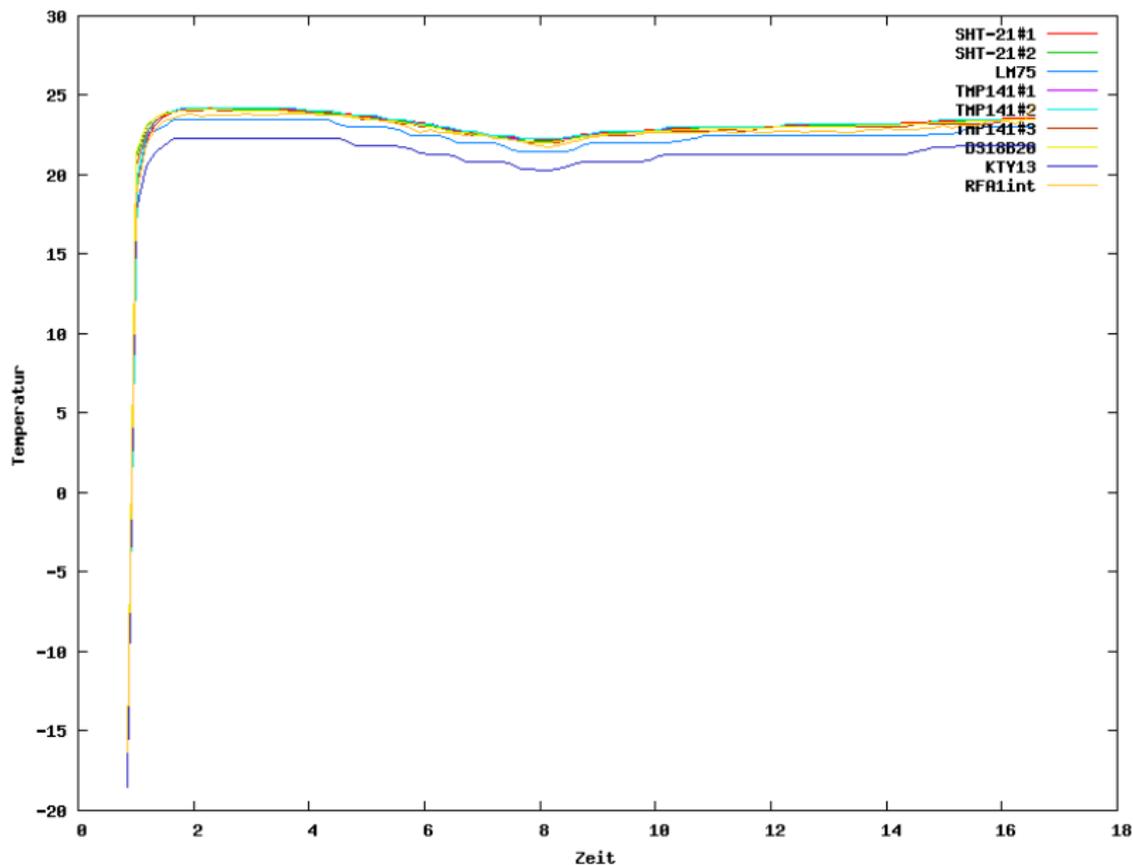
- Langzeittests in ruhiger Umgebung
- Vergleich mit Laborthermometer
- Tests in wechselnder Umgebung, einschließlich Klimaschrank Vötsch VT4004

