Technische Universität Ilmenau

Fakultät für Elektrotechnik und Informationstechnik

Diplomarbeit

Erarbeitung von Konzepten zur Überwachung und Ferndiagnose leistungselektronischer Systeme

vorgelegt von:	Jens Bierkandt
geboren am:	14. Oktober 1972 in Konstanz
Studiengang:	Elektrotechnik und Informationstechnik
Studienrichtung:	Elektronische Medientechnik
Verantwortlicher Professor:	Dr.-Ing. habil. T. Doll
Betreuende wiss. Mitarbeiter:	Dipl.-Ing. J. Lehmann
	Dipl.-Ing. R. Siemieniec

Beginn der Arbeit:	01.12.2001
Abgabe der Arbeit:	31.05.2002
Registriernummer:	2141-01 D/02

Ilmenau, den 31.05.2002

Danksagung

An dieser Stelle möchte ich mich bei dem verantwortlichen Professor meiner Diplomarbeit, Dr.-Ing. habil. T. Doll bedanken. Mein Dank gilt ebenfalls den betreuenden wissenschaftlichen Mitarbeitern Dipl.-Ing. J. Lehmann und Dipl.-Ing. R. Siemieniec.

Weiterhin möchte ich meinen Eltern danken, die mir das Studium ermöglicht haben.

Inhaltsverzeichnis

1. Einleitung **9**

2. Grundlagen **11**

2.1. Ziel der Überwachung 11
2.2. Einsatzgebiete mobiler Diagnose- und Wartungssysteme 11
2.3. Voraussetzungen für einen dauerhaften Messbetrieb 12
2.4. Mögliche Übertragungswege 13
2.4.1. Festnetz 13
2.4.2. Mobilfunk 13
2.4.3. Internet 14
2.4.4. Zusammenfassung 14

3. Auswahl und Installation der Komponenten **15**

3.1. Hardware 15
3.1.1. Messstation 16
3.1.2. Server 17
3.2. Software der Messstation 17
3.2.1. Betriebssystem 18
3.2.2. Remote Access Server (RAS) 18
3.2.3. Messprogramm 18
3.3. Software des Servers 18
3.3.1. Linux 18
3.3.2. Apache 19
3.3.3. MySQL 19
3.3.4. PHP: Hypertext Preprocessor 19
3.3.5. JPGraph 20
3.3.6. Zusammenfassung 20
3.4. Konfiguration eines sicheren Webservers 20

4. Entwicklung des Messbetriebs **23**
4.1. Übersicht . . . 23
4.1.1. Die Programme . . . 23
4.1.2. Die Programmstruktur . . . 24
4.1.3. Die Datenbankstruktur . . . 24
4.2. Abfrage der Daten der Messstation . . . 25
4.2.1. Definitionen der Schnittstellenparameter . . . 27
4.2.2. Verbindungsaufbau und Authentifizierung . . . 28
4.2.3. Datenübertragung . . . 28
4.3. Aufbereitung der Messdaten . . . 29
4.3.1. Einlesen der Daten in die Tabellen . . . 29
4.4. Menüführung . . . 29
4.4.1. Authentifizierung am Server . . . 30
4.4.2. Hauptmenü . . . 30
4.4.3. Konfigurationsmenü . . . 30
4.4.4. Auswahlmenü zur Darstellung der Daten . . . 32
4.4.5. Anzeige der Messwerte . . . 32
4.4.6. Administration der Datenbank . . . 36
4.5. Übersicht möglicher Datenmengen . . . 36
4.6. Sicherheit bei der Datenübertragung . . . 37
4.7. Mögliche gravierende Fehlerfälle . . . 37
4.7.1. Ausfall der Messstation . . . 37
4.7.2. Ausfall von Sensoren . . . 38
4.7.3. Kurzzeitige Unterbrechungen der Aufzeichnung . . . 38
4.7.4. Abbruch der Verbindung bei der Datenübertragung . . . 38
4.7.5. Ausfall des Servers . . . 39

5. Fernwartung der Messstationen **41**
5.1. Vorüberlegungen . . . 41
5.2. Funktionsweise . . . 42

6. Zusammenfassung **45**

A. Quellcode **49**
A.1. .htaccess . . . 49
A.2. .htpasswd . . . 49
A.3. admin.sql . . . 49
A.4. cron_job.php4 . . . 50

A.5. pppconn.sh 53
A.6. config.php4 54
A.7. convert2mysql.php4 55
A.8. index.php4 60
A.9. profile.php4 62
A.10. check_profile.php4 66
A.11. menugraf.php4 70
A.12. showgraf.php4 77
A.13. graph.php4 86
A.14. diff_count.php4 92
A.15. count.php4 99

B. Erklärung **103**

C. Thesen **105**

1. Einleitung

Zu Beginn des vorletzten Jahrhunderts war sicher den wenigsten Menschen bewusst, wie umfassend die Entdeckung des elektrischen Stroms unser Leben beeinflussen würde. Heutzutage ist ein Gerät ohne elektronische Bauteile kaum mehr vorstellbar. Kraftfahrzeuge sind durch sie sicherer geworden, Computer beherrschen unser Leben und die ganz normale Glühbirne möchte auch keiner mehr missen.

Inzwischen hängt von einigen Geräten auch unser Leben ab, Ausfälle können fatale wirtschaftliche Folgen haben. Fällt der Strom durch ein kleines Bauteil im industriellen Bereich aus, so ist das nicht nur sehr ärgerlich, sondern kann unter Umständen auch Schadenersatzansprüche nach sich ziehen.

So sind Hersteller elektronischer Bauteile, insbesondere leistungselektronischer Bauteile wie sie beispielsweise bei der Antriebstechnik oder Netzkupplung vorkommen, besonders daran interessiert, ihre Geräte im Rahmen wirtschaftlicher Grenzen ausfallsicher zu gestalten. Um eine möglichst exakte Vorhersage der Lebensdauer geben zu können, werden Vorserienmodelle solcher Bauteile meist im Rahmen der Qualitätssicherung einer beschleunigten Alterung unterzogen. Das heißt, sie werden Bedingungen ausgesetzt, die im wirklichen Einsatz auch auftreten, nur in kürzeren Abständen. Besonders im Bereich der Aufbau- und Verbindungstechnik von Leistungshalbleitern haben Temperaturschwankungen eine besondere Bedeutung. Je häufiger und größer die Temperaturunterschiede sind, desto stärker wird das Bauteil z.B. durch mechanischen Stress belastet.

Das Problem hierbei ist, dass zuerst die realen Bedingungen, denen eine Anlage ausgesetzt ist, bestimmt werden müssen. Dies wird zumeist durch eine langfristige Untersuchung der Bauelemente beziehungsweise deren Vorgänger oder Prototypen im normalen Einsatz herausgefunden.

Ziel meiner Diplomarbeit ist es, die Daten dieser Bauelemente unter realen Bedingungen zu sammeln, an einen Server zu übermitteln und dort grafisch auszuwerten. Zur Demonstration soll dies anhand einer Überwachung einer Windkraftanlage komplett entwickelt werden. Das System wurde dabei so ausgelegt, dass es viele Aufgabenbereiche abdecken kann und einfach zu skalieren ist. So ist es durchaus möglich, die Temperatursensoren in beliebigen anderen elek-

tronischen Geräten einzusetzen, ohne dass die Hardware und die Programme geändert werden müssen.

Es können beliebig viele Messrechner in das System aufgenommen werden und damit beliebig viele Sensoren abgefragt werden. Weiterhin können die Messwerte über das Internet in einem Webbrowser ausgewertet werden. So besteht die Möglichkeit, gleichzeitig von den unterschiedlichsten Betriebssystemen auf die Daten zuzugreifen.

2. Grundlagen

Bevor die Umsetzung der Messdatenerfassung am Beispiel einer Windkraftanlage gezeigt wird, möchte ich in diesem Kapitel vor allem die Gründe der Überwachung leistungselektronischer Systeme darlegen.

2.1. Ziel der Überwachung

Eine Frage, die oft gestellt wird, ist die nach der Lebensdauer von elektronischen Bauteilen, Baugruppen oder Systemen. Gerade in kritischen Bereichen, in denen ein Ausfall fatale wirtschaftliche Folgen haben könnte, ist eine möglichst genaue Vorhersage der Lebensdauer wichtig. So werden in künstlichen Umgebungen die Bauteile beschleunigt gealtert, um den Grund des Ausfalles herauszufinden. Dabei werden sie vorher fest definierten Kennwerten (z.B. Temperaturschwankungen) ausgesetzt, die im realen Einsatz auch auftreten können. Um diese Kennwerte herauszufinden, ist es notwendig, eine langfristige Überwachung der elektronischen Bauteile (z.B. die Temperaturabnahme an noralgischen Punkten) vorzunehmen. Nur so können sie, beziehungsweise die verbesserten Nachfolger und Prototypen, im Labor simulierten Werten ausgesetzt werden, welche die realen widerspiegeln.

2.2. Einsatzgebiete mobiler Diagnose- und Wartungssysteme

Sind die zu überwachenden leistungselektronischen Systeme ohne großen Aufwand erreichbar, können sie fest mit einem Gerät verdrahtet werden, welches die Messdaten auswertet. Oft kommt es jedoch vor, dass die Bauteile sich nicht in geschlossenen Räumen befinden und die Daten über eine große Entfernung übertragen werden müssen. Dies trifft z.B. bei Windkraftanlagen zu oder wenn Fahrantriebe überwacht werden sollen. Hier sind besondere Vorkehrungen zu treffen, um die Messwerte erfassen, zwischenspeichern, übertragen und auswerten zu können.

Ausfallgründe	Lösungsvorschläge
Softwareabstürze	hardwaremäßige Erkennung, automatischer Neustart
extreme Umweltbedingungen	Einsatz eines hochwertigen Industriecomputers
Ausfall von Hardwarebauteilen	redundante Ausführung wichtiger Bauteile
Stromausfall	Einsatz einer USV, welche Stromausfälle überbrückt
Festplattenausfall	Festplattenspiegelung mit Hilfe von RAID

Tabelle 2.1.: Häufige Gründe für den Ausfall eines Computers

2.3. Voraussetzungen für einen dauerhaften Messbetrieb

Es wäre fatal, wenn ein unzugänglicher Computer, wie ein Messrechner auf einer Windkraftanlage im Meer, durch verschiedene Umstände seine Daten nicht mehr aufzeichnet oder komplett abstürzt und nicht mehr neu gebootet werden kann. In so einem Fall bliebe nur die Möglichkeit, den Rechner manuell neu zu starten. Um dies zu verhindern bzw. die Ausfallzeiten so gering wie möglich zu halten, muss ein solches System entsprechend ausgerüstet und konfiguriert werden. Tabelle 2.1 gibt einen Überblick möglicher Fehlerfälle und zeigt Lösungsvorschläge zur Verhinderung eines Totalausfalles auf.

Die häufigste Ursache für den Ausfall eines Computersystems sind Softwarefehler und die damit verbundenen Abstürze. Diese müssen zuverlässig von der Hardware erkannt werden und den Rechner veranlassen, automatisch neu zu booten. Der in unserem Fall eingesetzte Messrechner ist aus diesem Grund ein hochwertiger Industriecomputer, der eine Watchdog-Funktion zur Erkennung von Softwareabstürzen besitzt.

Weiterhin ist mit schwierigeren Umweltbedingungen zu rechnen. Ein hochwertiger Computer in Verbindung mit einem robusten Industriegehäuse bietet hier den Vorteil, dass extreme Belastungen durch Stoß, Hitze oder ähnlichem besser kompensiert werden als bei einem Büro-PC.

Da Software nie 100%-ig fehlerfrei geschrieben werden kann, ist die Software so zu konfigurieren, dass sie bei einem Reboot automatisch die erforderlichen Programme startet. Weiterhin muss sie versuchen, die Daten zu retten und Skripte ausführen, um den weiteren Betriebsablauf (hier die Messdatenaufnahme) zu gewährleisten.

Nicht nur die Software, sondern auch die Hardware kann durch eine falsche Konfiguration oder Programmierung Fehlfunktionen erzeugen beziehungsweise altern. Um einen Komplettausfall der Anlage bis zu einer Reparatur zu verhindern, sollten die wichtigsten Komponenten redundant ausgelegt werden.

Um mögliche Festplattenfehler zu umgehen, sollten alle Daten auf eine weitere Festplatte gespiegelt werden, die im Fehlerfalle angesprochen wird. Dies kann z.B. durch ein RAID-

System ermöglicht werden, welches die Daten automatisch auf mehrere Festplatten schreibt. Fällt eine aus, bleibt immer noch eine Reservefestplatte mit dem kompletten Inhalt der Daten. Damit wird ein Zeitgewinn für eine manuelle Reparatur gewonnen.

Um Datenverlust vorzubeugen, ist weiterhin zu empfehlen, dass die Daten regelmäßig auf einen anderen Computer übertragen und dort als Backup gespeichert werden.

Die Fehlerfälle werden noch einmal genauer in Kapitel 4.7 auf Seite 37 anhand der konkreten Umsetzung eines Messsystems betrachtet.

2.4. Mögliche Übertragungswege

Für die Übertragung der Daten vom Messrechner zum Server, welcher die Daten sammelt und auswertet, stehen mehrere Möglichkeiten zur Verfügung. Darunter fällt in erster Linie, sich über das Festnetz oder Internet zu verbinden bzw. von jeder Technik etwas zu nutzen. Es ist jedoch zu beachten, dass sich die zu überwachenden Geräte in Gebieten befinden können, in denen einige Übertragungswege nicht zur Verfügung stehen. So kann es durchaus sinnvoll sein, den Mobilfunk in Betracht zu ziehen.

2.4.1. Festnetz

Die sicherste Lösung besteht darin, einen Messrechner direkt über das Telefonnetz mit dem Server zu verbinden. Ein Ausspionieren der Daten, wie es in ungünstigen Fällen im Internet vorkommen kann, ist hier nicht möglich. Da der Messrechner auch ferngewartet werden muss und ein Hackerangriff fatale Folgen haben könnte, sind unsere Systeme über das Festnetz verbunden.

2.4.2. Mobilfunk

Besteht nicht die Möglichkeit, dem Messrechner einen Telefonanschluss zur Verfügung zu stellen, muss auf die Mobilfunktechnik zurückgegriffen werden. Die Daten werden mittels eines an den Computer angeschlossenen Handys oder direkt über eine GSM-Karte übertragen. Der Nachteil ist jedoch die geringe Übertragungsgeschwindigkeit von maximal 9.600 Bit/s.

GPRS ist in diesem Fall keine Alternative. Hier können nur über einen Provider Daten übertragen werden, eine direkte Anwahl unseres Servers ist nicht möglich.

Als dritte Möglichkeit bietet sich allerdings HSCD an. Es erlaubt eine Direktverbindung und einen Datendurchsatz zwischen 14.400 und 43.200 Bit/s. Allerdings schlägt in allen Fällen der gegenwärtig teure Einheitenpreis der zur Verfügung stehenden Netzbetreiber zu Buche.

Der Mobilfunk ist somit nur dann als Alternative zu empfehlen, wenn keine anderen Möglichkeiten der Übertragung zur Verfügung stehen.

2.4.3. Internet

Das Internet bietet interessante Varianten. Es wäre möglich, dass sich der Messrechner automatisch ins Internet einwählt und eine Verbindung zum Server herstellt. Dabei muss nur der relativ günstige Einheitenpreis eines Internetproviders bezahlt werden.

Der Nachteil liegt jedoch darin, dass nicht aktiv vom Server auf den Messrechner zugegriffen werden kann (z.B. im Falle einer Fernwartung), sondern dass solange gewartet werden muss, bis der Messrechner sich selbstständig einwählt. Weiterhin besteht die Möglichkeit, Opfer eines Hackerangriffs zu werden. Es gibt zwar hoffnungsvolle Ansätze, dies z.B. mit einem Virtual Private Network (VPN) oder IPsec zu umgehen, doch eine 100%-ige Sicherheit kann nie gewährleistet werden.

Als letzter Nachteil ist die nicht immer gegebene Zuverlässigkeit des Internets zu nennen. So besteht die Möglichkeit, dass keine Verbindung zum Server zu bekommen ist, oder dass der Provider seine Zugangsdaten geändert hat und somit ohne Backuplösung nicht mehr mit dem Messrechner kommuniziert werden kann.

2.4.4. Zusammenfassung

Unter den oben genannten Aspekten wurde für unser System zugunsten des Festnetzes entschieden und, falls kein Telefonanschluss besteht, die Möglichkeit gewählt, den Mobilfunk zu nutzen. Das Internet wird zur Kommunikation nicht verwendet, wenngleich über die Telefonleitungen die gleichen Protokolle laufen (TCP/IP und FTP) wie im Internet.

Weiterhin wurde der Server als aktiver Computer festgelegt. Er wählt sich in regelmäßigen Intervallen bei den Messrechnern ein und überträgt die Daten auf seine Festplatte. Soll ein Messrechner gewartet werden, wird er in der gleichen Art und Weise mit dem Server verbunden.

