

Temperaturüberwachung für einen Serverraum

**Verteiltes System zur Datenerfassung,
-speicherung und -auswertung**

Inhaltsverzeichnis

1 Übersicht	5
2 Das Kelvin Board	7
2.1 Funktionsweise	7
2.2 Hardware des Boards	8
2.3 Hochladen einer eigenen Webpage	9
2.4 Belegung der Steckerleiste des PIC-WEB-Boards	9
2.5 Hardware und Anschluss der Temperatur-Sensoren	9
3 Das Programm „kelvin“	11
4 Das Programm „plotter“	13
5 Skripting und crontab	17
5.1 Das Shellsript „errortomail“	17
5.2 Die crontab-Einträge	17

1

Übersicht

Dieses Projekt ermöglicht die Überwachung eines Serverraums mithilfe von bis zu acht Temperatursensoren, deren Daten in eine MySQL-Datenbank abgespeichert werden können. Aus den Messwerten lassen sich übersichtliche Grafiken anfertigen, die den Temperaturverlauf der Sensoren über der Zeit wiedergeben.

Hauptbestandteil ist das so genannte **Kelvin Board**, an das die acht Temperatursensoren angeschlossen werden und das im Serverraum installiert ist. Das Board liest die Temperaturen ein und bietet diese über Netzwerk via TCP/IP in Textform oder als Webpage an. Bild 1.1 zeigt das Schema des Zusammenspiels von Hard- und Software des Systems.

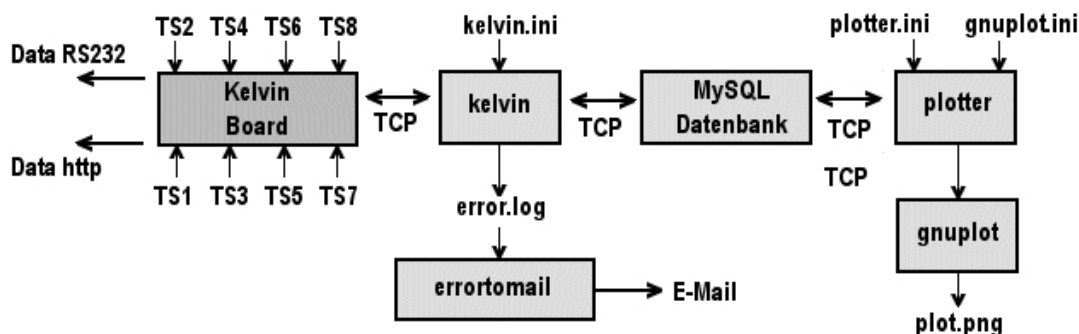


Bild 1.1: Hard- und Software-Übersicht

Das Kommandozeilen-Programm „kelvin“ fragt die Daten periodisch beim *Kelvin Board* ab und speichert diese mit Timestamp in eine MySQL-Datenbank.

Um den Temperaturverlauf in eine Grafik zu überführen kann das ebenfalls Kommandozeilen-Programm „plotter“ verwendet werden, das als Plotterengine das Programm „gnuplot“ verwendet. Das Shellscript „errortomail“ schickt die Fehlerausgabe von „kelvin“ gegebenenfalls an eine E-Mailadresse. Damit können Temperaturüberschreitungen und Fehler rechtzeitig erkannt werden.

Da das Kelvin Board und die Programme *kelvin* sowie *plotter* über TCP/IP kommunizieren, können diese beliebig über mehrere Rechner verteilt sein. Voraussetzung ist lediglich das auf dem Linux-Rechner auf dem *plotter* betrieben wird, das Programm *gnuplot* installiert ist. Informationen zur Installation finden sich unter www.gnuplot.info.

Um die Daten zu speichern wird die MySQL-Datenbank benötigt, die unter www.mysql.de dokumentiert ist. Die Datenbank kann auch auf einem anderen Rechner laufen als die restlichen Programme, aber es muss ein Login, bestehend aus Nutzernamen und Passwort eingerichtet werden, damit sich *kelvin* und *plotter* mit der Datenbank verbinden können.

Mit entsprechenden Einträgen in *crontab* können die Temperaturen periodisch abgefragt werden. Das folgende Listing gibt eine Übersicht wichtiger Dateien und der Verzeichnisstruktur.