Messung in ruhiger Umgebung

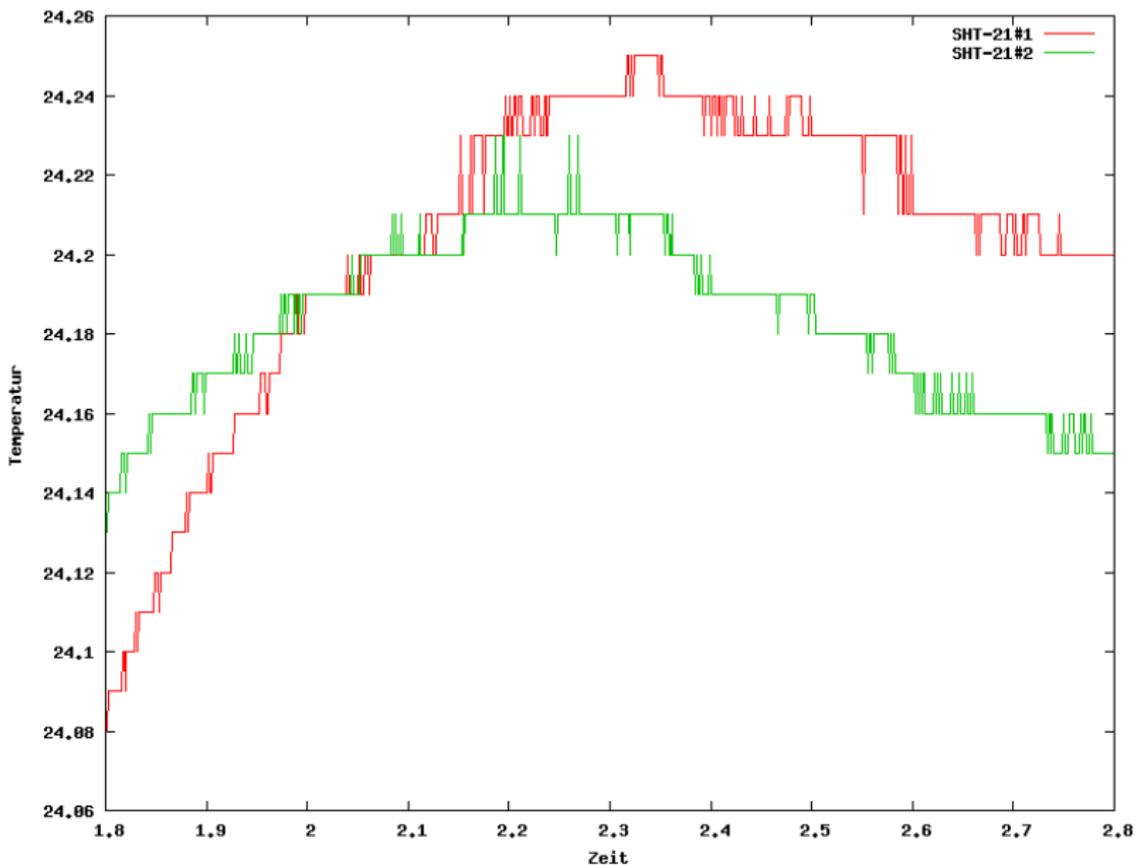
Messung in ruhiger Umgebung



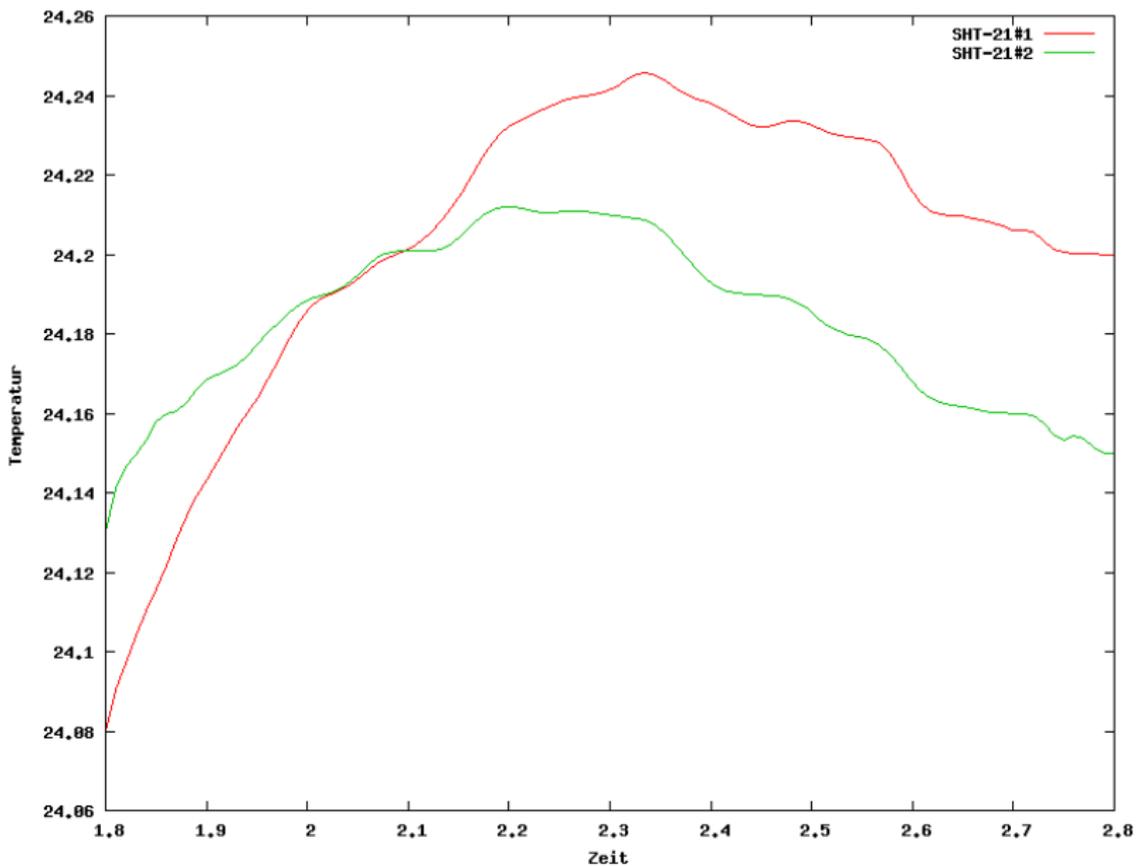
Messung in ruhiger Umgebung