3. Auswahl und Installation der Komponenten

Im folgenden gehe ich auf eine konkrete Umsetzung der Diagnose und Überwachung von leistungselektronischen Systemen ein. Betrachtet wird ein Konzept, in dem mehrere Messrechner in Windkraftanlagen eingebaut werden. Diese sollen Temperaturunterschiede mittels Sensoren an leistungselektronischen Bauteilen messen. Ein Server an der Universität fragt die Daten regelmäßig ab und diese können von dort über eine Webschnittstelle in den unterschiedlichsten Varianten dargestellt und interpretiert werden. Weiterhin besteht die Möglichkeit, die Messcomputer in den Windkraftanlagen von der Universität aus zu administrieren

Für den Datentransfer und die Verwaltung der Messcomputer wurde zugunsten der Variante in Abbildung 3.1 auf der nächsten Seite entschieden. Der Server fragt in regelmäßigen Abständen beliebig viele Messcomputer ab und überträgt die gemessenen Daten in eine Datenbank. Da im Einsatzgebiet dem Messrechner nicht immer eine schnelle ISDN-Leitung zur Verfügung steht, soll wahlweise ein analoges Modem oder eine GSM-Karte bzw. ein Handy als Datenübertragung dienen. An den Server besteht der Anspruch, mit allen Protokollen umgehen zu können. Auf dem Server können die Daten dann über eine Webschnittstelle grafisch dargestellt werden.

Das Internet als Übertragungsmedium wurde zunächst ausgeschlossen, da der Datenabruf bzw. die Fernwartung vom Server aus initiiert werden soll und es sicherheitstechnische Bedenken gibt.

3.1. Hardware

Die Hardwareauswahl zwischen Messrechner und Server gestaltet sich völlig unterschiedlich. Im folgenden werden die wichtigsten Voraussetzungen erläutert.

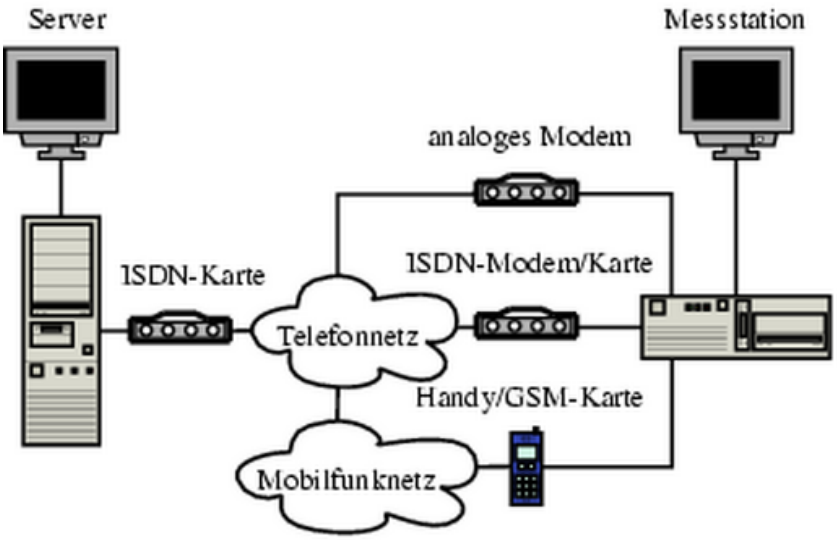

Abbildung 3.1.: Vernetzung des Servers mit einem Messrechner

Computer	Ausstattung
Mainboard	MITAC IPC MSC-3685A
Prozessor	Intel Pentium III, 800 MHz
Messkarte	NuDAQ PCI-9118 HR
Festplattenkapazität	1,95 GB

Tabelle 3.1.: Die wichtigsten Hardwarekomponenten der Messstation

3.1.1. Messstation

Der Messcomputer muss so zuverlässig wie möglich funktionieren, da eine Wartung auf den Windkraftanlagen relativ aufwendig wäre. Um dies zu gewährleisten, wurde ein Industriecomputer ausgewählt, der weitaus höheren Ansprüchen genügt als z.B. ein Office-Rechner. So ist es u.a. möglich, hardwaremäßig einen Absturz des Betriebssystems zu erkennen und den Rechner automatisch neu zu booten. Da Computerprogramme nie fehlerfrei geschrieben werden können, besteht somit ein Weg, der Gefahr eines Totalausfalles des Systems vorzubeugen.

Die Geschwindigkeit des Rechners spielt eher sekundär eine Rolle. Die Daten werden mit einer Messkarte gesammelt und auf der Festplatte gespeichert. Der Prozessor sollte dabei nur gering belastet werden. Auch bei der Datenabfrage durch den Server fallen keine hohen Rechenzeiten an.

Die eingesetzte Messkarte bietet eine Genauigkeit von 16 Bit bei analogem Eingang. Sie unterstützt eine Samplingrate von bis zu 100 KHz (Analog/Digital), kommt mit bipolaren und unipolaren Eingangssignalen zurecht und besitzt acht differenzielle Eingänge.

Eine Übersicht der wichtigsten Hardwarekomponenten der Messstation zeigt Tabelle 3.1.

Computer	Ausstattung
Prozessor	Intel Celeron 500 MHz
Speicherkapazität	320 MB
Festplattenkapazität	5 GB
ISDN-Karte	AVM Fritz PCI 2.0

Tabelle 3.2.: Die wichtigsten Hardwarekomponenten des Servers

3.1.2. Server

An die Serverhardware besteht beim Einlesen der Daten kein großer Geschwindigkeitsanspruch. Die Datenübertragung und -konvertierung in die Datenbank kann bequem nachts im Hintergrund laufen und ist nicht echtzeitrelevant. Lediglich bei der Berechnung der Grafiken muss der Server schnell genug sein, um die Daten in akzeptabler Zeit anzeigen zu können. Eine Übersicht der wichtigsten Hardwarekomponenten zeigt Tabelle 3.2.

Eine größere Hürde stellt das Interface zu den verschiedenen Datenübertragungsprotokollen dar. So soll mit möglichst geringem Aufwand ein analoges Modem, eine ISDN-Karte oder eine GSM-Karte auf dem Messrechner angesprochen werden können. Es wurde deshalb zugunsten einer ISDN-Karte entschieden, die softwaremäßig analoge Daten emulieren kann. Dies geschieht jedoch technisch bedingt nur mit einer maximalen Transferrate von 14.400 Bit/s. Für eine schnellere analoge Datenübertragung würde eine aktive ISDN-Karte benötigt werden, die hardwaremäßig die analogen Daten emuliert.

Der Server ist so eingerichtet, dass über eine Maske beliebig viele Messrechner eingegeben werden können, von denen dann sequentiell über die Telefonleitung die Daten abgerufen werden. So besteht die Möglichkeit, einen ganzen Park von Windkraftanlagen mit Messrechnern auszustatten und zu überwachen.

Weiterhin ist er mit einer Netzwerkkarte an das Internet angeschlossen. Die Softwareprogrammierung wurde so ausgelegt, dass die Daten von jedem Computer der Universität mit einem Login über einen Webbrowser ausgewertet werden können. Somit besteht Plattformunabhängigkeit. Es kann von jeder Prozessorarchitektur mit jedem beliebigen Betriebssystem auf den Server zugegriffen werden.

3.2. Software der Messstation

Auch bei den Messstationen steht die Stabilität des Computers im Vordergrund. Jedoch müssen Bauteile wie z.B. die Messkarte eingebaut werden, deren mitgelieferte Treiber nur wenige Betriebssysteme unterstützen.

3.2.1. Betriebssystem

Aufgrund der Messkarte, die nur Treiber für die Windows-Betriebssystemreihe von Microsoft zu Verfügung stellt, wurde zugunsten von Windows NT entschieden. Dieses Betriebssystem wurde ausreichend getestet und jahrelang verbessert, so dass es inzwischen als recht stabil angesehen wird. Zusätzlich wurde noch ein kleiner FTP-Server aufgesetzt, der für die Übertragung der Daten zuständig ist. Die Installation kann unter [1] nachgelesen werden.

3.2.2. Remote Access Server (RAS)

Der RAS wird benötigt, um eine Einwahl von einem anderen Computer zu ermöglichen. Nachdem das Modem erkennt, dass ein Anrufer eine Verbindung aufbauen möchte, kümmert das Programm sich selbstständig um die Authentifizierung und um das Starten der Übertragungsprotokolle wie TCP/IP. Eine genaue Beschreibung, wie ein Remote Access Server eingerichtet werden muss, kann unter [2] nachgelesen werden.

3.2.3. Messprogramm

Zur Abfrage der Daten von der Messkarte und deren Steuerung wird das HewlettPackard-Visual Engineering Environment (HP-VEE) [3] eingesetzt. Dieses Programm ist speziell zur Erstellung von Software für Test- und Messanwendungen entwickelt worden. Es kann die Entwicklungszeiten einer Softwareentwicklung in diesem Bereich um 25-80% verkürzen [4].

3.3. Software des Servers

Bei der Softwareauswahl stand die Stabilität des Betriebssystems im Vordergrund. Zusätzlich sollte das System kostengünstig, schnell und flexibel sein. Zur Auswahl standen dabei die Betriebssysteme der Windows-Serie von Microsoft und verschiedene Linux-Distributionen. Um alle Kriterien hinreichend zu erfüllen, fiel die Entscheidung zugunsten von Linux. Als Distribution kam SuSE 7.3 Professional [5] zum Einsatz, welche meiner Meinung nach eine einfache Installation ermöglicht und über einen guten Onlinesupport verfügt.

3.3.1. Linux

Die Installation von Linux mit der Distribution von SuSE gestaltet sich recht einfach. Es kann zu Beginn das typische Einsatzgebiet des Rechners ausgewählt werden und die benötigten Programme werden automatisch installiert. So kann zum Beispiel die Einstellung “Bürocomputer”

oder “Netzwerkserver” gewählt werden, welche die jeweils typischen Programme auf die Festplatte schreibt. Weiterhin wurden der Webserver Apache und die Skriptsprache PHP 4 automatisch eingerichtet. Lediglich die MySQL-Datenbank musste nachträglich hinzugefügt werden.

Eine solche Zusammenstellung wird LAMP-System genannt. LAMP ist die Abkürzung für Linux (als Betriebssystem), Apache (als Webserver), MySQL (als Datenbank) und PHP (als Programmiersprache). Diese Systeme sind unter Netzwerkcomputern sehr verbreitet und finden auch bei vielen großen Providern Anwendung. Eine Anleitung zur Einrichtung eines LAMP-Servers kann unter [6] nachgelesen werden.

3.3.2. Apache

Als Webserver wurde Apache [7] gewählt. Er läuft sowohl unter Linux als auch unter Windows, ist kostenlos zu verwenden und lässt sich bei Bedarf relativ einfach mit dem Administrationsprogramm Comanche [8] einrichten. Das Programm läuft sehr stabil und ist der mit 56% am häufigsten genutzte Webserver (Netcraft [9], Stand: Januar 2002). Zusätzliche Funktionen können als Modul eingebunden werden, wie zum Beispiel die Unterstützung von Secure Socket Layer (SSL) zum Verschlüsseln und sicheren Übertragen von Web-Seiten (mod_ssl). Standardmäßig dabei ist eine Funktion zur Authentifizierung. Bevor ein Benutzer die Webseiten mit seinem Browser abrufen kann, muss er sein Login und ein persönliches Passwort angeben. Außerdem wurden aus Sicherheitsgründen nur Zugriffe vom Netz der TU-Ilmenau gestattet. Das Konfigurationsskript dazu ist im Anhang A.1 auf Seite 49 zu finden.

3.3.3. MySQL

Da die anfallende Datenmenge der Messcomputer sehr umfangreich ist, ist es unumgänglich, mit einer professionellen Datenbank zu arbeiten. Hier wurde zugunsten von MySQL [10] entschieden, welche kostenfrei ist und sehr gut mit PHP zusammenarbeitet. Sie ist nicht auf Linux beschränkt, sondern kann auch unter anderen Betriebssystemen wie Windows genutzt werden. Administrieren lässt sie sich sehr gut mit dem Programm phpMyAdmin [11], welches über einen Webbrowser aufgerufen werden kann. Für einfache Abfragen, wie sie in unserem Fall vorkommen, liegt sie von der Geschwindigkeit her deutlich über dem Durchschnitt [12].

3.3.4. PHP: Hypertext Preprocessor

Da die Ergebnisse der Messdaten von jedem Computer aus mit beliebigen Betriebssystemen begutachtet werden sollen, fiel die Entscheidung auf eine Webschnittstelle, welche die Daten für einem Webbrowser mittels PHP [13] in Echtzeit generiert. So können von einem beliebigen

Software	Versionsnummer
Kernelversion von Linux	2.4.16
Webserver Apache	1.3.20
Datenbank MySQL	3.23.44-Max-log
Skriptsprache PHP	4.0.6
Administrationsprogramm Comanche	3.0 beta
Datenbankverwaltung phpMyAdmin	2.2.6
Grafikbibliothek JPGraph	1.6.1

Tabelle 3.3.: Softwareversionen des Web- und Datenbankservers

Rechner aus die Daten in einem Webbrowser angezeigt oder das Profil verschiedener Windkraftanlagen geändert werden.

PHP ist im Vergleich zu Perl eine recht junge Skriptsprache, doch erfreut sie sich auch dank der sehr guten Datenbankfunktionen immer größerer Beliebtheit. Einen guten Einstieg in die Programmierung bietet das Buch “PHP4 + MySQL” [14].

3.3.5. JPGraph

Um die Daten in einer Grafik übersichtlich zusammenzufassen und darzustellen, wurde die ausgezeichnete Klassenbibliothek JPGraph [15] mit eingebunden. Sie setzt auf die GD-Bibliothek [16] auf, welche umfangreiche Grafikfunktionen bereitstellt. Mit ihr können viele Varianten der Anzeige gewählt werden. Hier beschränken wir uns auf die Linienfunktionen, doch sind auch Kreis- oder Balkengrafiken möglich.

3.3.6. Zusammenfassung

Eine genaue Übersicht der wichtigsten Programme gibt Tabelle 3.3. Die nicht aufgeführten Programme können der Programmübersicht der SuSE Distribution 7.3 Professional entnommen werden.

3.4. Konfiguration eines sicheren Webservers

Ein Webserver kann ein großes Sicherheitsproblem darstellen. Auf der einen Seite soll er von überall erreichbar sein, auf der anderen aber gegen Angriffe so gut wie möglich geschützt werden.

In unserem Fall wurde eine Firewall auf dem Server installiert, die nur Zugriffe von außen über eine Secure Shell (ssh) und den Aufruf der Webseiten erlaubt. Der Webserver ist so konfiguriert worden, dass er nur Anfragen entgegennimmt, die aus dem Netzwerk der Universität

kommen. Weiterhin wurden alle erforderlichen Softwareupdates und Patches aufgespielt, um bekannt gewordene Sicherheitlücken zu schließen.

Eine umfangreiche Einleitung für die Konfiguration eines sicheren Webserver kann auf den Webseiten von SuSE [17] nachgelesen werden.

4. Entwicklung des Messbetriebs

4.1. Übersicht

Zum besseren Verständnis der Funktionsweise folgt zuerst eine Übersicht über die Programme, die Programmstrukturen und die Datenbankstrukturen.

4.1.1. Die Programme

Die einzelnen Aufgaben wurden auf eigene Programme verteilt. Eine Übersicht gibt Tabelle 4.1. Sie beinhaltet die Dateien, die einmal in der Nacht um 2.00 Uhr automatisiert durch das Programm *cron* aufgerufen werden. *Cron* dient zum regelmäßigen Ausführen und Anstoßen von Prozessen auf Linux-Systemen.

Das Programm *cron_job.php4* wird durch *cron* aufgerufen und steuert den Ablauf der Datenübertragung. Es liest aus der Datenbank alle Messrechner und deren Verbindungsdaten ein. Diese werden der Reihe nach an das Programm *pppconn.sh* übergeben, welches die Verbindung zu einem Messrechner herstellt. Nach dem erfolgreichen Download wird die Verbindung getrennt, die Datei als Backup gesichert, entpackt, von *convert2mysql.php4* konvertiert und in die Datenbank geschrieben.

Tabelle 4.2 auf der nächsten Seite gibt einen Überblick über die Dateien, welche der Benutzer mit seinem Webbrowser aufrufen kann. Sie erzeugen verschiedene Grafiken der Messdaten

Programm	Beschreibung
convert2mysql.php4	Konvertiert die Dateien der Messrechner und schreibt sie in die Datenbank.
cron_job.php4	Wird täglich ausgeführt. Steuert alles vom Verbindungsaufbau bis zum Eintrag der Daten in die Datenbank, indem es weitere benötigte Programme aufruft.
pppconn.sh	Shellskript zum Ansteuern der ISDN-Karte und initialisieren des Verbindungsaufbaus mit dem RAS-Server.