Kelvin Board Firmware

```
Code          # Ordner mit dem Sourcecode des Kelvin-Boards
Config        # Ordner mit Konfigurationsdateien
Webpage       # Webpage des Kelvin-Boards
```

Plotter and Kelvin

```
plots         # Ordner, in den die fertigen Plots gespeichert werden
scr           # Ordner mit dem Sourcecode von "kelvin" und "plotter"
temp          # Ordner für temporäre Daten
```

```
kelvin        # Programm zum Speichern der Temperaturdaten in MySQL
kelvin.ini    # Konfigurationsdatei von "kelvin"
error.log     # Fehler-Log-Datei von "kelvin"
errortomail   # Shellscript, das "error.log an eine E-Mail-Adresse schickt
plotter       # Programm zur Erzeugung von Grafiken
plotter.ini   # Konfigurationsdatei von "plotter"
gnuplot.ini   # Zusätzliche Einstellungen von "gnuplot"
```

2

Das Kelvin Board

2.1 Funktionsweise

Die Temperaturwerte der acht Sensoren werden jeweils in kurzen Zeitabständen mit einer Auflösung von 0,1 °C eingelesen und können über eine TCP/IP, RS-232 oder eine Webpage abgerufen werden. Während des Betriebs können neue Sensoren angesteckt oder entfernt werden, die selbstständig erkannt werden. Da die Temperatur digital vom Sensor zum Board übertragen wird, verfälscht die Länge des Kabels den Messwert nicht.

- Anzeige über die Webpage
Um die Daten über die Board-interne Webpage abzufragen, kann die IP-Adresse des Boards in die Adresszeile eines Browsers eingegeben werden. Diese Webseite wird direkt vom Board erzeugt und zeigt die Temperaturwerte in Echtzeit an.
- Datenabfrage über TCP/IP
Das Board lauscht auf einem einstellbaren Port auf Verbindungsanfragen über TCP/IP. Geht eine SYN Paket ein, so werden die Temperaturdaten in folgendem Format zurückgeschickt. Danach wird die Verbindung wieder geschlossen.

Beispiel einer Abfrage über TCP:

```
TS1:  50.2 C
TS2:  83.5 C
TS3:  ---.- C
TS4:  ---.- C
TS5:  23.0 C
TS6:  10.2 C
TS7:  ---.- C
TS8: 123.0 C
da8236889a61139e3da
```

Jede Zeile wird mit Carriage Return und Linefeed abgeschlossen. Die Signatur am Ende der Nachricht ist ein MD5-Hashwert über die gesendeten Daten und basiert auf einem einstellbaren Passwort. Dieser hexadezimale Hashwert ist immer 16 Zeichen lang und erlaubt es die Integrität der Nachricht und die Authentizität des Absenders zu überprüfen.

Um eine Verbindungsanfrage an das Board zu schicken kann der Befehl `telnet <Board IP-Adresse> <Port>` auf der Kommandozeile verwendet werden.

- Datenabfrage über RS-232
Wenn über die serielle Schnittstelle RS-232 des Kelvin-Boards ein beliebiges Zeichen empfangen wird, dann werden die Netzwerkadresse, der Listenport, die Subnet-Mask und das Gateway, sowie alle Temperaturen der Sensoren zurückgesendet. Die einstellbare Baudrate beträgt standardmäßig 19200 bps, 8 Datenbits, 1 Stoppbit und keine Flusssteuerung. Zur Abfrage eignet sich beispielsweise das Programm `Putty`.

Beispiel einer Abfrage über RS-232:

```
IP Address: 192.168.1.187 : 666
SubnetMask: 255.255.255.0
Gateway   : 192.168.1.1
```

```
TS1: 24.0 C
TS2: 21.7 C
TS3: 23.9 C
TS4: ---.- C
TS5: 27.6 C
TS6: 28.2 C
TS7: 27.6 C
TS8: ---.- C
```

Die Anbindung an ein Netzwerk erfolgt über Ethernet bzw. TCP/IP. Die IP-Adresse des Kelvin-Boards kann entweder statisch vergeben oder über DHCP dynamisch bezogen werden. Die Wahl wird in der Datei *Kelvin Board Firmware/Config/ProjectConfig.h* getroffen. Sie muss eingekompiliert und ins Board geflasht werden. Außerdem kann festgelegt werden, ob das Board auf Ping-Anfragen reagiert. Die MAC-Adresse kann beliebig eingestellt werden, wobei darauf zu achten ist, dass eine einmalige Adresse verwendet wird.