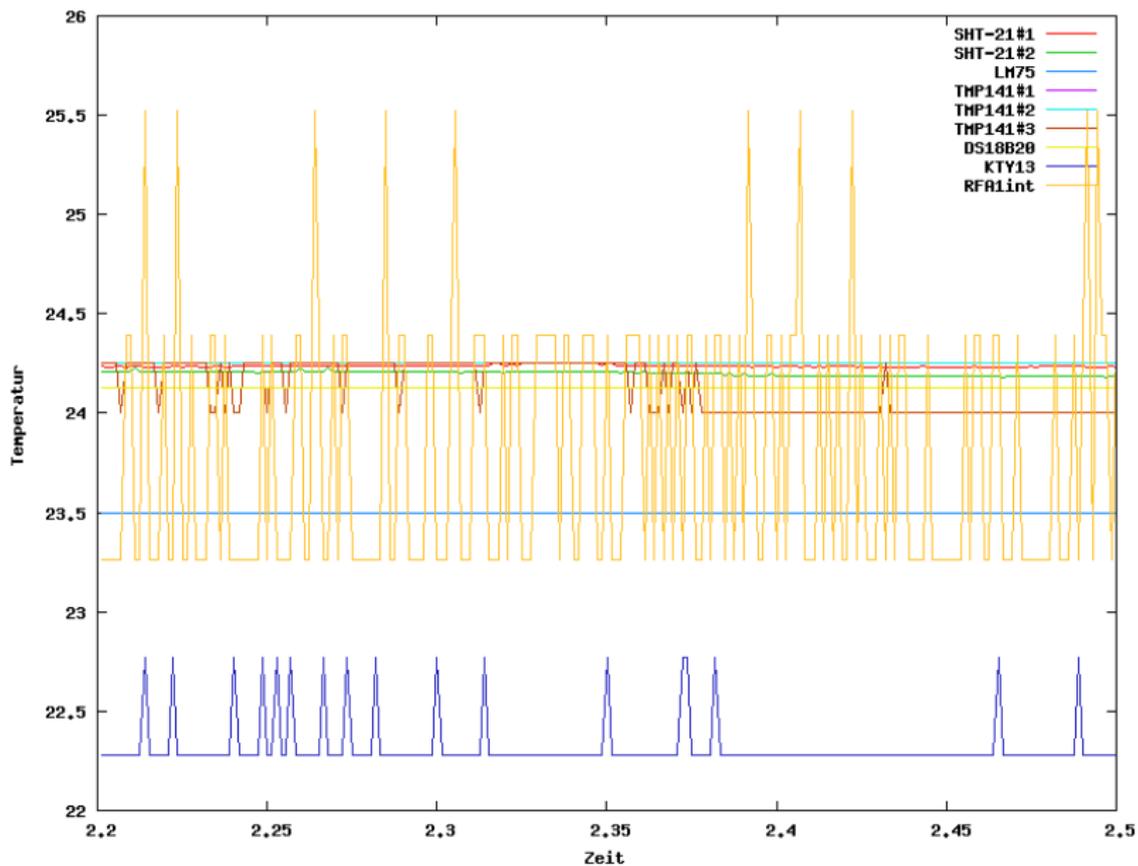
Messung in ruhiger Umgebung



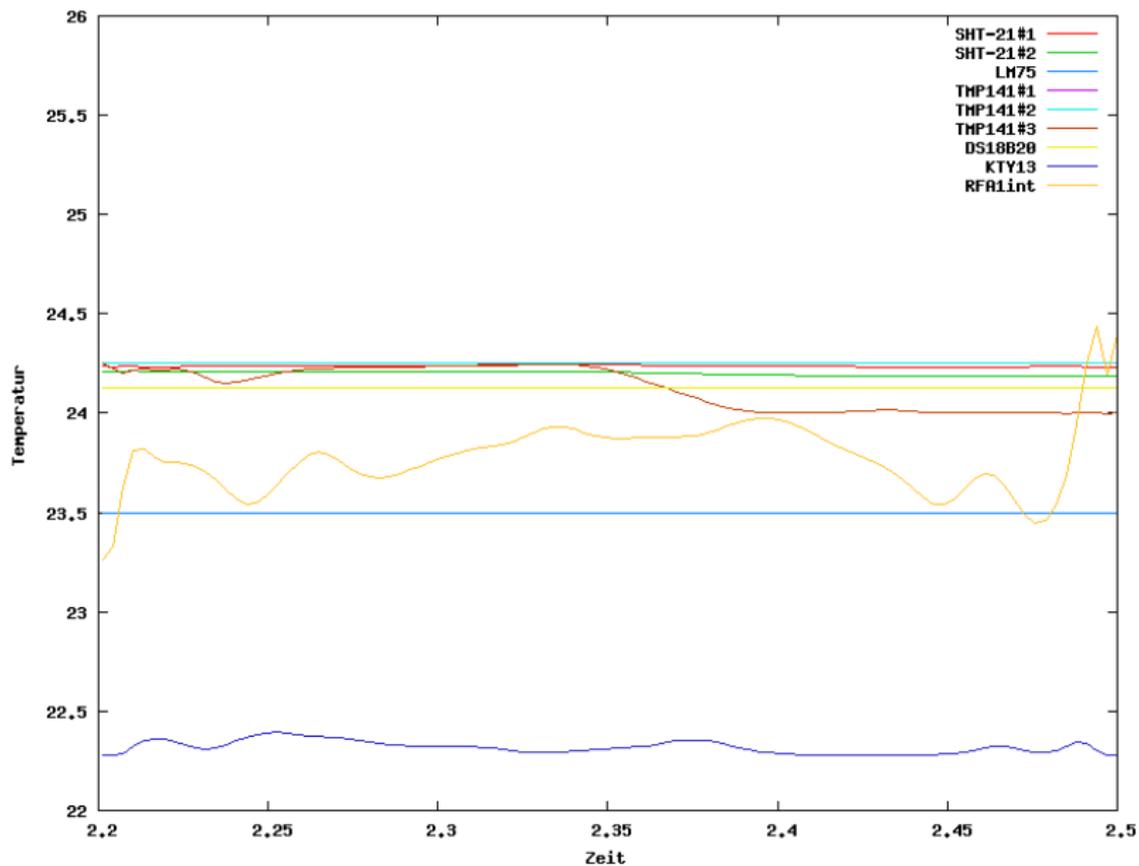
Messung in ruhiger Umgebung



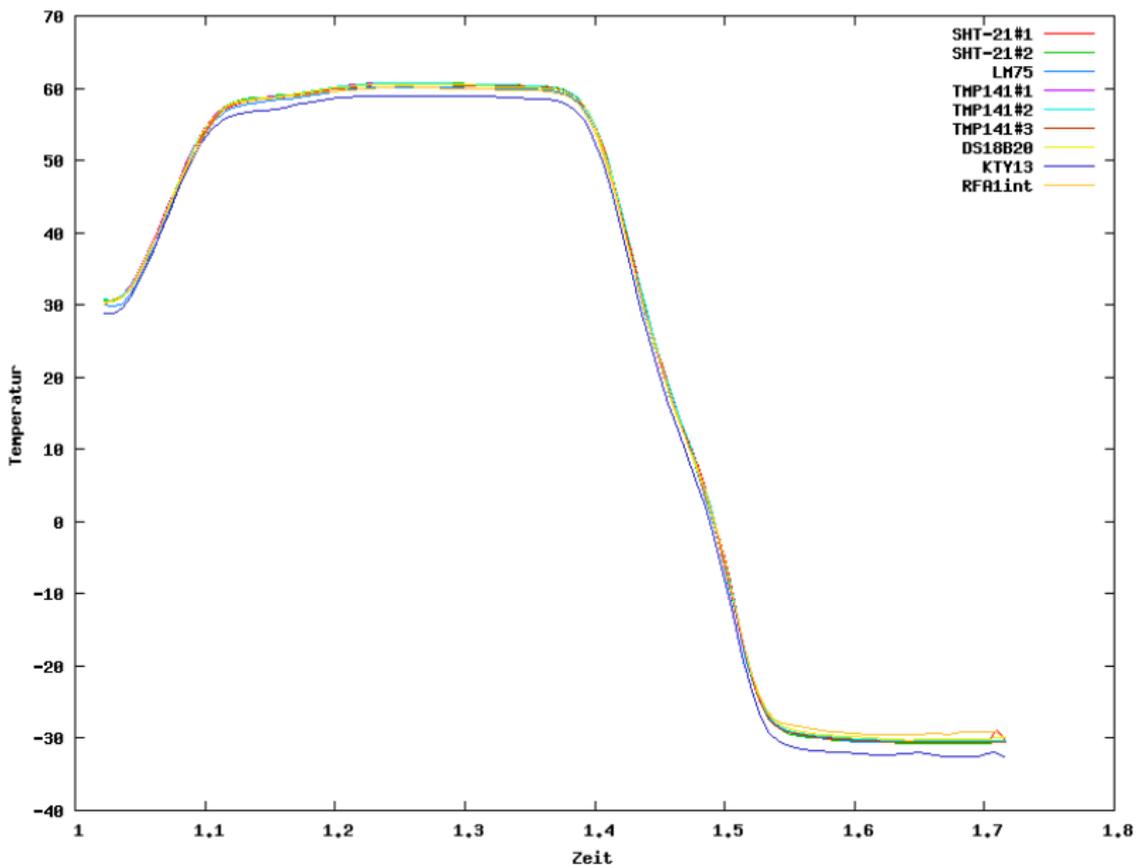
Messung in ruhiger Umgebung