Tabelle 4.1.: Übersicht der täglich ausgeführten Programme

Programm	Beschreibung
index.php4	Einstiegsseite. Von hier geht es zur Datendarstellung, zur Profileingabe oder zur Log-Datei.
menugraf.php4	Stellt eine Eingabemaske zur Verfügung, mit welcher die Datenanzeige eingestellt werden kann.
showgraf.php4	Überprüft die Daten, die der Benutzer in *menugraf.php4* eingegeben hat und bindet die Grafiken ein, indem es weitere Programme aufruft.
graph.php4	Erzeugt die Temperaturkurven anhand der vorgegebenen Einstellungen aus den Messwerten.
diff_count.php4	Erzeugt eine Tabelle mit den Temperaturhüben.
count.php	Stellt die Häufigkeit der Messwerte in einer Grafik dar.
config.php4	Einstellungen und häufig benutzte Programmfunktionen.
profile.php4	Eingabemaske zum Erzeugen oder Verändern eines Profiles einer Messstation.
check_profile.php4	Überprüft die Eingaben von *profile.php4* und schreibt sie in die Datenbank.
diplom.log	Logdatei, welche die Aktivitäten der Programme in Tabelle 4.1 auf der vorherigen Seite mitprotokolliert und per Webschnittstelle eingesehen werden kann.
tu_header.jpg	Bilddatei mit dem Logo der Technischen Universität Ilmenau.

Tabelle 4.2.: Übersicht der Programme zur Datenvisualisierung und Administration

und stellen eine Maske zur Profileingabe der Messrechner zur Verfügung.

4.1.2. Die Programmstruktur

Wie im vorherigen Kapitel aufgelistet, gibt es Programme, die täglich aufgerufen werden, und Skripte, die erst beim Aufruf über einen Webbrowser ausgeführt werden. Um den Zusammenhang zwischen den Programmen zu verdeutlichen, zeigt Abbildung 4.1 auf der nächsten Seite die Verknüpfungen der Programme, welche alle 24 Stunden aufgerufen werden. Die Skripte, die über die Webschnittstelle ausgeführt werden können, zeigt Abbildung 4.2 auf Seite 26.

4.1.3. Die Datenbankstruktur

Das Designen einer Datenbank bedarf einiger Vorüberlegungen. Vor allem, wenn ein großer Datendurchsatz und eine einfache Skalierbarkeit des Systems wie in unserem Fall geplant ist, kommt es auf eine effiziente Aufteilung der Daten an. Dabei ist zu beachten, ob die Datenbank in ein relationales Datenbankschema und in eine Normalform überführt werden muss. Dies dient zur Gewährleistung einer redundanz- und widerspruchsfreien Informationsspeicherung.

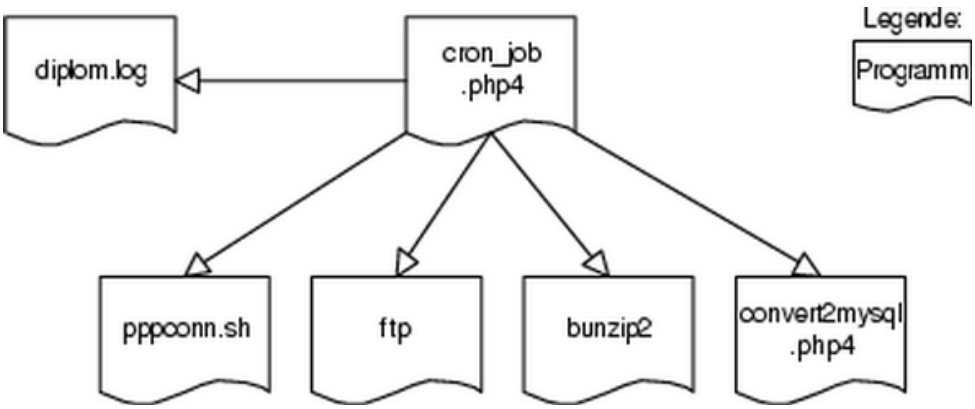

Abbildung 4.1.: Programmstruktur der täglich aufgerufenen Programme

Datenbankanomalien und Inkonsistenzen werden so vermieden. Durch den einfachen Aufbau der Tabellen sind beide Voraussetzungen gegeben.

Für unseren Zweck wurden zwei Tabellentypen eingeführt. Der erste Typ beinhaltet die Messdaten der Computer. Für jede Messstation wird dazu bei der Einrichtung in der Profilmaske über den Webbrowser automatisch eine eigene Tabelle eingerichtet. Abbildung 4.3 auf der nächsten Seite zeigt zwei Zeilen von Messwerten, die von einem Messrechner stammen. Sie wurden durch das Programm *convert2mysql.php4* konvertiert und in die Datenbank geschrieben.

Der zweite Tabellentyp beinhaltet die Informationen zum Verbindungsaufbau mit den Messrechnern. Hiervon gibt es nur eine zentrale Tabelle, sie heißt *admin*. Abbildung 4.4 auf Seite 27 zeigt eine Beispielkonfiguration zweier Messrechner. Diese Daten wurden in der Profilmaske über einen Webbrowser eingegeben und durch das Programm *check_profile.php4* in die Datenbank geschrieben.

Ein Skript zur Erstellung und Initialisierung der Tabelle *admin* ist im Anhang A.3 auf Seite 49 zu finden.

4.2. Abfrage der Daten der Messstation

Die Daten der Messrechner werden einmal am Tag abgeholt. Dazu wählt sich der Server der Reihe nach in die Messstationen ein und überprüft, ob Dateien für ihn zum Download bereitstehen. Ist dies der Fall, werden sie mit dem Programm *ftp* in ein temporäres Verzeichnis übertragen. Nach einer Überprüfung auf dem Server, ob der Download erfolgreich war, werden sie auf der Messstation gelöscht. Andernfalls erfolgt ein erneuter Versuch am nächsten Tag.

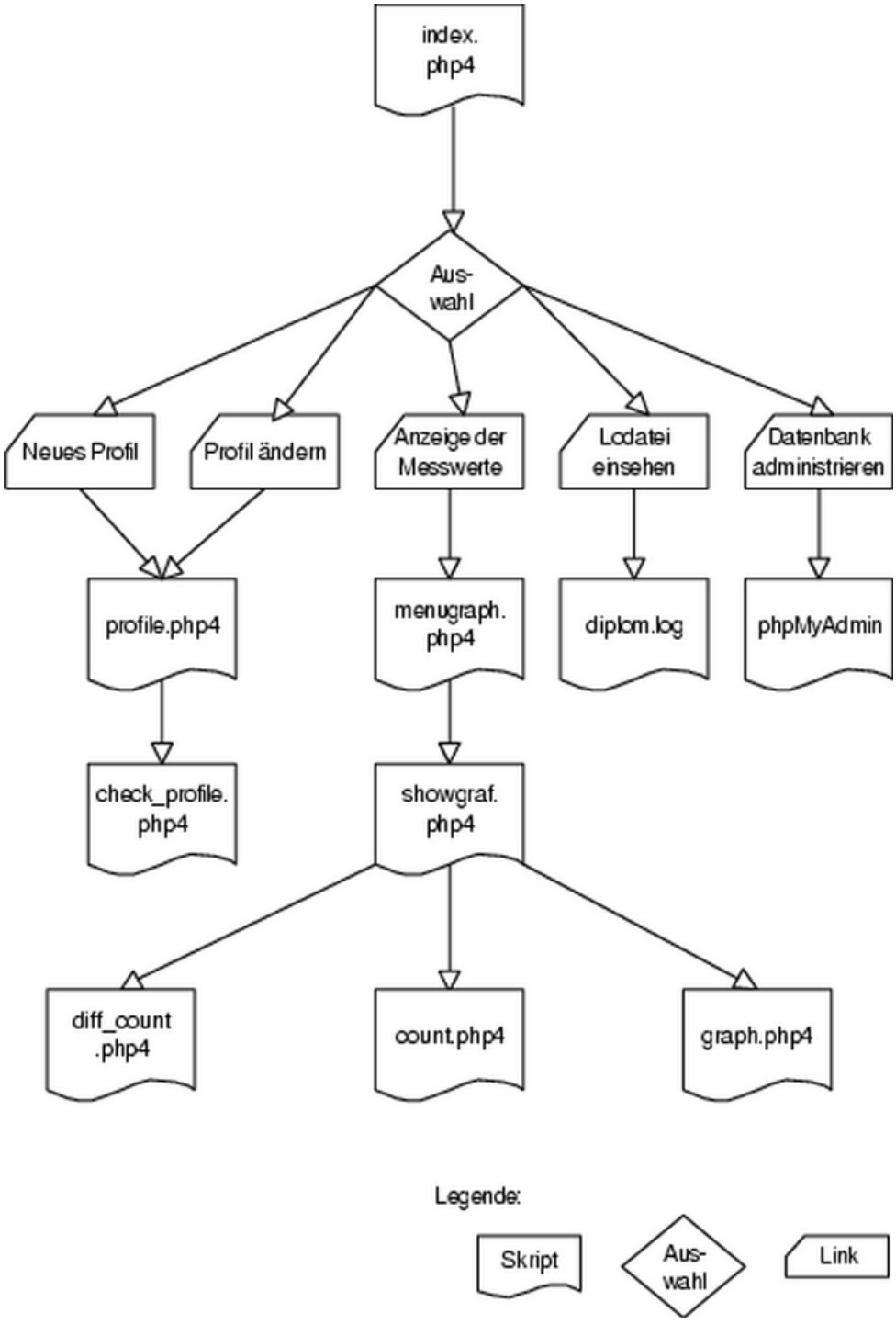

Abbildung 4.2.: Programmstruktur der Webseiten

id	timestamp	mictime	sensor1	sensor2	sensor3	sensor4	sensor5	sensor6
10191330660	2002-04-18 14:31:06	0	39	45	44	43	37	44
10191330661	2002-04-18 14:31:06	0.1	43	41	38	35	36	39

Abbildung 4.3.: Beispieldaten eines Messcomputers

name	computer	table_name	phone	protocol	user	passwd
Nordsee 1	mars-3	Platine 1	123456789	modem	gel-df1	gTEWjjW
Nordsee 5	fel-dl1	fel-dl1	125533	modem	test-login	ùZé^n

Abbildung 4.4.: Beispieldaten der Administrationstabelle

4.2.1. Definitionen der Schnittstellenparameter

Damit der Server mit den Daten der Messrechner umgehen kann, wurde eine Datenschnittstelle definiert. Sie beinhaltet die Art und Weise, wie der Messrechner die Daten in eine Datei ablegt. Der Server muss dabei fehlerhafte Datensätze erkennen.

Es wurde sehr viel Wert darauf gelegt, die Daten möglichst kompakt abzuspeichern, um die Übertragungszeit zwischen Messrechner und Server so gering wie möglich zu halten. Sie werden deshalb in binärer Form gesichert und zusätzlich vor der Übertragung mit einem Programm komprimiert.
Zeile 1 beinhaltet das Datum, die Startzeit im zehntel Sekundenbereich und die Größe des Zeitstempels in Bytes in folgendem Format:

```
2002-04-29 14:24:07.5 3
```

Zeile 2 beinhaltet den Anlagennamen. Dieser darf mehrmals vergeben werden.

```
Nordsee 1
```

Zeile 3 beinhaltet den Rechnernamen. Dieser darf nur einmal vergeben werden.

```
mars-3
```

Die darauffolgenden Zeilen sind ein beliebig langer Kommentarblock. Beginnen müssen die Zeilen mit einem doppelten Schrägstrich und abgeschlossen wird er komplett mit dem Hexadezimalwert 00h.

```
// Controlling Temperature
```

Die darauffolgenden Bytes, deren Länge in der ersten Zeile festgelegt wurde, beinhalten die Zeitdifferenz in zehntel Sekunden zum Zeitstempel in der ersten Zeile. Die Doppelpunkte dienen hier nur zur Übersichtlichkeit und werden in der Datei weggelassen.

```
00:00:01
```

Es folgen weitere sechs Bytes, ein Byte für jeden Messwert eines Sensors. Diese werden mit dem Stoppbyte 00h abgeschlossen und eine erneute Zeitdifferenz für die nächsten Messwerte kann geschrieben werden.

```
30:23:26:31:32:29:00
```

In der ersten Zeile kann auch eine feiner aufgelöste Startzeit stehen (z.B. 14:24:07.001). So ist es möglich, Werte z.B. im Millisekundenbereich abzufragen und zu übertragen. Entsprechend muss die Bytelänge jedes Zeitstempels, welche auch in der ersten Zeile festgelegt wird, vergrößert werden. Diese Option wurde nur für zukünftige Erweiterungen definiert. Im Fehlerfall könnte eine Datei mit hoher Abtastrate übertragen werden, um genauere Analysen zu ermöglichen. Da die Datenrate für eine tägliche Übertragung nicht geeignet ist, wurde dies noch nicht implementiert. Wir beschränken uns deshalb auf eine feste Abtastrate von zehn Werten pro Sekunde.

4.2.2. Verbindungsaufbau und Authentifizierung

Durch das täglich ausgeführte Programm *cron_job.php4* wird der Verbindungsaufbau mit den Messrechnern angestoßen. Zuerst werden die Daten des jeweiligen Messrechners von der Tabelle *admin* ausgelesen. Mit diesen Daten verbindet sich dann das Programm *pppconn.sh*. Es überprüft, ob eine ISDN-Verbindung vorhanden ist, oder ob ein analoges Modem softwaremäßig emuliert werden muss. Der Quellcode zu *cron_job.php4* befindet sich im Anhang A.4 auf Seite 50 und zu *pppcon.sh* im Anhang A.5 auf Seite 53.

Der Messrechner erkennt einen Anruf und weist das Modem an, abzuheben. Durch den RAS-Server, der auf jeder Messstation läuft, findet die Authentifizierung statt. Der anrufende Rechner muss sich mit einem Login und einem Passwort identifizieren. Damit wird verhindert, dass ein unberechtigter Zugriff auf die Messrechner stattfindet. Diese entnimmt er wiederum der Tabelle *admin*, in der vorher über den Webbrowser die korrekten Daten eingegeben worden sind.

Die Tabelle *admin* befindet sich wie die Datentabellen in der MySQL-Datenbank. Dort ist sie durch ein Zugangskennwort geschützt. Zusätzlich wird das Passwort für den Zugang zu den Messrechnern in der Tabelle *admin* verschlüsselt abspeichert. Der String zur Verschlüsselung kann in der Datei *config.php4* geändert werden. Diese ist im Anhang A.6 auf Seite 54 zu finden.

4.2.3. Datenübertragung

War der Verbindungsaufbau und die Authentifizierung des anrufenden Rechners erfolgreich, so überträgt das Programm *cron_job.php4* per FTP-Protokoll alle Dateien, die in dem Transferver-

zeichniss vorliegen. Ist der Download einer Datei vollständig gelungen, so wird sie anschließend auf dem Messrechner gelöscht. Somit wird verhindert, dass sie noch einmal übertragen wird.

Wird die Verbindung ungewollt unterbrochen, ist der Download nicht erfolgreich gewesen und die Datei wird auf dem Messrechner nicht gelöscht. Das Programm *cron_job.php4* versucht dann bei einem erneuten Verbindungsaufbau am nächsten Tag, die Datei noch einmal herunterzuladen.

Wurden alle Dateien eines Rechners heruntergeladen, wird die Verbindung unterbrochen. Durch ein anschließendes Kopieren in ein Back-Up Verzeichnis kann im Fehlerfall oder bei Datenverlust noch einmal darauf zurückgegriffen werden.

Die Meldungen der einzelnen Programme werden in *diplom.log* mitprotokolliert. Diese kann über einen Link auf der Hauptseite angesehen werden. Hat eine Verbindung nicht stattgefunden oder ist ein Fehler aufgetreten, kann dort die Ursache herausgefunden werden. Auch eine gelungene Konvertierung der Daten und deren Ablage in der Datenbank werden aufgezeichnet.

4.3. Aufbereitung der Messdaten

Nachdem die Dateien eines Rechners heruntergeladen wurden, werden sie mit dem Programm *bunzip2* entpackt.

4.3.1. Einlesen der Daten in die Tabellen

Das Einlesen der Daten in die Tabellen übernimmt das Programm *convert2mysql.php4*. Es überprüft zunächst, ob die Datei einen korrekten Header, wie in Kapitel 4.2.1 auf Seite 27 beschrieben, besitzt. Ist dies der Fall, werden die Daten, wenn die Stoppbits richtig gesetzt wurden, in die Tabelle des jeweiligen Messrechners geschrieben. Der Quellcode von *convert2mysql.php4* kann im Anhang A.7 auf Seite 55 nachgelesen werden.