2.2 Hardware des Boards

Als Hardware des Sensor-Boards kommt das Development-Board „PIC-WEB REVISION-B“ der Firma Olimex (www.olimex.com) zum Einsatz, das den Mikrocontroller PIC18F67J60 von Microchip verwendet (Bild 2.1). Das Projekt kann auf andere Hardware portiert werden, wenn die entsprechenden Einstellungen in der Datei *Kelvin Board Firmware/Config/HardwareProfile.h* vorgenommen werden. Alle Konfigurationen des Kelvin-Boards können in der oben genannten getätigt werden.

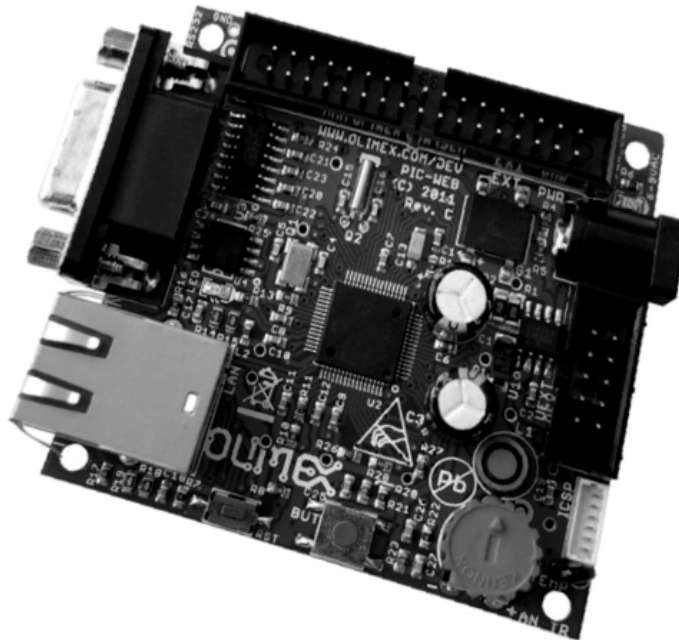


Bild 2.1: Das Olimex-Board PIC-WB

Um das Projekt zu kompilieren und ins Board zu flashen, eignet sich die kostenlose Entwicklungsumgebung „MPLAB IDE“ von Microchip, die von www.microchip.com heruntergeladen werden kann. Dort findet sich ebenfalls der benötigte Compiler „mplab18“ der in einer Testversion bezogen werden kann, die nach 60 Tagen lediglich die Optimierung des Codes abschaltet. Nachdem das Projekt kompiliert wurde, kann es direkt mit einem Programmer ins Board übertragen werden. Dazu kann beispielsweise das „ICD 3“ oder „PICkit 3“ benutzt werden. Die nötigen Anleitungen sind auf der Webseite von Microchip zu finden.



Bild 2.3: Ansicht des Temperatursensors

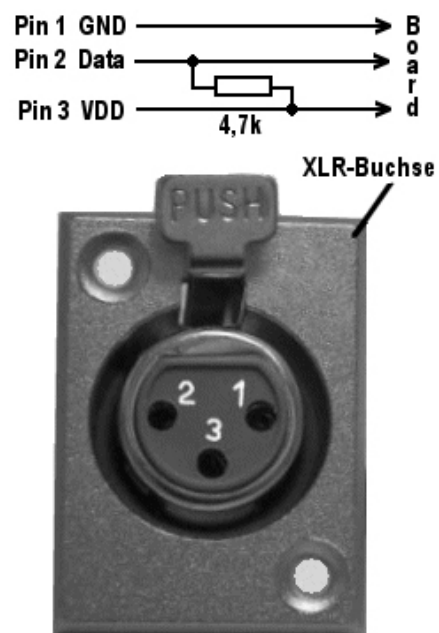


Bild 2.4: XLR-Buchse und Anschluß-Schema

3

Das Programm „kelvin“

Mit dem Kommandozeilen-Programm „kelvin“ können die Temperaturdaten des Kelvin-Boards abgefragt und zusammen mit einem Timestamp in eine MySQL-Datenbank gespeichert werden.

Um eine periodische Abfrage der Daten zu erreichen, kann man entweder einen Cronjob einrichten, der das Programm jeweils einmal aufruft oder man stellt ein Update-Intervall in Sekunden ein. Dann beendet sich das Programm nicht nach der Ausführung sondern wartet die Intervall-Zeit ab und führt danach das Programm in einer Endlosschleife aus. Zum Beenden dient die Tastenkombination [Strg] + [C].