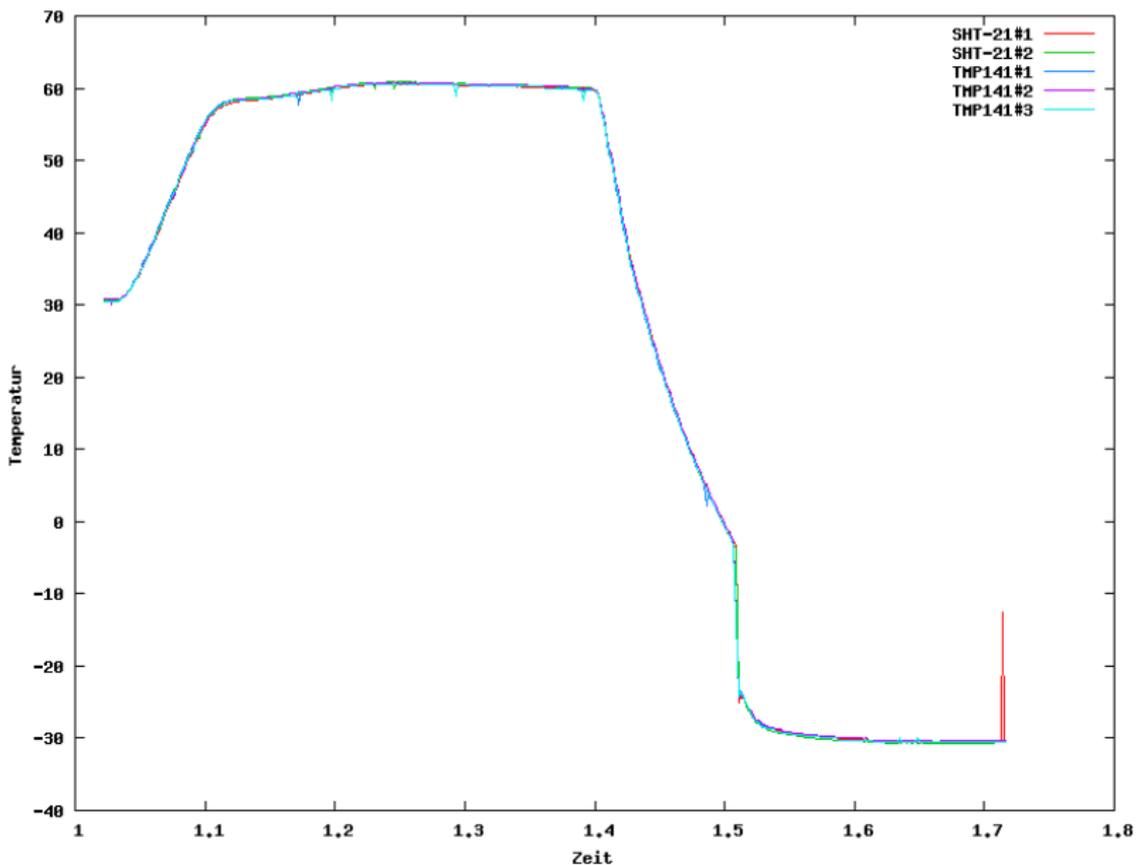
Messung in ruhiger Umgebung



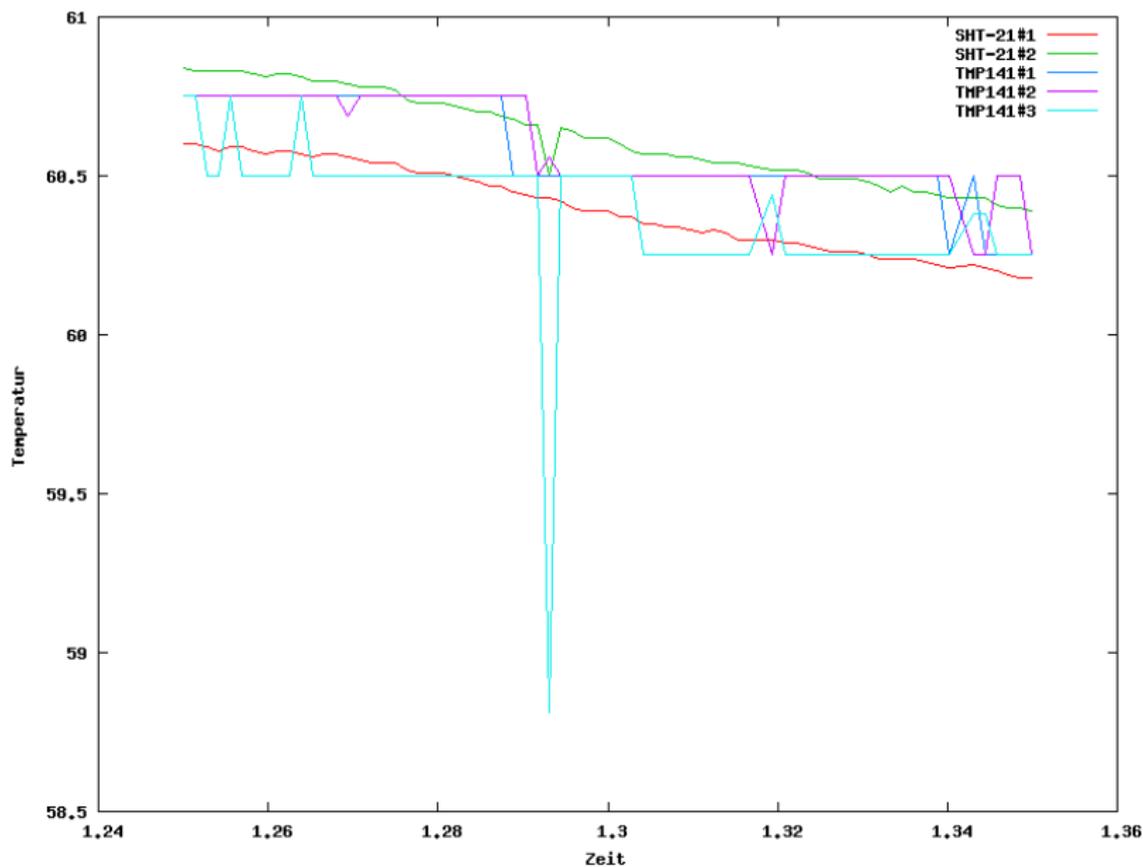
Messung im Klimaschrank



Messung im Klimaschrank



Messung im Klimaschrank



Temperaturmessung

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)
- alle anderen sehr eng gemeinsam gruppiert

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)
- alle anderen sehr eng gemeinsam gruppiert
- ATmega128RFA1: schlechte direkte Auflösung (1,13 K), aber gute Mittelwertbildung; Offset ca. -0,5 K, aber Gradient bei $-30\text{ }^{\circ}\text{C}$