Erst jetzt sind die Messwerte ohne Hilfsmittel zur Umwandlung der binären Zahlen mit korrektem Zeitstempel zu erkennen. Um dies zu überprüfen, befindet sich auf der Einstiegsseite des Projektes ein Link zum Programm *phpMyAdmin*, mit welchem komfortabel die Datenbank durchgesehen werden kann. Dies sollte allerdings nur zu Kontrollzwecken geschehen, eine manuelle Administration ist nur in Ausnahmefällen zu empfehlen.

4.4. Menüführung

Um die aufgenommenen Daten interpretieren zu können, wurde eine Webschnittstelle konzipiert. Der Benutzer hat die Möglichkeit, sich die Daten in unterschiedlichen Formaten und unter

verschiedenen Bedingungen darstellen zu lassen. Weiterhin wurde eine Menüführung zum Einrichten weiterer Messrechner und eine Oberfläche zur Administration der MySQL-Datenbank eingebunden.

4.4.1. Authentifizierung am Server

Damit die Daten nicht unbefugt eingesehen werden können, wurde das Verzeichnis, in dem sich die Webseiten befinden, gesperrt. Nur eine korrekte Eingabe eines Benutzernamens und des dazugehörigen Passwortes erlauben den Zugriff auf die Seiten. Die Konfigurationsskripte im Anhang A.1 auf Seite 49 und Anhang A.2 auf Seite 49 zeigen die Vorgehensweise eines Verzeichnisschutzes unter dem Webserver Apache.

4.4.2. Hauptmenü

Beim Aufrufen der Webadresse und nach erfolgreicher Authentifizierung befindet sich der Benutzer im Hauptmenü. Dort hat er die Möglichkeit, sich für einen Messrechner zu entscheiden und sich die Daten anzeigen zu lassen.

Weiterhin kann er die Messrechner verwalten, indem er z.B. die Telefonnummer oder das Passwort ändert, und neue hinzufügen. Ein weiterer Link führt zu einer Logdatei, in der sämtliche Aktionen des Servers festgehalten werden und die im Fehlerfall Aufschluss über die Ursache geben soll.

Ein Verweis leitet zur manuellen Datenbankadministration weiter. Da aus Sicherheitsgründen ein Verändern der Daten nicht vorgesehen ist, können nur über diese Oberfläche eine Tabelle und die Daten eines Messcomputers gelöscht werden.

Im Anhang A.8 auf Seite 60 ist der Quellcode von *index.php4* zur Erstellung des Hauptmenüs zu finden. Abbildung 4.5 auf der nächsten Seite zeigt einen Bildschirmausschnitt des Hauptmenüs.

4.4.3. Konfigurationsmenü

Das Konfigurationsmenü *profile.php4* dient zum Einbinden von neuen Messrechnern und deren Administration. So können z.B. die Zugangsdaten wie Telefonnummer und Passwort eingegeben oder geändert werden. Abbildung 4.6 zeigt einen Bildschirmausschnitt des Konfigurationsmenüs. Im Anhang A.9 auf Seite 62 kann der Quellcode eingesehen werden.

Nach der Eingabe wird mit dem Programm *check_profile.php4* überprüft, ob die Daten korrekt sind. Diese werden dann in die Datenbanktabelle *admin* übernommen. Wird ein neuer Messrechner eingebunden, wird gleichzeitig eine neue Datentabelle angelegt. Der Quellcode von *check_profile.php4* kann im Anhang A.10 auf Seite 66 nachgelesen werden.

Abbildung 4.5.: Bildschirmausschnitt der Webschnittstelle - Hauptmenü

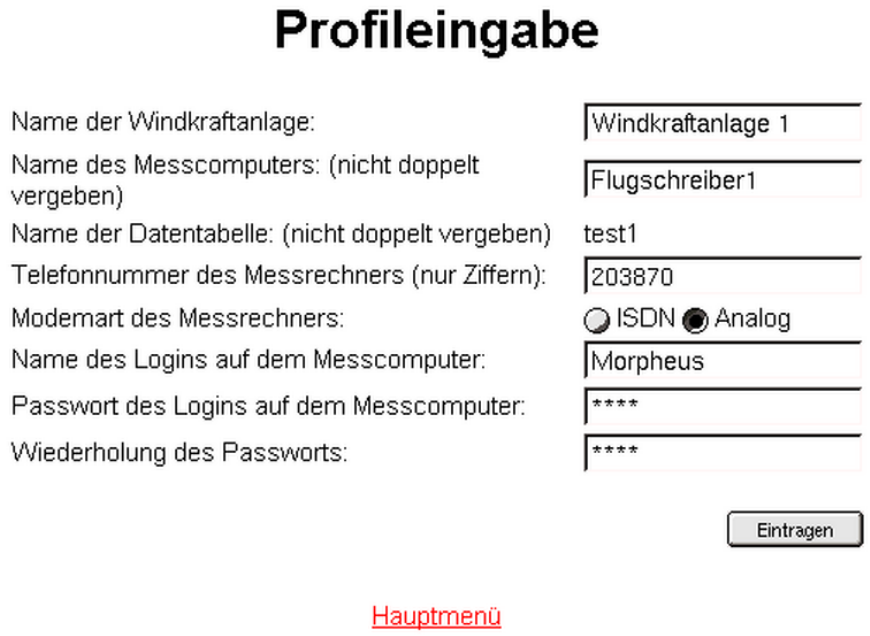

Abbildung 4.6.: Bildschirmausschnitt der Webschnittstelle - Konfiguration

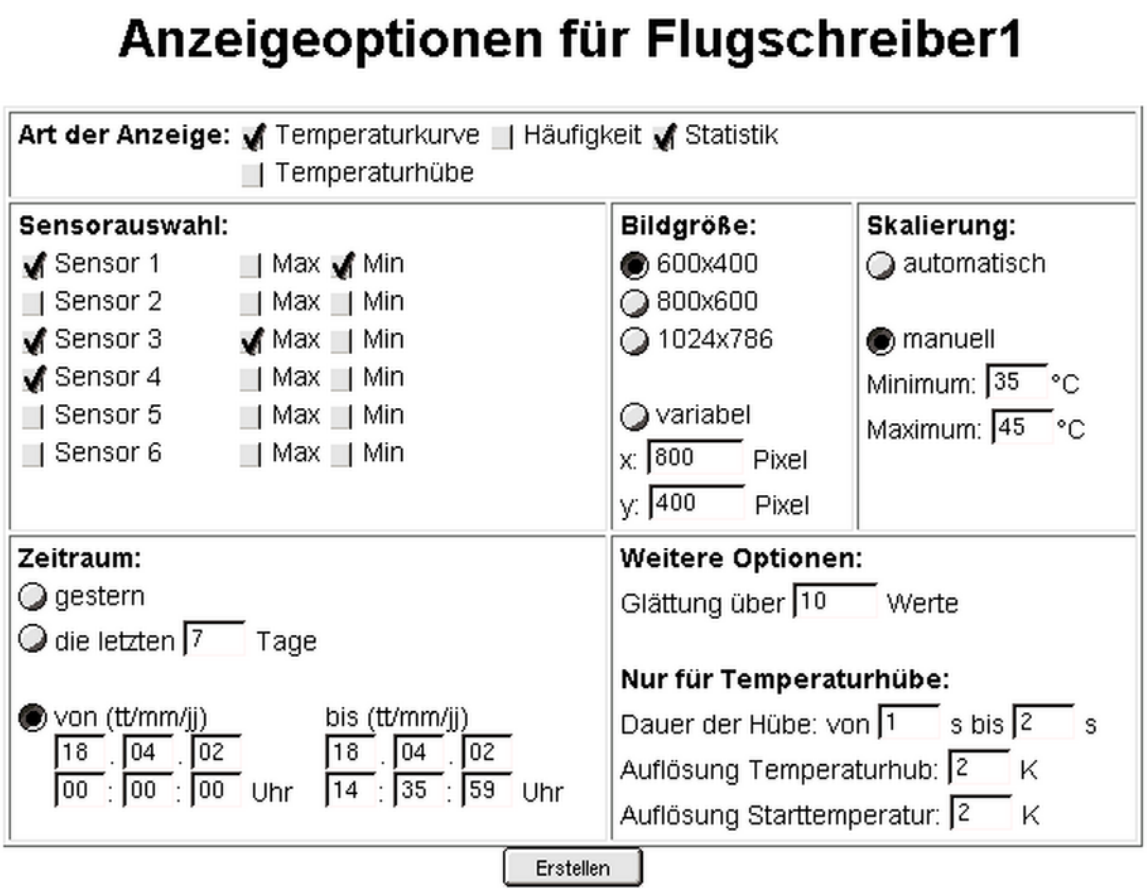

Abbildung 4.7.: Bildschirmausschnitt der Webschnittstelle - Datenauswahl

4.4.4. Auswahlmenü zur Darstellung der Daten

Nachdem im Hauptmenü ein Messrechner ausgewählt wurde, können dessen gemessene Daten betrachtet werden. Zuvor stellt jedoch das Programm *menugraf.php4* fest, welche Daten ausgewählt werden sollen. Unter zahlreichen Optionen lässt sich so z.B. der Zeitraum einstellen oder die Größe der auszugebenden Grafik festlegen. Abbildung 4.7 zeigt einen Bildschirmausschnitt des Auswahlmenüs. Im Anhang A.11 auf Seite 70 kann der kommentierte Quellcode eingesehen werden.

4.4.5. Anzeige der Messwerte

Nach der Auswahl des Datenbestandes und Eingrenzung mit *menugraf.php4* überprüft das Programm *showgraf.php4* die eingegebenen Daten. Danach werden die benötigten Messwerte mit dem angegebenen Glättungswert neu berechnet und in eine temporäre Tabelle geschrieben.

Das Glätten der Werte nimmt die meiste Rechenzeit in Anspruch. Sie ist aber unumgänglich, um hochfrequente Störungen auszufiltern. Zur Berechnung wird der Mittelwert von n bis n+Glättungsfaktor gebildet. Danach erfolgt die Berechnung des Mittelwertes von n+1 bis

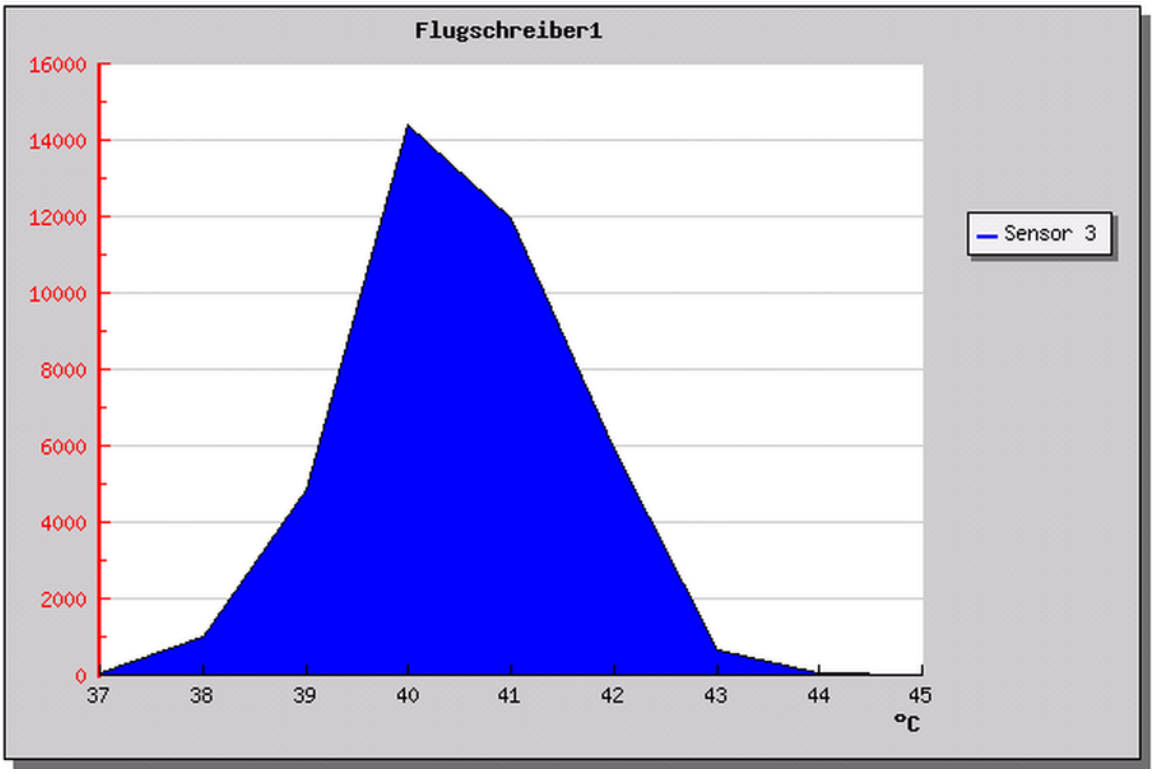

Abbildung 4.8.: Anzahl der gemessenen Werte

n+1+Glättungsfaktor und so weiter. Als Optimierung werden schon berechnete Additionen aus der vorherigen Mittelwertberechnung mit einbezogen.

Nachdem die temporäre Tabelle fertiggestellt ist, werden die Daten mit der Eingrenzung je nach Auswahl des Benutzers an die Programme zur Bilderzeugung weitergeleitet.

Es gibt vier verschiedene Darstellungsmöglichkeiten. Die erste Darstellungsmöglichkeit berechnet eine Grafik mit der Anzahl der gemessenen Werte über den angegeben Zeitraum. Abbildung 4.8 zeigt eine Auswahl mit Beispielwerten.

Die zweite zeigt die Temperaturschwankungen der einzelnen Sensoren. Zusätzlich kann der Minimalwert und der Maximalwert in die Grafik mit eingebunden werden. Einen Bildschirmausschnitt einer Sinuskurve mit hochfrequenter Überlagerung ohne Glättung zeigt Abbildung 4.9 auf der nächsten Seite, die gleichen Messwerte mit einer Glättung über 30 Werte zeigt Abbildung 4.10 auf der nächsten Seite.

Als dritte Darstellungsmöglichkeit können die Temperaturhübe angezeigt werden. Je nach Auswahl in *menugraf.php4* werden schnelle oder langsame Hübe ausgegeben. Somit können kurz- oder langfristige Temperaturschwankungen herausgefunden werden. Abbildung 4.11 auf Seite 35 zeigt einen Ausschnitt aus der Darstellung.

Als letzte Möglichkeit kann noch eine Statistik über die gesammelten Werte ausgegeben werden. Dies wird in Abbildung 4.12 auf Seite 35 verdeutlicht.