Bei der Abfrage der Daten wird auch der vom Kelvin-Board mitgesendete Hashwert der Daten überprüft, um die Integrität der Daten und die Authentizität des Absenders festzustellen. Das Hash-Passwort muss mit dem Hash-Passwort im Kelvin-Board übereinstimmen. Falls das Programm einen anderen abweichenden Hashwert errechnet, dann wird eine Fehlermeldung ausgegeben und die nicht vertrauenswürdigen Daten werden nicht in die MySQL-Datenbank aufgenommen.

Immer wenn ein Fehler auftritt, wird dieser mit einer Zeitangabe auf der Standard-Fehlerausgabe ausgegeben. Es ist zu empfehlen, die Fehlerausgabe in eine Log-Datei umzulenken, um alle Fehler zu protokollieren (z. B. mittels `./kelvin 2>>/home/kelvin/error.log`). Die Datei `error.log` wird im Shellsript `errortomail` verwendet, um den Administrator über aufgetretene Fehler zu informieren.

Alle Einstellungen, wie zum Beispiel das Passwort für MySQL, können in der Konfigurationsdatei `kelvin.ini` vorgenommen werden, oder als Kommandozeilenparameter übergeben werden.

Beim Programmstart wird im aktuellen Verzeichnis nach der Datei `kelvin.ini` gesucht und gegebenenfalls als Grundeinstellungen geladen. Als erster Kommandozeilenparameter kann mittels `--load <Dateiname>` auch eine alternative Konfigurationsdatei mit absolutem Pfad geladen werden. Danach werden die Kommandozeilenparameter ausgewertet, die die in der Konfigurationsdatei getroffene Einstellungen gegebenenfalls überschreiben.

Die Syntax bestehen immer aus zwei Teilen: Dem Befehl, eingeleitet mit einem „-“ und dem Übergabewert der vom Befehl durch mindestens ein Leerzeichen getrennt sein muss.

In der Konfigurationsdatei werden leere Zeilen und Zeilen, die mit einem #-Zeichen beginnen, übersprungen. In jeder Zeile darf nur ein Befehl stehen und muss vom Übergabewert durch mindestens ein Leer- oder Tabulatorzeichen abgetrennt sein. Folgende Konfigurationsparameter sind möglich:

```
# [-l], [--load]           Lade Konfigurationsdatei
# [-i], [--interval]      Update-Intervall in Sekunden

# [-mu], [--user]         MySQL-User

# [-p], [--password]      MySQL-Passwort
# [-d], [--database]      MySQL-Database
# [-tb], [--table]        MySQL-Table
# [-mh], [--sqlhost]      MySQL-Host

# [-mp], [--sqlport]      MySQL-Port
# [-kh], [--boardhost]    Kelvin-Board Hostname
# [-kp], [--boardport]    Kelvin-Board Port

# [-hp], [--hashpassword] MD5-Hash-Passwort
# [-at], [--alarm]        Alarm-Temperatur in °C
```

- `-l Dateiname` oder `--load Dateiname`
Die angegebene Datei, z. B. `kelvin.ini`, wird geladen und die darin befindlichen Einstellungen gelesen. Dieser Befehl kann nur als erster Kommandozeilenparameter aufgerufen werden.
- `-i Zahl` oder `--interval Zahl`
Wird eine Zahl größer Null angegeben, wird diese als Update-Intervall in Sekunden interpretiert. Dann wird das Programm nach einem Durchlauf nicht beendet, sondern suspendiert sich, bis die gewählte Wartezeit abgelaufen ist. Wird der Wert 0 übergeben, läuft das Programm genau einmal durch und beendet sich dann selbst.
- `-mu Name` oder `--user Name`
Um sich mit einer MySQL-Datenbank zu verbinden, muss man sich mit Username und Passwort einloggen. Der übergebene Name ist der Username für MySQL.
- `-p Zeichenkette` oder `--password Zeichenkette`
Die Zeichenkette ist das Passwort für die MySQL-Datenbank.
- `-d Name` oder `--database Name`
Der Name der Datenbank innerhalb von MySQL. Wenn diese Datenbank nicht vorhanden ist, so wird sie neu angelegt.
- `-tb Name` oder `--table Name`
Der Name der Tabelle innerhalb von MySQL unter der die Daten gespeichert werden sollen.
- `-mh Hostname` oder `--sqlhost Hostname`
Die IP-Adresse, unter der die MySQL-Datenbank zu erreichen ist. `localhost` adressiert den lokalen Rechner.
- `-mp Nummer` oder `--sqlport Nummer`
Die Portnummer, unter der die MySQL-Datenbank zu erreichen ist. 3306 ist der Standardport für MySQL.
- `-kh Adresse` oder `--boardhost Adresse`
Die IP-Adresse des Kelvin-Boards. Wenn diese per DHCP zugeteilt wurde, kann die Adresse über RS-232 eingesehen werden.
- `-kp Nummer` oder `--boardport Nummer`
Die Portnummer, unter der das Kelvin-Board zu erreichen ist.
- `-hp Passwort` oder `--hashpassword Passwort`
Das Passwort, das zum Erzeugen des MD5-Hashwerts der Daten bei der Übertragung dient. Dieses muss gleich dem Passwort sein, das auf dem Kelvin-Board gespeichert ist (siehe Datei `Kelvin Board Firmware/Config/ProjectConfig.h`).
- `-at Zahl` oder `--alarm Zahl`
Die Zahl wird als Temperaturwert in Celsiursgraden interpretiert. Überschreitet ein Sensor diese Temperatur, so wird eine entsprechende Warnung mit Zeitangabe auf der Standard-Fehlerausgabe ausgegeben.