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)
- alle anderen sehr eng gemeinsam gruppiert
- ATmega128RFA1: schlechte direkte Auflösung (1,13 K), aber gute Mittelwertbildung; Offset ca. -0,5 K, aber Gradient bei -30 °C
- SHT-21: Genauigkeit eines guten Laborthermometers, Nutzung als Referenz und für hohe Ansprüche

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)
- alle anderen sehr eng gemeinsam gruppiert
- ATmega128RFA1: schlechte direkte Auflösung (1,13 K), aber gute Mittelwertbildung; Offset ca. -0,5 K, aber Gradient bei -30 °C
- SHT-21: Genauigkeit eines guten Laborthermometers, Nutzung als Referenz und für hohe Ansprüche
- TMP141 haben gutes Preis-/Leistungsverhältnis bei ansprechender Genauigkeit

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)
- alle anderen sehr eng gemeinsam gruppiert
- ATmega128RFA1: schlechte direkte Auflösung (1,13 K), aber gute Mittelwertbildung; Offset ca. -0,5 K, aber Gradient bei -30 °C
- SHT-21: Genauigkeit eines guten Laborthermometers, Nutzung als Referenz und für hohe Ansprüche
- TMP141 haben gutes Preis-/Leistungsverhältnis bei ansprechender Genauigkeit

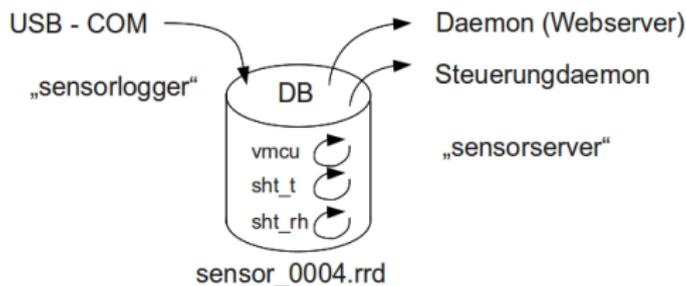
Feuchtigkeitsmessung

Temperaturmessung

- größte Abweichung: unkalibrierter KTY13-6, aber konstant -1,8 K
- LM75: -0,7 K, (ist mit $\pm 3\text{K}$ spezifiziert)
- alle anderen sehr eng gemeinsam gruppiert
- ATmega128RFA1: schlechte direkte Auflösung (1,13 K), aber gute Mittelwertbildung; Offset ca. -0,5 K, aber Gradient bei -30 °C
- SHT-21: Genauigkeit eines guten Laborthermometers, Nutzung als Referenz und für hohe Ansprüche
- TMP141 haben gutes Preis-/Leistungsverhältnis bei ansprechender Genauigkeit

Feuchtigkeitsmessung

- kein Vergleich, nur SHT-21 selbst; gegenseitige Abweichung < 1% RH



- **Speicherung in Round Robin Datenbank (rrdtool Projekt)**

- Aktuelle Messwerte am interessantesten
- pro Sensor (addr) eine Datenbank (.rrd-File)
- je Meßgröße (vmcu, sht_t) eine oder mehrere Round Robin Archive möglich

- **Sensorlogger**

- Gateway → USB → Interface
- Python-Daemon zur Erfassung einlaufender Daten
- Lesen der Daten vom ser. Interface (/dev/ttyUSB0): addr:4, acc: [27, 13, 80], sht_rh: 26902, vmcu: 2744, sht_t: 25800

Mini RRD to HTTP Server

Sensor	Supply /V	Temperature /C	Humidity /%	Accel X /g
0003	2.724	22.048	48.722	0.422
0004	2.870	22.370	45.563	0.281

Sensorserver

- Applikationen zur Auswertung (Visualisierung)
- Zugriff auf RRD-Datenbank-Daten
- Einfacher Mini-HTTP-Server in Python
- Erzeugen von Tabellen mit aktuellen Werten (`tableserver`)
- Grafiken aus RRD-Datenbanken (Visualisierung) (`sensorserver`)

```
import rrdtool
import serial

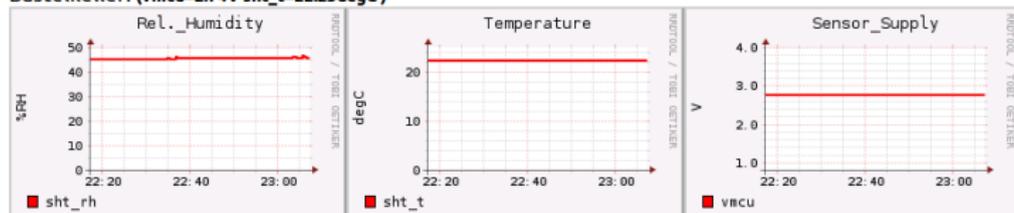
def rrd_create(tstamp, addr, **kwargs):
    # ... prep to create new .rrd file
    rrdtool.create(r.dbname, *args)

def rrd_update(tstamp, addr, **kwargs):
    # ... prep to update .rrd file with new data
    rrdtool.update(r.dbname, arg)
```