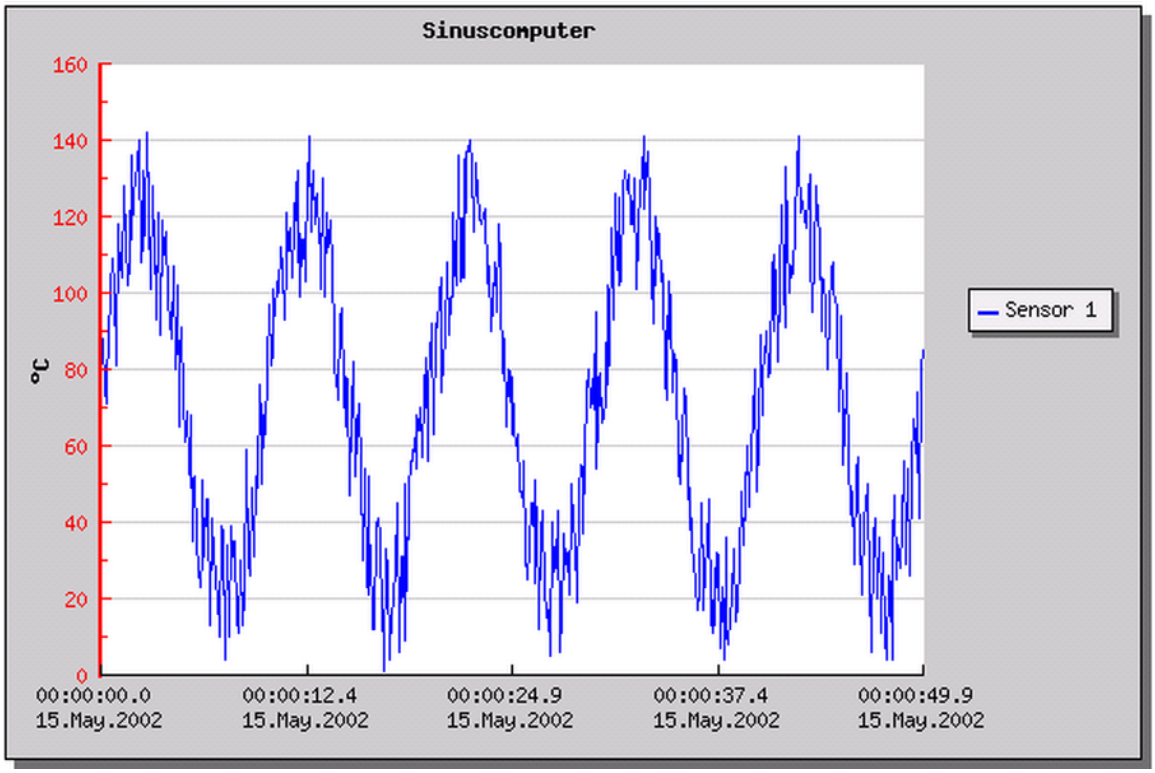

Abbildung 4.9.: Temperaturkurve ohne Glättung

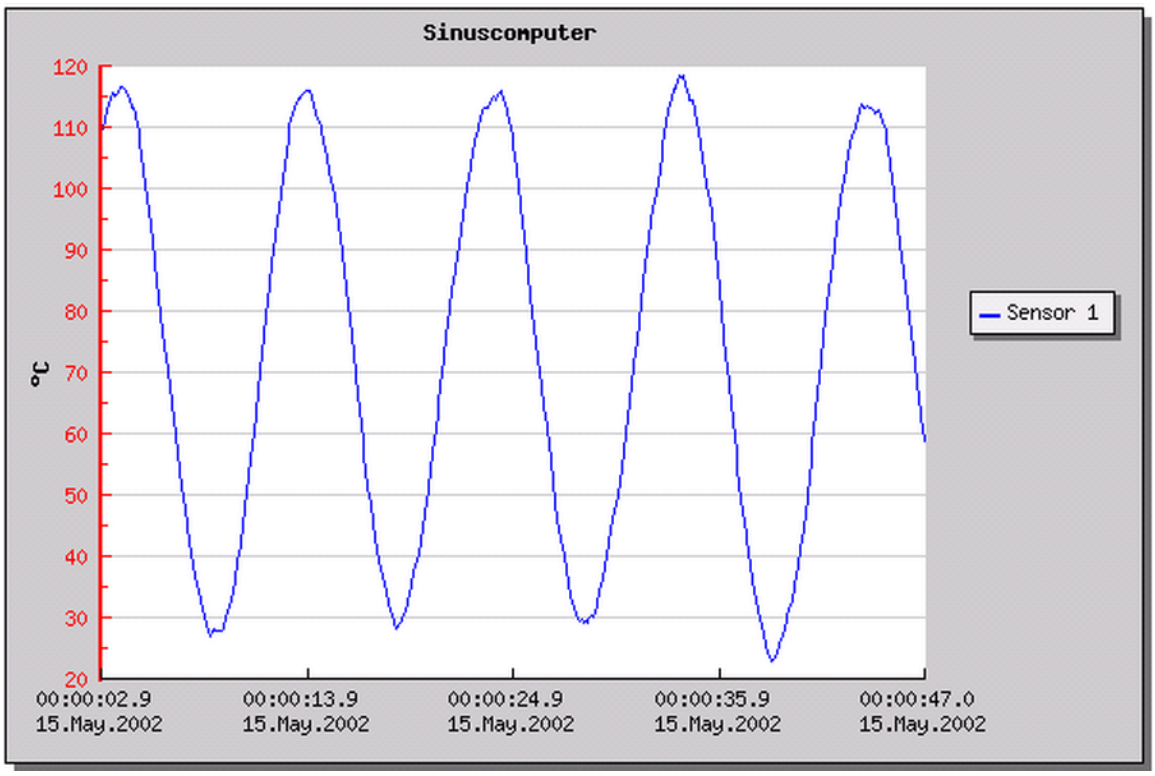

Abbildung 4.10.: Temperaturkurve mit Glättung über 30 Messwerte

Positive Temperaturhübe für Flugschreiber1, Sensor 3
Hübe zwischen 1 und 5 Sekunden

Starttemperatur in °C	37	38	39	40	41
Temperaturhub in K					
1	-	2	14	41	18
2	-	10	110	107	6
3	1	47	145	46	2
4	4	51	57	6	-
5	6	15	5	1	-
6	2	3	-	-	-

Negative Temperaturhübe für Flugschreiber1, Sensor 3
Hübe zwischen 1 und 5 Sekunden

Starttemperatur in °C	40	41	42	43	44	45
Temperaturhub in K						
-1	8	37	25	2	-	-
-2	7	83	99	11	-	-
-3	3	55	140	39	-	-
-4	-	5	45	57	3	-
-5	-	-	3	19	10	1
-6	-	-	-	-	1	-

Abbildung 4.11.: Ausgabe der Temperaturhübe

Statistik

Erster Zeitstempel: 2002-04-18 14:31:06
Letzter Zeitstempel: 2002-04-18 15:35:59
Anzahl der Werte pro Sensor: 38874
Glättung über 10 Werte

	Sensor 1	Sensor 3	Sensor 4
Minimalwert	36.70	36.70	36.60
Maximalwert	44.20	44.60	44.20
Durchschnitt	40.511658	40.489926	40.496658

Abbildung 4.12.: Ansicht der Statistikausgabe

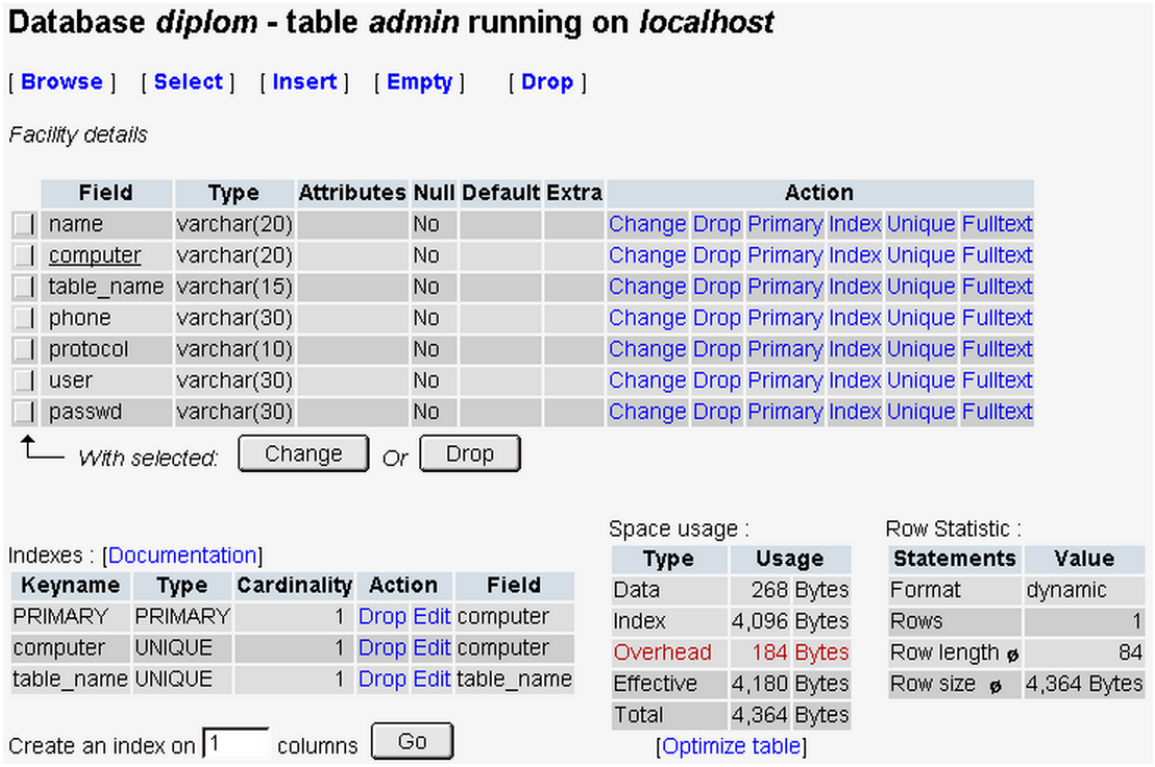

Abbildung 4.13.: Bildschirmausschnitt von *phpMyAdmin*

Die Quellcodes können im Anhang ab A.12 auf Seite 77 nachgelesen werden.

4.4.6. Administration der Datenbank

Die Administration der Datenbank erfolgt durch das Programm *phpMyAdmin*. Es ist ein im Quellcode vorliegendes PHP-Programm, mit dem komfortabel MySQL-Datenbestände konfiguriert und verändert werden können. In unserem Fall dient es nur zur Kontrolle der Datenbestände und zum Löschen von Datensätzen nicht mehr benutzter Messcomputer.

Die Abbildung 4.13 zeigt einen Bildschirmausschnitt von *phpMyAdmin*.

4.5. Übersicht möglicher Datenmengen

Bei der Entwicklung des Projektes stellte sich heraus, dass mit großen Datenmengen bei der Messaufnahme zu rechnen ist. Da aber erst nach einer gewissen Laufzeit des Systems abgeschätzt werden kann, welche Daten benötigt werden und welche nicht, müssen alle Aufzeichnungen übertragen werden.

Die Sensoren werden zehnmal in der Sekunde abgefragt. Mit dem Zeitstempel und ohne der vernachlässigbaren Implementierung eines Headers wie in Kapitel 4.2.1 auf Seite 27 beschrieben, ergibt sich folgende Gleichung:

Programm	Datenmenge nach Komprimierung	Rate
zip	5.195.719 Bytes	1,663
bzip2	4.858.792 Bytes	1,778
lha	5.106.849 Bytes	1,692

Tabelle 4.3.: Vergleich von Komprimierungsprogrammen

(3+6+1)*10*60*60*24 Bytes/Tag = 8.640.000 Bytes/Tag

Durch geeignete Kompression kann die zu übertragene Datenmenge erheblich reduziert werden. Dies ist allerdings abhängig von der Art der Daten. Werden überwiegend kleine Temperaturschwankungen gemessen, kann besser komprimiert werden. Bei dem Vergleich in Tabelle 4.3 wurde der schlechteste Fall angenommen, alle Werte sind zufällig erzeugt worden und unterliegen großen Schwankungen.

Da *bzip2* den höchsten Komprimierungsfaktor aufweist, wurde dieses Programm zur Komprimierung verwendet.

4.6. Sicherheit bei der Datenübertragung

Bei der Datenübertragung war zu beachten, dass die Daten nicht von unbefugten Personen eingesehen werden können. Durch die Wahl der Telefonleitung als Übertragungsmedium konnte dies so gut wie ausgeschlossen werden. Eine Telefonleitung abzuhören gestaltet sich im Normalfall deutlich schwieriger, als im Internet Daten abzufangen. Weiterhin muss sich der Benutzer an der Messstation durch einen Benutzernamen und ein Passwort authentifizieren.

Bei erhöhten Sicherheitsansprüchen kann hier auf eine Verschlüsselungsmethode wie Pretty Good Privacy (PGP) gesetzt werden.

4.7. Mögliche gravierende Fehlerfälle

Um Fehlfunktionen vorzubeugen, welche einen Datenverlust mit sich bringen können, müssen mögliche Fehlerquellen im Voraus erörtert werden. Dadurch kann durch geeignete Erkennung und Programmierung versucht werden, Betriebsstörungen weitestgehend zu verhindern.

4.7.1. Ausfall der Messstation

Der wohl gravierendste Störfall ist der Ausfall der Messstation, da dort kontinuierlich Datenströme anfallen und bearbeitet werden müssen. Dies kann durch einen Stromausfall, durch Hard- und Softwarefehler sowie mechanisch bedingte Alterserscheinungen hervorgerufen werden.

Kurzfristige Stromunterbrechungen bzw. Spannungsänderungen können durch eine USV (Unterbrechungsfreie Stromversorgung) überbrückt werden.

Tritt ein Softwarefehler auf und die Messstation stürzt ab, kann sie im Normalfall nur manuell neu gebootet werden. Hier hilft eine Watchdog-Karte, die diesen Fehlerfall erkennt und versucht, den Rechner hardwaremäßig neu zu booten. Ist der Computer in Betrieb, werden in regelmäßigen Abständen durch eine Software Impulse an die Karte gesendet. Bleiben diese aus, ist anzunehmen, dass der Computer abgestürzt ist. Dieser wird nach einer gewissen Zeitspanne durch die Karte neu gestartet.

Hardwarefehler sind schwieriger zu erkennen und zu beheben. Es kann durch Redundanz von Komponenten wie beispielsweise die Spiegelung von Festplatten eine gewisse Sicherheit erreicht werden. Im Fehlerfall springt das noch funktionierende Modul für das defekte ein und dieses kann ohne Datenverlust ersetzt werden.

4.7.2. Ausfall von Sensoren

Es besteht die Möglichkeit, dass einzelne Sensoren ausfallen. Dies kann durch Vergleich von zwei Sensorwerten, die mit verschiedenen Sensoren an einem Punkt gemessen werden und unterschiedliche Werte anzeigen, herausgefunden werden. Auch bei geeigneter Interpretierung der Messdaten ist eine Fehlfunktion festzustellen.

Das Problem könnte durch eine Fernadministration gelöst werden, bei der die Daten des noch funktionierenden Sensors die des defekten ersetzen. Voraussetzung ist allerdings die doppelte Abnahme der Messwerte.

4.7.3. Kurzzeitige Unterbrechungen der Aufzeichnung

Wie in Kapitel 4.7.1 auf der vorherigen Seite beschrieben, kann durch verschiedene Umstände der Datenstrom abreißen und erst nach gewisser Zeit wieder aufgenommen werden. Beim Neustart der Messstation wird überprüft, ob eine abgebrochene Datei mit Daten vorliegt. Ist dies der Fall, wird sie zur Übertragung in das Transferverzeichnis kopiert und eine neue Datei mit der aktuellen Startzeit im Header angelegt. Der Server hat die Aufgabe, beide Dateien herunterzuladen (im Extremfall auch mehr), diese zusammenzusetzen und mit richtigem Zeitstempel in die Datenbank zu schreiben.

4.7.4. Abbruch der Verbindung bei der Datenübertragung

Im Betrieb kann es vorkommen, dass keine Verbindung vom Server mit der Messstation zu bekommen ist oder diese nach gewisser Zeit abreißt. Hier wurden serverseitig mehrere Vorkehrungen getroffen.

Im Messrechner existiert ein temporäres Verzeichnis, in dem die Messdaten der Sensoren stehen. Alle Dateien, die sich dort befinden, werden bei einer Verbindung mit dem Server heruntergeladen, und nach einer Überprüfung im Transferverzeichnis auf der Messstation gelöscht. Tritt ein Fehler auf, so lädt der Server die Datei mit dem nächsten Downloadzyklus erneut herunter. Somit werden unterbrochene Verbindungen zuverlässig erkannt.

4.7.5. Ausfall des Servers

Auch der Server kann, wie die Messstation, ausfallen. Dies ist jedoch nicht so schwerwiegend, da die Messrechner ihre Daten solange halten, bis sie abgerufen werden. Außerdem hat der Server keine zeitkritischen Daten zu verarbeiten. Zur Sicherheit hängt er trotzdem an einer USV.

5. Fernwartung der Messstationen

Eine weitere wichtige Aufgabe des Servers ist die Fernwartung der Messstationen. Es gibt inzwischen mehrere Programme, mit denen dies verwirklicht werden kann. Auf einem lokalen Computer kann der Bildschirminhalt des zu administrierenden Computers dargestellt werden. Der Benutzer hat den Eindruck, als ob er direkt am entfernten Rechner arbeiten würde. Tabelle 5.1 gibt eine Übersicht der gebräuchlichsten Tools.

5.1. Vorüberlegungen

In unserem Projekt haben wir uns für das Fernadministrationsprogramm Virtual Network Computing (VNC) [18] von AT&T entschieden. Es ist kostenfrei und unterstützt die gebräuchlichsten Plattformen. So kann beispielweise ein Macintosh einen Windows-Computer administrieren oder, wie in unserem Fall, ein Linuxsystem einen NT-Rechner. Abbildung 5.1 auf der nächsten Seite zeigt einen Bildschirmausschnitt während des Betriebs der Messstation.

Um allerdings Daten von einem anderen Computer zu übertragen, ist VNC nicht geeignet. Da aber eine Verbindung mit Internetprotokollen besteht, kann das File Transfer Protocol (FTP) verwendet werden. Mit ihm ist es möglich, Programme auf den entfernten Rechner zu laden (z.B. ein Systemupdate) oder Programme auf den lokalen Rechner zu holen.