Das folgende Listing zeigt den Inhalt der `kelvin.ini` mit Standardwerten:

```

--interval      5
--user          root
--password      1234
--database      Kelvin
--table         Temperatur
--sqlhost       localhost
--sqlport       3306
--boardhost     192.168.1.187
--boardport     666
--hashpassword  1234567
--alarm         50.0

```

4

Das Programm „plotter“

Um den Temperaturverlauf in eine Grafik zu überführen, kann das Kommandozeilen-Programm „plotter“ verwendet werden. Es liest – wie auch *kelvin* – seine Einstellungen aus einer Datei oder übernimmt sie von der Kommandozeile. Die Konfigurationsdatei heißt *plotter.ini*.

Über die Parameter lassen sich gezielt Zeitbereiche auswählen, die der zu erstellenden Grafik zugrunde liegen. Als Plotterengine wird *gnuplot* verwendet, das seinerseits über die Datei *gnuplot.ini* konfiguriert werden kann. Aus diesem Grund muss *gnuplot* auf dem System installiert sein, auf dem das Programm *plotter* ausgeführt werden soll.

Die Temperaturwerte aus der MySQL-Datenbank werden über der Zeitachse für die ausgewählten Sensoren angezeigt. Um den Zeitbereich einzuschränken, lässt sich ein Start- und ein Endzeitpunkt festlegen. Dies geschieht beispielsweise durch die Angaben `-start '2012-12-24 18:00:00'` und `-end '2013-01-01 00:00:00'` erreichen. Alternativ zum Start- und Endpunkt kann auch ein Zeitbereich mittels einer Bereichsangabe (Range) festgelegt werden. Dazu wird eine Zahl und eine Einheit verwendet, die zusammen den Range ergeben, zum Beispiel `-range 2 -unit day`.

Es sind fünf verschiedene Kombinationen zulässig, um den Zeitbereich anzugeben, wobei Start, End und Range nicht alle gleichzeitig definiert werden können. Die Bedeutung der Kombinationen von Start, End und Range zeigt die Übersicht in Bild 4.1.

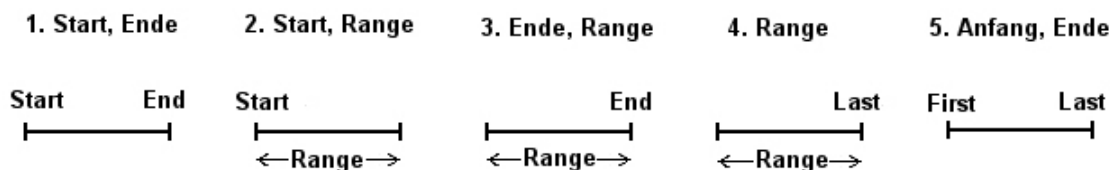


Bild 4.1: Übersicht der Werteintervalle

Dabei gilt:

1. Wenn nur Start und End gesetzt sind, wird der Zeitraum dazwischen ausgewählt.
2. Wenn nur Start und Range gesetzt sind, wird der Zeitraum ab Start bis Start plus Range ausgewählt.
3. Wenn nur End und Range gesetzt sind, wird der Zeitraum von End minus Range bis Range ausgewählt.
4. Wenn nur Range gesetzt ist, wird als End-Zeitpunkt der letzte bzw. aktuelle Datensatz gewählt.
5. Wenn keine Angabe gemacht wird, wird der gesamte Datensatz zugrunde gelegt.