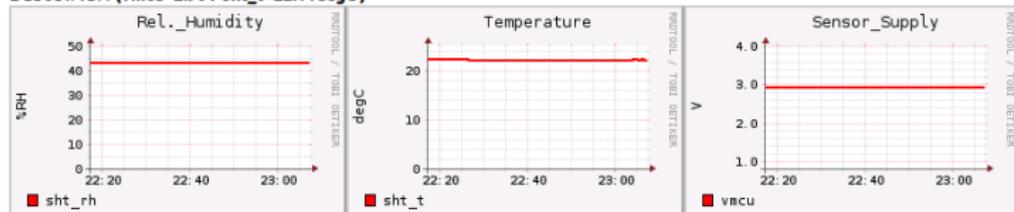
```
sport = serial.Serial(SPORT, BAUD)
sport.open()
while 1:
    try:
        x = sport.readline().strip()
        y = eval("dict(%s)" % x) # well, it's a demo!
        y["tstamp"] = int(time.time())
        if not SENSORS.has_key(y["addr"]):
            rrd_create(**y)
            rrd_update(**y)
    except:
        print "ERROR:", sys.exc_info()
```

Sensorserver (Last update 2011-03-15 23:07:09)

Bastelkeller: (vmcu=2.74V sht_t=22.23degC)



Bastelflur: (vmcu=2.90V sht_t=22.11degC)



```
python sensorlogger.py -p /dev/ttyUSB0 -D  
python sensorserver.py -v  
python tableserver.py -v
```

- Demo: Erfassung, Übertragung und Visualisierung der Sensordaten
- Hinzufügen von Benachrichtigungsfunktionen (SMS, E-Mail)
- Optimierung des Energieverbrauchs
 - Optimierung des Sensor-Datenprotokolls
 - Verlängerung der Knoten-Schlafzustände
- Steuerungsfunktionalität (Bidirektionaler Datenverkehr)
 - Sensorknoten empfangen Daten
 - Bidirektionaler Datenverkehr
- Einbindung in Heimautomatisierungssystem
 - Sensorknoten + Gateway (Modbus Protokoll?)
 - Framework zur Prozessvisualisierung, Steuerungsfunktionalität

Wir möchten uns bei den folgenden Firmen und Personen für die Unterstützung bei diesem Vortrag bedanken.

Herrn Daniel Thiele für die vielen Freizeitstunden bei Entwicklung und Aufbau der RadioFaro-Module.

SENSIRION
THE SENSOR COMPANY

für die großzügige Bemusterung mit digitalen Feuchtesensoren vom Typ SHT21. <http://www.sensirion.com>



für die Bereitstellung der Muse231-Prototypen.
<http://www.ib-dt.de/>

- <http://uracoli.nongnu.org/clt2011/index.html>
 - Quelltexte zur Demo (Knoten, Gateway, Daemons)
 - Links zu weiteren Quellen
- <http://uracoli.nongnu.org>
 - Projektwebseite
- <http://lists.nongnu.org/mailman/listinfo/uracoli-devel>
 - Entwickler Mailingliste
- <http://www.sax.de/~joerg/tiny230>
 - Dokumentation Tiny230 Board
- <http://www.uracoli.de/radiofaro.html>
 - Dokumentation RadioFaro Board
- <http://www.ib-dt.de>
 - Entwicklerseite des Muse231 Prototyps

- 1 Einleitung
- 2 Drahtlose Sensornetzwerke
- 3 Funkstandards
- 4 Drahtloses Datenlogger System
- 5 Boards
- 6 Zutaten für die Firmware-Entwicklung
- 7 *μ*racoli - Die AVR-Radio-Library
- 8 Vergleich von Sensoren
- 9 Versuchsaufbau
- 10 Messungen
- 11 Schlussfolgerungen
- 12 Datenauswertung auf dem PC
- 13 Datenlogger
- 14 Web-Server
- 15 Ausblick
- 16 Danksagung