Zu beachten ist auch die Geschwindigkeit der Datenübertragung. Während bei einer Verbindung über Netzwerkkarten fast kein Unterschied zu bemerken ist, treten bei Verbindungen über das Telefonnetz doch beträchtliche Geschwindigkeitseinbußen auf. Eine ISDN-Karte bringt noch gute Ergebnisse bei der Übertragung der Bildschirmdaten. Wird allerdings die Verbindung

Programmname	Internetadresse
pcAnywhere	`http://www.symantec.com/pcanywhere/Consumer/`
VNC	`http://www.uk.research.att.com/vnc/index.html`
x0rfbserver	`http://www.hexonet.de/software/x0rfbserver/`
ssh	`http://www.openssh.com/`

Tabelle 5.1.: Übersicht der gebräuchlichsten Programme zur Fernadministration

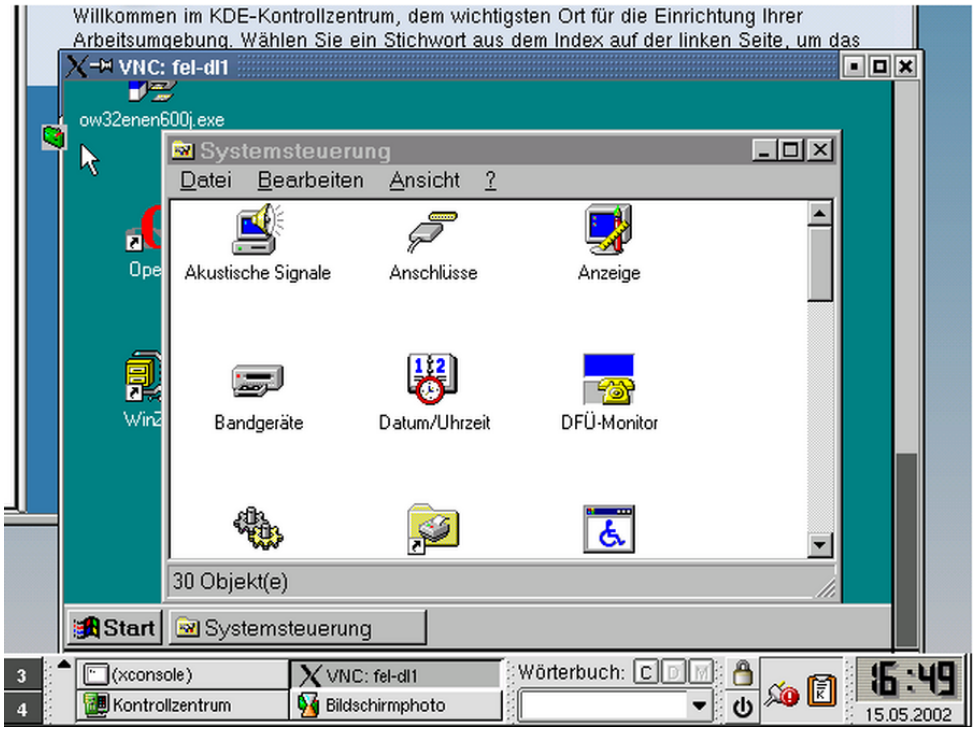

Abbildung 5.1.: Bildschirmausschnitt einer VNC-Verbindung von Linux nach Windows NT

über ein analoges Modem aufgebaut, dauert es recht lange, bis ein komplettes Bild übertragen ist. Da jedoch immer nur der Bildschirmausschnitt mit den Änderungen neu übertragen wird, ist eine Fernadministration noch möglich.

5.2. Funktionsweise

Auf der Messstation muss ein VNC Server eingerichtet sein. Auf diesen kann dann mittels des VNC-Clients eines beliebigen Betriebssystems zugegriffen werden.

Zuerst wird eine Verbindung zur Messstation hergestellt. In unserem Fall wird dafür für jeden Computer bei der Profileingabe ein Icon auf dem Desktop erstellt. Dadurch kann eine manuelle Verbindung mit einem bestimmten Messrechner hergestellt werden. Danach wird das Clientprogramm zur Fernadministration gestartet und nach der Angabe des Passwortes kann auf dem lokalen Rechner so gearbeitet werden, als ob man sich vor dem entfernten befinden würde. Abbildung 5.2 auf der nächsten Seite gibt einen Überblick der Funktionsweise.

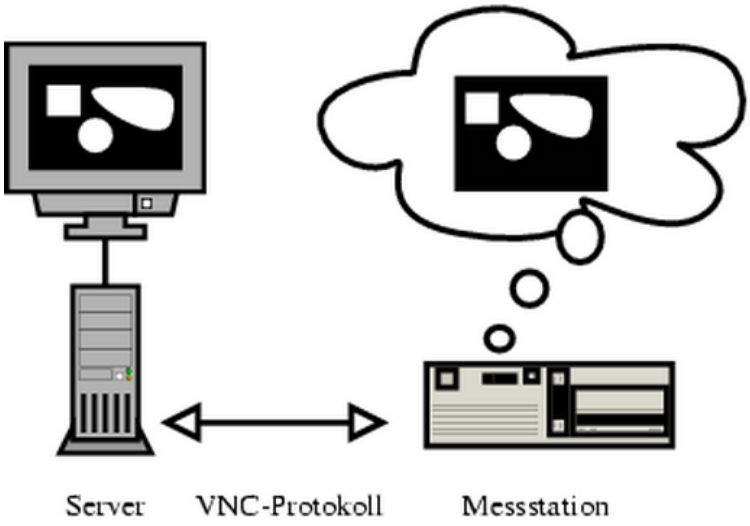

Abbildung 5.2.: Funktionsweise der Fernadministration

6. Zusammenfassung

Ziel dieser Diplomarbeit war es, ein Konzept zur Übertragung und anschließenden grafischen Aufbereitung von gemessenen Betriebsgrößen leistungselektronischer Bauteile zu erstellen. Diese können sodann zur Bestimmung der zu erwartenden Lebensdauer verwendet werden. Dabei wurde darauf Wert gelegt, die Messdaten wirtschaftlich zu übertragen und das System kostengünstig und stabil zu konzipieren. Weiterhin sollten entfernte Messrechner durch eine Fernadministration gewartet werden können.

Durch eine sorgfältige Auswahl der Hard- und Softwarekomponenten wurde ein Messrechner und ein Server als Prototypen für den Betrieb in einer Windkraftanlage zusammengestellt. Weiterhin wurden die Folgen eines Ausfalles der Computer betrachtet und Lösungsvorschläge zur Vermeidung erörtert.

Zugunsten der Sicherheit und Flexibilität wurde als Übertragungsmedium das Telefonnetz und optional das Mobilfunknetz gewählt. Untersuchungen haben dabei ergeben, dass die Messwerte bei der hier entwickelten Form der Ablage von Daten in einer Datei und anschließender Komprimierung durch eine Telefonleitung täglich heruntergeladen und in eine Datenbank geschrieben werden können. Die Daten könnten sich noch weiter komprimieren lassen, wenn geringe Temperaturschwankungen auf den Messrechnern nicht aufgezeichnet werden. Dies kann aber erst nach einer Probelaufzeit geändert werden.

Das größere Problem stellt der Server dar, der die Messwerte in einer Datenbank ablegt und über die Webschnittstelle die hier entwickelten Grafikarten zur Interpretation der Daten erstellt. Er muss mit sehr vielen Daten umgehen können. In unserem Fall werden Mittelwerte der Daten benötigt, um kurzfrequente Störungen herauszufiltern. Damit erhöht sich der Rechenaufwand je nach Glättungsfaktor erheblich. So werden für die Berechnung der Grafik mit Daten von einer Woche unter Umständen mehrere Minuten benötigt. Da das System aber in einer Probezeit sehr variabel gehalten werden muss, ist dieses Zeitproblem nicht zu umgehen.

Ist nach einer gewissen Testlaufzeit der benötigte Glättungsfaktor bekannt, können die Berechnungen sofort nach dem Empfang der Daten nachts durchgeführt werden. Damit wäre eine Anzeige der Grafiken mit vielen Messwerten deutlich schneller.

Eine Fernwartung von Computersystemen über das Internet oder eine Telefonleitung ist

durch eine große Anzahl von verfügbaren Programmen einfach zu bewerkstelligen. So kann ein Softwareupdate aufgespielt oder die Erfassung der Messwerte geändert werden.

Systeme zur Datenübertragung und Auswertung werden in Zukunft einen noch größeren Stellenwert einnehmen. Auf der einen Seite wird die zunehmende Vernetzung von Geräten eine einfache Datenübertragung ermöglichen, auf der anderen Seite können durch Bauteile, deren zu erwartende Lebensdauer bestimmt wurde, erhebliche Folgekosten durch Garantieansprüche eingespart werden.

Literaturverzeichnis

[1] Installation eines FTP-Servers unter NT http://www.tu-berlin.de/zrz/information/nt/ftp-install/Default.htm

[2] Einrichten eines RAS http://www.aspenres.com/Website/comstock/help/nethelp/netInstalling_RAS_Server_on_NT_40.html

[3] HP-VEE http://www.hp.com

[4] Verkürzung der Entwicklungszeiten durch HP-VEE http://www.somatec.com/vee.htm

[5] Die SuSE Homepage http://www.suse.de

[6] Behme, Hennig; Aller Anfang, Apache für Linux mit PHP und Perl konfigurieren, IX, Heft 6, Juni 2000, Seite 48

[7] Die Apache Homepage http://www.apache.org

[8] Die Comanche Homepage http://www.covalent.net/projects/comanche/

[9] Die Netcraft Homepage http://www.netcraft.com/survey/

[10] Die MySQL Homepage http://www.mysql.com

[11] Die phpMyAdmin Homepage http://www.phpwizard.net/projects/phpMyAdmin/

[12] Performancevergleich von Datenbanken http://www.mysql.com/information/benchmarks.html

[13] Homepage PHP http://www.php.net

[14] Stoll, Rolf. D; Leierer, Gudrun Anna: PHP4 + MySQL, DATA Becker, 2000, 1. Auflage

[15] Homepage JPGraph http://www.aditus.nu/jpgraph/

[16] Homepage GD-Bibliothek http://www.boutell.com/gd/

[17] Konfiguration eines sicheren Webservers http://www.suse.de/de/support/howto/secure_webserv/index.html

[18] Programm zur Fernadministration http://www.uk.research.att.com/vnc/

A. Quellcode

A.1. .htaccess

```
order allow,deny
deny from all
allow from 141.24.
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /home/schtorch/public_html/diplom/.htpasswd
require user testuser
```

A.2. .htpasswd

```
testuser:1234567890
```

A.3. admin.sql

```
#
# Tabellenstruktur für Tabelle ‘admin‘
#

CREATE TABLE admin (
  name varchar(20) NOT NULL default ’’,
  computer varchar(20) NOT NULL default ’’,
  table_name varchar(15) NOT NULL default ’’,
  phone varchar(30) NOT NULL default ’’,
  protocol varchar(10) NOT NULL default ’’,
  user varchar(30) NOT NULL default ’’,
  passwd varchar(30) NOT NULL default ’’,
```

```
  PRIMARY KEY  (computer),
  UNIQUE KEY computer (computer),
  UNIQUE KEY table_name (table_name)
) TYPE=MyISAM COMMENT='Facility details';
```

A.4. cron_job.php4

```
<?php
  /*****************************************
  /* Programm cron_job.php4
  /* Cron-skript to download data from computer,
  /* extract it and save it into database
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  // Timeout to ftp-connection is 90 sec
  error_reporting(63); // 0 = none, 63 = full
  include("/home/schtorch/public_html/diplom/config.php4");
  include("./convert2mysql.php4");
  // Get computer details from database
  $db = mysql_connect($hostname, $username, $password);
  mysql_select_db($database, $db);
  $result = mysql_query("SELECT *,
    DECODE(passwd,\"$secret_key\") FROM 'admin'", $db);
  $i = 0;
  while ($row = mysql_fetch_row($result)) {
    echo mysql_error($db);
    $name[$i] = $row[0];
    $computer[$i] = $row[1];
    $table[$i] = $row[2];
    $phone[$i] = $row[3];
    $protocol[$i] = $row[4];
    $user[$i] = $row[5];
    $passwd[$i] = $row[7];
    $i++;
```

```
}
$fp_log =
  fopen("/home/schtorch/public_html/diplom/diplom.log",
  "a+");
for ($i = 0; $i < sizeof($name); $i++) {
  // connect to computer
  fputs($fp_log, date('Y-m-d H:i:s')." - Anwahl von
    ".$computer[$i]." mit ".$phone[$i]."
    ".$protocol[$i]." ".$user[$i]."\n");
  system("/home/schtorch/diplom/mikehup1 pppconn.sh
    $phone[$i] $protocol[$i] $user[$i] $passwd[$i] &");
  echo "Anwahl\n";
  $a = time()+20;
  $b = time();
  while ($b < $a) {
    $b = time();
  }
  // check if connection exists
  $errstr = exec("ps --no-heading -C capipppd");
  if ($errstr != "") {
    fputs($fp_log, date('Y-m-d H:i:s')." - Verbindung zu
      ".$computer[$i]." besteht\n");
    // download via ftp
    $conn_id = 0;
    $conn_id = ftp_connect("192.168.14.10");
    // login with username and password
    $login_result = ftp_login($conn_id, $user[$i],
      $passwd[$i]);
    // check connection
    if ((!$conn_id) || (!$login_result)) {
      fputs($fp_log, date('Y-m-d H:i:s')." -
        FTP-Verbindung zu ".$computer[$i]." NICHT
        erfolgreich\n");
      break;
    } else {
      fputs($fp_log, date('Y-m-d H:i:s')." -
        FTP-Verbindung zu ".$computer[$i]."
```

```
    erfolgreich\n");
}
// download the files
$files = 0;
$list = ftp_nlist($conn_id, "*.bz2");
if (isset($list[0])) {
  do {
    $download = ftp_get($conn_id,
      "backup/".$list[$files], $list[$files],
      FTP_BINARY);
    // check download status
    if (!$download) {
      fputs($fp_log, date('Y-m-d H:i:s')." -
        File-Download $list[$files] von
        ".$computer[$i]." NICHT erfolgreich\n");
    } else {
      fputs($fp_log, date('Y-m-d H:i:s')." -
        File-Download $list[$files] von
        ".$computer[$i]." erfolgreich\n");
      ftp_delete($conn_id, $list[$files]);
    }
    $files++;
  }
  while($files < sizeof($list));
}
else
fputs($fp_log, date('Y-m-d H:i:s')." - Keine Dateien
  auf ".$computer[$i]." zum herunterladen\n");
// close the FTP stream
ftp_quit($conn_id);
// close connection
system("killall -HUP pppd");
fputs($fp_log, date('Y-m-d H:i:s')." - Verbindung
  beendet\n");
// unzip data
shell_exec("cd /home/schtorch/diplom/backup &&
  bunzip2 -k *.bz2");
```

```
      fputs($fp_log, date('Y-m-d H:i:s')." - Entpacken
        erfolgreich beendet\n");
      // backup all .bz files
      shell_exec("cd /home/schtorch/diplom/backup && mv
        *.bz2 /home/schtorch/diplom/safe");
      // convert and save data into database
      $cmd = "cd /home/schtorch/diplom/backup && /bin/ls
        *.lst";
      $fp_ls = popen($cmd, "r");
      while ($file_name = trim(fgets($fp_ls, 100))) {
        $erg =
          convertFile("/home/schtorch/diplom/backup/$file_n
          ame");
        fputs($fp_log, date('Y-m-d H:i:s')." -
          convert2mysql.php4: ".$erg);
        unlink("/home/schtorch/diplom/backup/$file_name");
      }
    }
    else
    fputs($fp_log, date('Y-m-d H:i:s')." - Verbindung zu
      ".$computer[$i]." konnte nicht hergestellt werden\n");
    fputs($fp_log, date('Y-m-d H:i:s')." - Ende der
      Skriptabarbeitung für ".$computer[$i]."\n\n");
    echo "Ende $computer[$i]\n";
  }
  fclose($fp_log);
?>
```

A.5. pppconn.sh

```
#!/bin/sh -x
# Script to connect to remote computer
# (C) 2002, Jens Bierkandt
# Change the local Telfonnumber (MSN) below
MSN=1697
# Nothing needs to be edited below
```

```
BASENAME=${0##*/}
NUMBER=$1
PROTOCOL=$2
USER=$3
PASSWORD=$4
if [ -z "$PASSWORD" ]; then
echo "Usage: $0 <NUMBER> <PROTOCOL> <USER> <PASSWORD>"
exit 1
fi
exec /home/schtorch/diplom/capipppd \
--controller 1 \
--number $NUMBER \
--msn $MSN \
--protocol $PROTOCOL \
-- \
lock \
noauth \
-chap \
ipcp-accept-local \
ipcp-accept-remote \
debug \
idle 30 \
plugin userpass.so \
replacedefaultroute \
user $USER \
password $PASSWORD
```

A.6. config.php4

```
<?php
  /*****************************************
  /* Programm config.php4
  /* global variables
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
```

```
  /*****************************************
  */
  error_reporting(63); // 0 = none, 63 = full
  //DB-Information
  //insert your DB configuration here
  $database = "1234";
  $username = "1234";
  $password = "1234";
  $hostname = "1234";
  // Key for Passwortencryption
  $secret_key = "1234567890";
?>
```