Folgende Optionen lassen sich in der Konfigurationsdatei setzen:

```

# [-l], [--load]      Lade Config-Datei
# [-mu], [--user]    MySQL-User
# [-p], [--password] MySQL-Passwort
# [-d], [--database] MySQL-Datenbank
# [-tb], [--table]   MySQL-Tabelle
# [-h], [--host]     MySQL-Host
# [-mp], [--port]    MySQL-Port
# [-mo], [--sqlout]  Zielort der SQL-Ausgabedatei
# [-go], [--gnuout]  Zielort der SQL-Gnuplot-Ausgabedatei
# [-i], [--gnuini]   Pfad zur Datei gnuplot.ini, die mit geladen wird
# [-gp], [--gnupic]  Zielort der zu erzeugenden Grafikdatei
# [-s], [--start]    Start-Zeitpunkt im SQL-Datumsformat
# [-e], [--end]      Ende-Zeitpunkt im SQL-Datumsformat
# [-r], [--range]    Range (s. o.)
# [-u], [--unit]     Einheit des Range
# [-x], [--sensors]  Auswahl der Sensoren [all, ts1,ts2,ts3...]
# [-o], [--smooth]   Interpolation:
                        "smooth csplines",
                        "smooth bezier",
                        "smooth acsplines"
#
# [-t1], [--nameTS1] Name der Kurve in Grafik für TS1 bis TS8 (t1 .. t8)
# [-c1], [--colourTS1] Linienfarbe für TS1 bis TS8 (c1 .. c8)
# [-w1], [--styleTS1] Darstellung der Werte für TS1 bis TS8 (w1 .. w8)
                        z. B. "lines", "points", "linespoints"

```

Befehls Erläuterung für das Programm plotter:

- `-l` *Dateiname* oder `--load` *Dateiname* Die angegebene Datei z.B `plotter.ini` wird geladen und die darin befindlichen Einstellungen werden gelesen. Dieser Befehl kann nur als erster Kommandozeilenparameter aufgerufen werden.
- `-mu` *Name* oder `--user` *Name* Um sich mit einer MySQL-Datenbank verbinden zu können muss man sich als User mit Passwort einloggen. Der übergebene Name ist der Username für MySQL.
- `-p` *Zeichenkette* oder `--password` *Zeichenkette* Die Zeichenkette ist das Passwort für die MySQL-Datenbank.
- `-d` *Name* oder `--database` *Name* Der Name der Datenbank innerhalb von MySQL. Wenn diese Datenbank nicht vorhanden ist, so wird sie neu angelegt.
- `-tb` *Name* oder `--table` *Name* Der Name der Tabelle innerhalb von MySQL unter der die Daten gespeichert werden sollen.
- `-h` *Hostname* oder `--host` *Hostname* Die IP-Adresse unter der die MySQL-Datenbank zu erreichen ist. `localhost` adressiert den lokalen Rechner.
- `-mp` *Nummer* oder `--port` *Nummer* Die Portnummer unter der die MySQL-Datenbank zu erreichen ist. 3306 ist der Standardport für MySQL.
- `-mo` *Dateiname* oder `--sqlout` *Dateiname* In diese Datei wird die Ausgabe der MySQL-Datenbank nach der individuellen Abfrage geschrieben. Sie enthält die Daten, aus denen die Grafik erzeugt wird, danach kann sie gelöscht werden.
- `-go` *Dateiname* oder `--gnuout` *Dateiname* In diese Datei werden die Anweisungen geschrieben, die dann von gnuplot ausgeführt werden. Innerhalb dieser Anweisungen wird die `gnuplot.ini` geladen.
- `-i` *Dateiname* oder `--gnuini` *Dateiname* In dieser Datei können Einstellungen für gnuplot vorgenommen werden, die bei der Erstellung der Grafik mit berücksichtigt werden.
- `-gp` *Dateiname* oder `--gnupic` *Dateiname* Der Zielort und der Name der zu erzeugenden Gnuplot-Grafik
- `-s` *Datum/Uhrzeit* oder `--start` *Datum/Uhrzeit* Der Startzeitpunkt beispielsweise im Format `YYYY-MM-DD hh:mm:ss` ab dem die Daten für die Erzeugung der Grafik verwendet werden. Jedes MySQL-Datumsformat ist gültig.
- `-e` *Datum/Uhrzeit* oder `--end` *Datum/Uhrzeit* Der Endzeitpunkt beispielsweise im Format `YYYY-MM-DD hh:mm:ss` bis zu dem die Daten für die Erzeugung der Grafik verwendet werden. Jedes MySQL-Datumsformat ist gültig.