A.7. convert2mysql.php4

```
<?php
  /*****************************************
  /* Programm convert2mysql.php4
  /* convert binary data into MySQL data
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  function convertFile($file) {
    // needed: name of input file
    error_reporting(63); // 0 = none, 63 = full
    include("/home/schtorch/public_html/diplom/config.php4")
      ;
    set_magic_quotes_runtime(0);
    //Check if file exists and open it
    $fp = fopen($file, "r");
    if (!$fp) {
      return "Datei ".$file." konnte nicht geladen
        werden\n";
    };
```

```
// Read first line with Timestamp
$datetime = fgets($fp, 150);
// Get date
$date = substr($datetime, 0, 10);
if (!(preg_match("/^....-..-../", $date))) {
  fclose ($fp);
  return "Konnte Datum in Datei $file nicht finden
    (Fehler in Messdatei)\n";
};
// Get time
$time = substr($datetime, 11, 8);
if (!(preg_match("/^..:..:../", $time))) {
  fclose ($fp);
  return "Konnte Zeit in Datei $file nicht finden
    (Fehler in Messdatei)\n";
};
// Get sampling rate
if (substr($datetime, 19, 1) == '.') {
  // we deal with microtime
  $a = explode(" ", substr($datetime, 20));
  $rate = strlen($a[0]); // 1/10, 1/100 ...
  $timelen = $a[1]; // how many bytes for timestamp used
  $start_mictime = $a[0];
}
if (!isset($rate)) {
  $rate = 0; // rate in seconds
}
if (!isset($timelen)) {
  $timelen = substr($datetime, 20);
}
if (!isset($start_mictime)) {
  $start_mictime = "";
}
// Get name of station in outback
$station_name = trim(fgets($fp, 400));
// Get name of computer in station
$computer_name = trim(fgets($fp, 400));
```

```
// Find the corresponding table to computer
$db = mysql_connect($hostname, $username, $password);
mysql_select_db($database, $db);
$result = mysql_query("SELECT ‘table_name‘ FROM ‘admin‘
  WHERE computer=\"$computer_name\"", $db);
$row = mysql_fetch_row($result);
echo mysql_error($db);
$table = $row[0];
mysql_close($db);
if (!isset($table)) {
  fclose ($fp);
  return "Computer $computer_name in Datei $file
    unbekannt (Eingabefehler in Profilmaske)\n";
}
while (!feof($fp)) {
  $a = fread($fp, 1);
  $a = unpack(’C1size’, $a);
  if ($a[’size’] == 0) // Ignore all until Startbit 0x00
  {
    break;
  }
}
// Try to open Temp-File for Output
$fp_out = fopen("MySQL_tmp.txt", "w+");
if (!$fp_out) {
  fclose ($fp);
  return "Konnte temporäre Datei nicht anlegen
    (Festplatte voll?)\n";
};
// convert data
$year = substr($date, 0, 4);
$month = substr($date, 5, 2);
$day = substr($date, 8, 2);
$hour = substr($time, 0, 2);
$minute = substr($time, 3, 2);
$second = substr($time, 6, 2);
$start_time = mktime($hour, $minute, $second, $month,
```

```
    $day, $year);
  while (!feof($fp)) {
    $time_offset = "";
    $mictime = "";
    for ($j = 0; $j < $timelen; $j++) //read date and
      time offset
    {
      $a = fread($fp, 1);
      if ($a == "") continue 2;
      $a = unpack('C1size', $a);
      $b = dechex($a['size']);
      if (strlen($b) == 1) $b = "0".$b;
      $time_offset .= $b;
    }
    $time_offset = hexdec($time_offset)+$start_mictime;
    //echo "time1: ".$time_offset."\n";
    // do we deal mith mictime?
    if ($rate) {
      if (strlen($time_offset) < $rate) {
        $mictime = $time_offset;
      } else {
        $mictime = substr($time_offset,
          strlen($time_offset)-$rate);
      }
      $mictime = "0.".$mictime;
    }
    //echo "Mic: ".$mictime."\n";
    if (strlen($time_offset) > $rate) {
      $time_offset = substr($time_offset, 0,
        strlen($time_offset)-$rate);
    } else {
      $time_offset = "";
    }
    //echo "time2: ".$time_offset."\n";;
    $timesec = $start_time;
    $timesec += $time_offset;
    $id = $timesec.$mictime*10;
```

```
    $timesec = date("Y-m-d H:i:s", $timesec);
    if ($a != "") fwrite($fp_out,
      $id.";".$timesec.";".$mictime.";");
    for ($j = 0; $j < 6; $j++) {
      $a = fread($fp, 1);
      if ($a == "") break;
      $a = unpack('C1size', $a);
      fwrite($fp_out, $a['size'].";");
    }
    $a = fread($fp, 1);
    if ($a == "") break;
    $a = unpack('C1size', $a);
    fwrite($fp_out, "\n");
    if ($a['size']  != 0) {
      return "Konnte Stoppbits in Datei $file nicht
        finden (Fehler in Messdatei)\n";
    }
  }
  fclose($fp);
  fclose($fp_out);
  set_magic_quotes_runtime(get_magic_quotes_gpc());
  // Write data in table
  $db = mysql_connect($hostname, $username, $password);
  mysql_select_db($database, $db);
  mysql_query("LOAD DATA LOCAL INFILE 'MySQL_tmp.txt'
    IGNORE INTO TABLE '$table' FIELDS TERMINATED BY ';'
    LINES TERMINATED BY '\n' ", $db);
  if (mysql_error($db)) {
    mysql_close($db);
    return (mysql_error($db));
  };
  mysql_query("OPTIMIZE TABLE '$table'", $db);
  mysql_close($db);
  unlink("/home/schtorch/diplom/MySQL_tmp.txt");
  return "Datei $file erfolgreich konvertiert\n";
};
```

A.8. index.php4

```
<?php
  /*****************************************
  /* Programm index.php4
  /* Main menu
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  // Name of computers and their tables needed
  error_reporting(63); // 0 = none, 63 = full
  include("./config.php4");
  $db = mysql_connect($hostname, $username, $password);
  mysql_select_db($database, $db);
  $result = mysql_query("SELECT
    ‘name‘,‘computer‘,‘table_name‘ FROM ‘admin‘ WHERE 1",
    $db);
  echo mysql_error($db);
  $i = 0;
  while($row = mysql_fetch_row($result)) {
    $name[$i] = $row[0];
    $computer[$i] = $row[1];
    $table[$i] = $row[2];
    $i++;
  }
  mysql_close($db);
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Analyse und Administration</title>
</head>
<body>
<center><a href="http://www.tu-ilmenau.de">
<img src="tu_header.jpg" width=574 height=40 align="middle"
```

```
  border=0></a>
<h1>Überwachung von Windkraftanlagen</h1>
<h2>Ein Tool zur Analyse und Administration von
  Windkraftanlagen</h2>
Ansicht der Messwerte einer Anlage, bitte wählen Sie:
  <br><br>
<form action="menugraf.php4" method="POST">
<select name="table">
<?php
  for ($i = 0; $i < sizeof($name); $i++) {
    echo "<option
      value=\"$table[$i]\">$name[$i] ($computer[$i])</
      option>";
  }
?>
</select>  
<input type="submit" name="Anzeigen" value="Anzeigen">
</form> <br>
Profil einer Anlage ändern, bitte wählen Sie:   <br> <br>
<form action="profile.php4" method="POST">
<select name="table">
<?php
  for ($i = 0; $i < sizeof($name); $i++) {
    echo "<option
      value=\"$table[$i]\">$name[$i] ($computer[$i])</
      option>";
  }
?>
</select>  
<input type="submit" name="Ändern" value="Ändern">
</form> <br>
Anlegen eines <a href="profile.php4">neuen Profils</a>
  einer Anlage<br> <br>
Log-Datei der Datendownloads <a
  href="diplom.log">ansehen</a> <br><br>
Datenbank manuell <a
  href="../diplom/phpMyAdmin/index.php">administrieren</a>
```

```
  <br><br><br >
<hr>
(c) 2002 Jens Bierkandt, entstanden im Rahmen meiner
  Diplomarbeit </center>
</body>
</html>
```

A.9. profile.php4

```
<?php
  /*****************************************
  /* Programm profile.php4
  /* edit or add profile
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  error_reporting(63); // 0 = none, 63 = full
  include("/home/schtorch/public_html/diplom/config.php4");
  $db = mysql_connect($hostname, $username, $password);
  mysql_select_db($database, $db);
  if (isset($table)) {
    $result = mysql_query("SELECT * FROM ‘admin‘ WHERE
      table_name=\"$table\"", $db);
    $row = mysql_fetch_row($result);
    echo mysql_error($db);
    $name = $row[0];
    $computer = $row[1];
    $table = $row[2];
    $phone = $row[3];
    $protocol = $row[4];
    $user = $row[5];
    $result = mysql_query("SELECT
      DECODE(passwd,\"$secret_key\") FROM ‘admin‘ WHERE
      table_name=\"$table\"", $db);
```

```
    $row = mysql_fetch_row($result);
    $passwd = $row[0];
  } else {
    $name = "";
    $computer = "";
    $table = "";
    $phone = "";
    $protocol = "";
    $user = "";
    $passwd = "";
  }
  mysql_close($db);
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Analyse und Administration</title>
</head>
<body>
<center><a href="http://www.tu-ilmenau.de">
<img src="tu_header.jpg" width=574 height=40 align="middle"
  border=0></a>
<h1>Profileingabe</h1>
<form action="check_profile.php4" method="POST">
<table width="500">
<tr>
<td align="left">Name der Windkraftanlage:</td>
<td align="left">
<?php
  echo "<input type=\"text\" name=\"name\" value=\"$name\"
    maxlength=20>";
?>
</td></tr>
<tr>
<td align="left">Name des Messcomputers: (nicht doppelt
  vergeben)</td>
<td align="left">
```

```
<?php
  echo "<input type=\"text\" name=\"computer\"
    value=\"$computer\" maxlength=20>";
?>
</td></tr>
<tr>
<td align="left">Name der Datentabelle: (nicht doppelt
  vergeben)</td>
<td align="left">
<?php
  if (isset($row[0])) {
    echo $table;
    echo "<input type=\"hidden\" name=\"table\"
      value=\"$table\">";
  }
  else
  echo "<input type=\"text\" name=\"table\"
    value=\"$table\" maxlength=15>";
?>
</td></tr>
<tr>
<td align="left">Telefonnummer des Messrechners (nur
  Ziffern):</td>
<td align="left">
<?php
  echo "<input type=\"text\" name=\"phone\"
    value=\"$phone\" maxlength=30>";
?>
</td></tr>
<tr>
<td align="left">Modemart des Messrechners:</td>
<td align="left">
<?php
  echo "<input type=\"radio\" name=\"protocol\"
    value=\"hdlc sync\"";
  if ($protocol == "hdlc sync" or $protocol == "") echo "
    checked";
```

```
  echo "> ISDN ";
  echo "<input type=\"radio\" name=\"protocol\"
    value=\"modem\"";
  if ($protocol == "modem") echo " checked";
  echo "> Analog";
?>
</td></tr>
<tr>
<td align="left">Name des Logins auf dem Messcomputer:</td>
<td align="left">
<?php
  echo "<input type=\"text\" name=\"user\" value=\"$user\"
    maxlength=30>";
?>
</td></tr>
<tr>
<td align="left">Passwort des Logins auf dem
  Messcomputer:</td>
<td align="left">
<?php
  echo "<input type=\"password\" name=\"passwd\"
    value=\"$passwd\" maxlength=30>";
?>
</td></tr>
<tr>
<td align="left">Wiederholung des Passworts:</td>
<td align="left">
<?php
  echo "<input type=\"password\" name=\"passwd1\"
    value=\"$passwd\" maxlength=30>";
  if (isset($row[0]))
  echo "<input type=\"hidden\" name=\"change\"
    value=\"true\">";
?>
</td></tr>
<tr><td colspan=2 align="right">  <br>
<input type="submit" name="Eintragen" value="Eintragen">
```

```
</td></tr>
</table>
</form>
<br><a href=index.php4>Hauptmenü</a><br>
<hr>
(c) 2002 Jens Bierkandt, entstanden im Rahmen meiner
  Diplomarbeit </center>
</body>
</html>
```

A.10. check_profile.php4

```
<?php
  /*****************************************
  /* Programm check_profile.php4
  /* checks data entered in profile.php4
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  error_reporting(63); // 0 = none, 63 = full
  include("./config.php4");
  // connect to database
  $db = mysql_connect($hostname, $username, $password);
  mysql_select_db($database, $db);
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Analyse und Administration</title>
</head>
<body>
<center><a href="http://www.tu-ilmenau.de">
<img src="tu_header.jpg" width=574 height=40 align="middle"
  border=0></a >
```

```
<h1>Kontrolle der Profileingabe</h1>
<br><br>
<?php
  $errorstr = "";
  // check if name of table already exists
  if ((!isset($change)) AND $table != "") {
    $result = mysql_query("SELECT table_name FROM ‘admin‘
      WHERE table_name=\"$table\"", $db);
    $row = mysql_fetch_row($result);
    echo mysql_error($db);
    $table_check = $row[0];
    if ($table_check) $errorstr .= "Tabellenname schon
      vorhanden<br>\n";
  }
  // Check the entered variables
  if ($name == "")
  $errorstr .= "Name der Anlage fehlt<br>\n";
  if ($computer == "")
  $errorstr .= "Name des Messcomputers fehlt<br>\n";
  if ($computer != "") {
    $result = mysql_query("SELECT ‘computer‘,‘table_name‘
      FROM ‘admin‘ WHERE computer=\"$computer\"", $db);
    $row = mysql_fetch_row($result);
    echo mysql_error($db);
    $computer_check = $row[0];
    $table_check = $row[1];
    if ($computer_check) {
      if (!isset($change) OR ($table_check  != $table))
      $errorstr .= "Computername schon vorhanden<br>\n";
    }
  }
  if ($table == "")
  $errorstr .= "Name der Datentabelle fehlt<br>\n";
  if ($phone == "")
  $errorstr .= "Telefonnummer der Anlage fehlt<br>\n";
  if (preg_match("/.*\D/", $phone) == true)
  $errorstr .= "Telefonnummer enthält ungültige
```

```
    Zeichen<br>\n";
  if ($user == "")
  $errorstr .= "Name des Logins auf dem Messcomputer
    fehlt<br>\n";
  if ($passwd == "")
  $errorstr .= "Passwort des Logins fehlt\n";
  if ($passwd1 == "")
  $errorstr .= "Passwortwiederholung des Logins fehlt\n";
  if ($passwd != $passwd1)
  $errorstr .= "Passwortwiederholung ist falsch\n";
  if (!$errorstr) // Erverything seems to be ok
  {
    mysql_query("replace into admin values(\"$name\",
      \"$computer\" ,
    \"$table\", \"$phone\", \"$protocol\", \"$user\",
      ENCODE(\"$passwd\",\"$secret_key\"))", $db);
    echo mysql_error($db);
    if (!isset($change)) // new entry, new data table needs
      to be created
    {
      mysql_query("
      CREATE TABLE $table (
      id bigint(20) unsigned NOT NULL,
      timestamp datetime NOT NULL default '0000-00-00
        00:00:00',
      mictime double unsigned NOT NULL default '0',
      sensor1 tinyint(3) unsigned NULL,
      sensor2 tinyint(3) unsigned NULL,
      sensor3 tinyint(3) unsigned NULL,
      sensor4 tinyint(3) unsigned NULL,
      sensor5 tinyint(3) unsigned NULL,
      sensor6 tinyint(3) unsigned NULL,
      PRIMARY KEY  (id),
      UNIQUE KEY timestamp (timestamp,mictime),
      INDEX (id),
      INDEX (timestamp)
      ) TYPE=MyISAM COMMENT='Sensor data'
```

```
    ", $db);
    echo mysql_error($db);
    echo "Neue Datentabelle wurde erstellt.<br>\n";
    echo "Die ersten Daten werden nach dem nächsten
      Datendownload vom Messrechner bereit stehen.<br>\n";
  }
  // Create shortcut on KDE desktop to connect
  $fp = fopen("/home/schtorch/KDesktop/Connect to
    $computer.desktop", "w+");
  if (!$fp) echo "Konnte Shortcut auf dem Desktop nicht
    anlegen";
  fwrite($fp, "
  [Desktop Entry]
  Comment[en_US]=
  Encoding=UTF-8
  Exec=/home/schtorch/diplom/pppconn.sh $phone $protocol
    $user $passwd
  Icon=
  MimeType=
  Name[en_US]=Connect to $computer
  Path=
  ServiceTypes=
  SwallowExec=
  SwallowTitle=
  Terminal=true
  TerminalOptions=
  Type=Application
  X-KDE-SubstituteUID=false
  X-KDE-Username=
  ");
  fclose($fp);
  echo "Das Profil wurde erfolgreich
    abgespeichert.<br>\n";
  echo "<br><a href=\"index.php4\">Hauptmenü</a>";
} else {
  echo "Leider ist folgender Fehler aufgetreten:<br>\n";
  echo $errorstr;
```

```
  }
  mysql_close($db);
?>
<br>
<hr>
(c) 2002 Jens Bierkandt, entstanden im Rahmen meiner
  Diplomarbeit </center>
</body>
</html>
```