- `-r Zahl` oder `--range Zahl` Die Zahl ergibt zusammen mit der Einheit einen Zeitraum, wie zum Beispiel: `-r 1 -u day`
- `-u Einheit` `--unit Einheit` Die Einheit in der der Zeitraum definiert werden soll. Gültig sind alle MySQL-Einheiten, wie zum Beispiel: `MINUTE`, „`HOUR`“, „`DAY`“, „`WEEK`“, „`MONTH`“ oder „`YEAR`“
- `-x Auswahl` oder `--sensors` Die Auswahl der Sensoren, die angezeigt werden sollen. Es ist eine beliebige Kombination aus den Sensoren „`ts1`“ bis „`ts8`“ möglich, die durch ein Komma aber ohne Leerzeichen getrennt werden. Bsp: „`ts1,ts4,ts3`“ Mit „`all`“ werden alle Sensoren ausgewählt.
- `-o Text` oder `--smooth Text` Mit Text kann eine Interpolations- oder Approximationsfunktion ausgewählt werden, die für alle Sensoren gilt. Dadurch können sehr zackige Kurve geglättet dargestellt werden. Es sind alle gnuplot Funktionen gültig wie zum Beispiel: „`smooth csplines`“, „`smooth bezier`“ oder „`smooth acsplines`“.
- `-t1 Name` oder `--nameTS1 Name` Der Name des jeweiligen Sensors, wie er in der Legende der Grafik angezeigt werden soll. Dies gilt analog für alle Sensoren `TS1` bis `TS8`.
- `-c1 Farbe` oder `--colourTS1 Farbe` Die Farbe des jeweiligen Graphen. Gültig sind alle Farbbezeichnungen von gnuplot als Text oder als RGB Farbcode, zum Beispiel: „`#CCCCCC`“. Dies gilt analog für alle Sensoren `TS1` bis `TS8`.
- `-w1 Text` oder `--styleTS1 Text` Mit Text kann für jeden Sensor eine individuelle Darstellung gewählt werden. Gültig sind alle Bezeichnungen von gnuplot wie „`lines`“, „`points`“ oder „`line-spoints`“. Dies gilt analog für alle Sensoren `TS1` bis `TS8`.

Die Standardwerte der Konfigurationsdatei sind:

```

--user          root
--password      1234
--database      Kelvin
--table         Temperatur
--host          localhost
--port          3306
--sqlout        /home/kelvin/temp/sqloutput
--gnuout        /home/kelvin/temp/gnuoutput
--gnuini        /home/kelvin/gnuplot.ini
--gnupic        /home/kelvin/plots/plot.png
--sensors       all
--nameTS1       Server 1
--colourTS1     black
--styleTS1      lines
--nameTS2       Server 2
--colourTS2     green
--styleTS2      lines
--nameTS3       Server 3
--colourTS3     blue
--styleTS3      lines
--nameTS4       Server 4
--colourTS4     magenta
--styleTS4      lines
--nameTS5       Server 5
--colourTS5     light-blue
--styleTS5      lines
--nameTS6       Server 6
--colourTS6     red
--styleTS6      lines
--nameTS7       Server 7
--colourTS7     grey
--styleTS7      lines
--nameTS8       Server 8
--colourTS8     orange
--styleTS8      lines

```


5

Skripting und crontab

5.1 Das Shellscript „errortomail“

Das Script „errortomail“ prüft, ob die Datei *error.log* existiert und nicht leer ist. Gegebenenfalls wird der Inhalt dieser Datei an eine voreinstellbare E-Mail Adresse gesendet und danach die Log-Datei gelöscht. Sinnvollerweise muss das Skript in regelmäßigen Abständen per Cronjob aufgerufen werden, um einen Administrator rechtzeitig über einen Temperaturanstieg oder den Ausfall des Kelvin-Boards zu informieren. Im Script selbst müssen die folgenden beiden Variablen entsprechend angepasst werden.

```
ERRORLOG="/home/kelvin/error.log"    # Die zu überwachende Log-Datei
MAILADDRESS="kelvin@non-datur.de"    # Die Ziel-Mailadresse
```

Listing des Scripts:

```
#!/bin/bash
ERRORLOG="/home/mike/error.log"
MAILADDRESS="kelvin@db.da.private.hm.edu"

# Die Groesse der Datei err.log wird ermittelt
if [ -e $ERRORLOG ]
then
  BYTECOUNT=$(du -k $ERRORLOG | cut -f1)
else
  BYTECOUNT=0
fi

if [ $BYTECOUNT -ne 0 ]
then
  # Die Datei err.log hat Inhalt und wird per mail verschickt und geloescht
  cat $ERRORLOG | mailx -s "Kelvin Errorlog" $MAILADDRESS
  rm $ERRORLOG
fi
```

5.2 Die crontab-Einträge

Vorausgesetzt, der Inhalt des Ordners *Plotter and Kelvin* befindet sich unter */home/kelvin*, kann ein rontab-Eintrag folgendermaßen aussehen. Wenn für *kelvin.ini* und *plotter.ini* nur der Besitzer Leserecht besitzt, werden die Passwörter für andere Nutzer verborgen.

```
# Minute Stunde Tag Monat Wochentag
# 0-59 0-23 1-31 1-12 0-6 (0=So)

# Alle 10 Minuten werden die Temperaturwerte der Server im Serverraum abgerufen
# und in die MySQL Datenbank gespeichert.

0,10,20,30,40,50 * * * * /home/kelvin/kelvin -l /home/kelvin/kelvin.ini
>/dev/null 2>>/home/kelvin/error.log

#Immer um ca Mitternacht wird ein Plot über den letzten Tag erstellt
```

```
55 23 * * * /home/kelvin/plotter -l /home/kelvin/plotter.ini
    -r 1 -u day -gp /home/kelvin/plots/lastday.png >/dev/null 2>&1

# Immer Sonntags um ca Mitternacht wird ein Plot über die letzte Woche erstellt

56 23 * * 0 /home/kelvin/plotter -l /home/kelvin/plotter.ini
    -r 1 -u week -gp /home/kelvin/plots/lastweek.png >/dev/null 2>&1

# Immer am ersten des Monats um ca Mitternacht wird ein Plot über
# den letzten Monat erstellt

57 23 1 * * /home/kelvin/plotter -l /home/kelvin/plotter.ini
    -r 1 -u month -gp /home/kelvin/plots/lastmonth.png -o "smooth bezier"
    >/dev/null 2>&1

# Jede Stunde wird ein Skript gestartet, dass ueberprueft ob Daten in der
# Error.log vorhanden sind und diese dann an eine Mailadresse schickt.

5 * * * * /home/kelvin/errortomail
```


Stichwortverzeichnis

tocchapterStichwortverzeichnis

1-Wire-Protokoll, 9

Anzeige über Webpage, 7

crontab, 11, 17

crontab-Einträge, 17

Dateien, 5

Datenabfrage über RS-232, 7

Datenabfrage über TCP/IP, 7

Datenbank, 5

Error-log, 11

error.log, 11, 17

errortomail, 5, 17

Fehler-Log, 11

gnuplot, 5, 13

gnuplot.ini, 13

Hashwert, 7, 11

Hochladen Webpage, 9

ICD 3, 8

kelvin, 5, 11

Kelvin Board Firmware/Config/ProjectConfig.h,
8, 9, 12

Kelvin-Board, 5, 7, 9

Kelvin-Board, Hardware, 8

kelvin.ini, 11

Konfigurationsdatei gnuplot.ini, 13

Konfigurationsdatei kelvin.ini, 11

Konfigurationsdatei plotter.ini, 13

MPFS2.exe, 9

MPLAB IDE, 8

MySQL, 5, 11

Netzwerkanbindung, 8

Olimex, 8

Parameter kelvin.ini, 12

PIC-WEB, 8

PICKit 3, 8

plotter, 5, 13

plotter.ini, 13

Port, 7

Programm gnuplot, 5

Programm kelvin, 5, 11

Programm plotter, 5, 13

Putty, 7

Serverraum, 5

Shellscript errortomail, 5, 17

Start, End, Range, 13

Steckerbelegung, PIC-WEB, 9

TCP/IP, 7

telnet, 7

Temperaturdaten, 11

Temperaturmessung, 5

Temperatursensoren, 5, 7, 9

Temperatursensoren, Anschluss, 9

Temperaturwerte, 13

Template, 9

Verzeichnisstruktur, 5

XLR-Buchse, Belegung, 9

XLR-Stecker, 9

Zeitbereiche, 13