A.11. menugraf.php4

```
<?php
  /*****************************************
  /* Programm menugraph.php4
  /* lets select data to be displayed
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  // Name of computers needed
  set_time_limit(60);
  error_reporting(63); // 0 = none, 63 = full
  include("./config.php4");
  // Check, if table name is given
  if (!isset($table))
  Header("Location: index.php4"); // Forward user to main
    page to reselect table
  $db = mysql_connect($hostname, $username, $password);
  if (!$db) {
    return;
  }
  mysql_select_db($database, $db);
  $result = mysql_query("SELECT ‘name‘,‘computer‘ FROM
    ‘admin‘ WHERE table_name=\"$table\"", $db);
```

```
  echo mysql_error($db);
  $row = mysql_fetch_row($result);
  $name = $row[0];
  $computer = $row[1];
  mysql_close($db);
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Optionen der Messwerte</title>
</head>
<body>
<center><a href="http://www.tu-ilmenau.de">
<img src="tu_header.jpg" width=574 height=40 align="middle"
  border=0></a>
<h1>Anzeigeoptionen für
<?php
  echo $computer;
?>
</h1>
<form action="showgraf.php4" method="POST">
<table border="1" width=600>
<tr>
<td colspan=3>
<table align="left">
<tr><td>
<b>Art der Anzeige:</b>
</td><td>
<input type="radio" name="type" value="temp" checked>
  Temperaturkurve
<input type="radio" name="type" value="count"> Häufigkeit
<input type="checkbox" name="stat" value="1" checked>
  Statistik
</td></tr>
<tr><td></td><td>
<input type="radio" name="type" value="diff_count">
  Temperaturhübe
```

```
</td></tr>
</table>
</td>
</tr>
<tr>
<td valign="top">
<table align="left">
<tr>
<td><b>Sensorauswahl:</b></td>
</tr>
<tr>
<td><input type="checkbox" checked name="sensor[]"
  value="1"> Sensor 1</td>
<td><input type="checkbox" name="max[]" value="1"> Max</td>
<td><input type="checkbox" name="min[]" value="1"> Min</td>
</tr>
<tr>
<td><input type="checkbox" name="sensor[]" value="2">
  Sensor 2</td>
<td><input type="checkbox" name="max[]" value="2"> Max</td>
<td><input type="checkbox" name="min[]" value="2"> Min</td>
</tr>
<tr>
<td><input type="checkbox" name="sensor[]" value="3">
  Sensor 3</td>
<td><input type="checkbox" name="max[]" value="3"> Max</td>
<td><input type="checkbox" name="min[]" value="3"> Min</td>
</tr>
<tr>
<td><input type="checkbox" name="sensor[]" value="4">
  Sensor 4</td>
<td><input type="checkbox" name="max[]" value="4"> Max</td>
<td><input type="checkbox" name="min[]" value="4"> Min</td>
</tr>
<tr>
<td><input type="checkbox" name="sensor[]" value="5">
  Sensor 5</td>
```

```
<td><input type="checkbox" name="max[]" value="5"> Max</td>
<td><input type="checkbox" name="min[]" value="5"> Min</td>
</tr>
<tr>
<td><input type="checkbox" name="sensor[]" value="6">
  Sensor 6</td>
<td><input type="checkbox" name="max[]" value="6"> Max</td>
<td><input type="checkbox" name="min[]" value="6"> Min</td>
</tr>
</table>
</td>
<td>
<table>
<tr>
<td><b>Bildgröße:</b></td>
</tr>
<tr>
<td><input type="radio" checked name="size"
  value="600x400"> 600x400</td>
</tr>
<tr>
<td><input type="radio" name="size" value="800x600">
  800x600</td>
</tr>
<tr>
<td><input type="radio" name="size" value="1024x786">
  1024x786</td>
</tr>
<tr>
<td> </td>
</tr>
<tr><td><input type="radio" name="size"> variabel</td></tr>
<tr><td>x: <input type="text" name="width_graph" size="6"
  value="800"> Pixel</td></tr>
<tr><td>y: <input type="text" name="height_graph" size="6"
  value="400"> Pixel</td></tr>
</table>
```

```
<td valign="top">
<table>
<tr>
<td><b>Skalierung:</b></td>
</tr>
<tr>
<td><input type="radio" checked name="scale" value="auto">
  automatisch</td>
</tr>
<tr>
<td> </td>
</tr>
<tr>
<td><input type="radio" name="scale" value="man">
  manuell</td>
</tr>
<tr>
<td>Minimum: <input type="text" name="scale_min" size="3"
  value="35">°C</td>
</tr>
<tr>
<td>Maximum: <input type="text" name="scale_max" size="3"
  value="45">°C</td>
</tr>
</table>
</td>
</tr>
<tr><td valign="top">
<table>
<tr>
<td colspan=2><b>Zeitraum:</b></td>
</tr>
<tr>
<td><input checked type="radio" name="timedef"
  value="yesterday"></td><td align="left">gestern</td >
</tr>
<tr>
```

```
<td>
<input type="radio" name="timedef" value="since"></td><td>
die letzten <input type="text" name="days" value="7"
  size="3"> Tage</td>
</td>
</tr>
<tr>
<td> </td>
</tr>
</tr>
<tr>
<td valign="top">
<input type="radio" name="timedef"
  value="diff"></td><td>von (tt/mm/jj)<br>
<input type="text" name="from_d" size="2" value=""
  maxlength="2">.
<input type="text" name="from_m" size="2" value=""
  maxlength="2">.
<input type="text" name="from_y" size="2" value=""
  maxlength="2"><br>
<input type="text" name="from_h" size="2" value="00"
  maxlength="2">:
<input type="text" name="from_min" size="2" value="00"
  maxlength="2">:
<input type="text" name="from_s" size="2" value="00"
  maxlength="2"> Uhr<br></td>
<td>bis (tt/mm/jj)<br>
<input type="text" name="to_d" size="2" value=""
  maxlength="2">.
<input type="text" name="to_m" size="2" value=""
  maxlength="2">.
<input type="text" name="to_y" size="2" value=""
  maxlength="2"><br>
<input type="text" name="to_h" size="2" value="23"
  maxlength="2">:
<input type="text" name="to_min" size="2" value="59"
  maxlength="2">:
```

```
<input type="text" name="to_s" size="2" value="59"
  maxlength="2"> Uhr<br>
</td>
</tr>
</table>
</td><td valign="top" colspan=2>
<table>
<tr>
<td><b>Weitere Optionen:</b></td>
</tr>
<tr>
<td>Glättung über <input type="text" name="average"
  value=10 size=5 maxlength=5> Werte</td>
</tr>
<tr><td> </td></tr>
<tr>
<td><B> Nur für Temperaturhübe:</B></td>
</tr>
<tr>
<td>Dauer der Hübe:
von <input type="text" name="diff_min" value="10" size="3"
  maxlength="5"> s
bis  <input type="text" name="diff_max" value="60" size="3"
  maxlength="5"> s
</td>
</tr>
<tr>
<td>Auflösung Temperaturhub: <input type="text"
  name="temp_diff" value=2 size=3 maxlength=3> K</td >
</tr>
<tr>
<td>Auflösung Starttemperatur: <input type="text"
  name="start_diff" value=2 size=3 maxlength=3> K</td >
</tr>
</table></td>
</tr>
</table>
```

```
<?php
  echo "<input type=\"hidden\" name=\"table\"
    value=\"$table\">";
  echo "<input type=\"hidden\" name=\"computer\"
    value=\"$computer\">";
?>
<input type="submit" value="Erstellen">
</form>
</body>
</html>
```

A.12. showgraf.php4

```
<?php
  /*****************************************
  /* Programm showgraph.php4
  /* controls input from menugraf
  /* and displays the data
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
  // Date in the past
  header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "
    GMT");
  // always modified
  header("Cache-Control: no-store, no-cache,
    must-revalidate");
  // HTTP/1.1
  header("Cache-Control: post-check=0, pre-check=0", false);
  header("Pragma: public");
  // HTTP/1.0
  set_time_limit(600);
  error_reporting(63); // 0 = none, 63 = full
```

```
include("./config.php4");
// Check variables and initialize
if (!isset($table))
Header("Location: index.php4"); // Forward user to main
  page to reselect table
$yestdate = time()-60*60*24;
$yestyear = date("Y", $yestdate);
$yestmonth = date("m", $yestdate);
$yestday = date("d", $yestdate);
if (!isset($average)) $average = 10;
if ($average > 99999) $average = 99999;
if (!isset($timedef)) {
  $from_time = $yestyear.$yestmonth.$yestday."000000";
  $to_time = $yestyear.$yestmonth.$yestday."235959";
} else {
  if ($timedef == "yesterday") {
    $from_time = $yestyear.$yestmonth.$yestday."000000";
    $to_time = $yestyear.$yestmonth.$yestday."235959";
  }
  if ($timedef == "since") {
    $fromdate = time()-60*60*24*$days;
    $fromyear = date("Y", $fromdate);
    $frommonth = date("m", $fromdate);
    $fromday = date("d", $fromdate);
    $from_time = $fromyear.$frommonth.$fromday."000000";
    $to_time = $yestyear.$yestmonth.$yestday."235959";
  }
  if ($timedef == "diff") {
    if ($from_d < 1 || $from_d > 31
    || $from_m < 1 || $from_m > 12
    || $from_y < 0 || $from_y > 99
    || $from_h < 0 || $from_h > 24
    || $from_min < 0 || $from_min > 59
    || $from_s < 0 || $from_s > 59
    || $to_d < 1 || $to_d > 31
    || $to_m < 1 || $to_m > 12
    || $to_y < 0 || $to_y > 99
```

```
    || $to_h < 0 || $to_h > 24
    || $to_min < 0 || $to_min > 59
    || $to_s < 0 || $to_s > 59
    ) {
      echo "Falsche Datumeingabe";
      return 0;
    } else {
      $unix_from_time = mktime($from_h, $from_min,
        $from_s, $from_m, $from_d, $from_y);
      $unix_to_time = mktime($to_h, $to_min, $to_s,
        $to_m, $to_d, $to_y);
      $from_time = date("YmdHis", $unix_from_time);
      $to_time = date("YmdHis", $unix_to_time);
    }
  }
  if ($scale == "man" AND (!isset($scale_min) OR
    !isset($scale_max))) $scale = "auto";
};
// Initialize connection to DB
$db = mysql_connect($hostname, $username, $password);
if (!$db) {
  return;
}
mysql_select_db($database, $db);
echo mysql_error();
// To supress some overlayed high waves, we average the
  values by given number and put it in a temp table
mysql_query("DROP TABLE tmp", $db);
// Create temp table
$result = "CREATE TABLE tmp (
id bigint(20) unsigned NOT NULL,
timestamp datetime NOT NULL default '0000-00-00 00:00:00',
mictime double unsigned NOT NULL default '0',";
for ($a = 0; $a < sizeof($sensor); $a++) {
  $result .= "sensor".$sensor[$a]." FLOAT(3,2) NULL ,";
}
$result = substr($result, 0, strlen($result)-1);
```

```
$result .= ") TYPE=MyISAM COMMENT='Temp data';";
$result = mysql_query($result, $db);
echo mysql_error();
$t = time();
// average the data I
/*
mysql_query("LOCK TABLES `tmp` WRITE, `$table`
  READ;",$db);
echo mysql_error();
for ($i=0;$i<$average;$i++) {
$result="INSERT IGNORE INTO `tmp` SELECT `id`,
  `timestamp`, `mictime`, ";
for ($a=0;$a<sizeof($sensor);$a++) {
$result.="AVG(`sensor".$sensor[$a]."`),";
}
$result=substr($result,0,strlen($result)-1);
$result.=" FROM `$table` WHERE timestamp>=$from_time AND
  timestamp<=$to_time GROUP BY
  ((id-$i)-((id-$i)%$average))/$average;";
$result=mysql_query($result,$db);
echo mysql_error();
}
mysql_query("UNLOCK TABLES;",$db);
//echo time()-$t;
*/
// Select data II
$result = "SELECT id, timestamp, mictime, ";
for ($b = 0; $b < sizeof($sensor); $b++) {
  $result .= "sensor".$sensor[$b]." ,";
}
$result = substr($result, 0, strlen($result)-1);
$result .= "FROM `$table` WHERE timestamp>=$from_time AND
  timestamp<=$to_time ORDER BY id";
// initialize values
for ($a = 0; $a < $average; $a++) {
  for ($b = 0; $b < sizeof($sensor); $b++) {
    $arr[$b][$a] = 0;
```

```
  }
}
for ($b = 0; $b < sizeof($sensor); $b++) {
  $last_avg[$b] = 0;
}
$first_avg = 0;
$query = mysql_query($result, $db);
echo mysql_error($db);
$fp_out = fopen("MySQL_tmp.txt", "w+");
while($row = mysql_fetch_row($query)) {
  $outstr = $row[0].";".$row[1].";".$row[2];
  for ($b = 0; $b < sizeof($sensor); $b++) {
    $arr[$b][$first_avg] = $row[$b+3];
    $last_avg[$b] += $row[$b+3];
    $outstr .= ";".($last_avg[$b]/$average);
  }
  $first_avg++;
  if ($first_avg == $average) $first_avg = 0;
  for ($b = 0; $b < sizeof($sensor); $b++) {
    $last_avg[$b] -= $arr[$b][$first_avg];
  }
  fwrite($fp_out, $outstr."\n");
}
fclose($fp_out);
mysql_query("LOAD DATA LOCAL INFILE 'MySQL_tmp.txt'
  IGNORE INTO TABLE 'tmp' FIELDS TERMINATED BY ';' LINES
  TERMINATED BY '\n' ", $db);
//echo time()-$t;
$result = mysql_query("SELECT MIN(id), MAX(id) FROM
  tmp;", $db);
$row = mysql_fetch_row($result);
if ($row[0] == "") {
  // No data available
  echo "<b>Fehler:</b> Keine Daten im Zeitraum mit
    angegebenen Glättungsfaktor";
  return;
}
```

```
  $min_id = $row[0];
  $max_id = $row[1];
  // We only use the values, that have used the average
    completely
  mysql_query("DELETE FROM tmp where
    id<".($min_id+$average-1).";", $db);
  mysql_query("DELETE FROM tmp where
    id>".($max_id-$average+1).";", $db);
  $result = mysql_query("SELECT MIN(id) FROM tmp;", $db);
  $row = mysql_fetch_row($result);
  if ($row[0] == "") {
    // No data available
    echo "<b>Fehler:</b> Keine Daten im Zeitraum durch
      angegebenen Glättungsfaktor";
    return;
  }
  if (isset($size)) {
    if ($size && strpos($size, "x")) {
      $arr = explode('x', $size);
      $width_graph = $arr[0];
      $height_graph = $arr[1];
    };
  };
  if (!(isset($width_graph) || isset($height_graph))) {
    $width_graph = 600;
    $height_graph = 400;
  };
  if (!isset($diff_min)) $diff_min = 5;
  if (!isset($diff_max)) $diff_max = 30;
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
  Transitional//EN">
<html>
<head>
<title>Anzeige der Messdaten</title>
</head>
<body>
```

```
<?php
  if (isset($type)) {
    if ($type == "temp") {
      echo "<p><img
        src=\"graph.php4?from_time=$from_time&to_time=$to_t
        ime&width_graph=$width_graph&height_graph=$height_gr
        aph&table=$table&";
      for ($i = 0; $i < sizeof($sensor); $i++) {
        echo "sensor[]=$sensor[$i]&";
      }
      if (isset($max)) {
        for ($i = 0; $i < sizeof($max); $i++) {
          echo "sendmax[]=$max[$i]&";
        }
      }
      if (isset($min)) {
        for ($i = 0; $i < sizeof($min); $i++) {
          echo "sendmin[]=$min[$i]&";
        }
      }
      if ($scale == "man") echo
        "scale_min=$scale_min&scale_max=$scale_max&";
      echo
        "scale=$scale&computer=$computer&average=$average;
      \" alt=\"Temperaturkurve\"></p>";
    }
    if ($type == "diff_count") {
      include("./diff_count.php4");
    }
    if ($type == "count") {
      echo "<p><img
        src=\"count.php4?from_time=$from_time&to_time=$to_t
        ime&width_graph=$width_graph&height_graph=$height_gr
        aph&table=$table&";
      for ($i = 0; $i < sizeof($sensor); $i++) {
        echo "sensor[]=$sensor[$i]&";
      }
```

```
      echo "computer=$computer&average=$average
      \" alt=\"Häufigkeit\"></p>";
    }
  }
  if (isset($stat)) {
    echo "
    <table border=1><tr><td>
    <p><h2>Statistik</h2></p>
    </td></tr>";
    // display exact time selected
    $result = mysql_query("SELECT MIN(timestamp),
      MAX(timestamp) FROM ‘tmp‘ WHERE
      timestamp>=$from_time AND timestamp<=$to_time", $db);
    $row = mysql_fetch_row($result);
    $from_time1 = $row[0];
    $to_time1 = $row[1];
    echo "<tr><td>
    Erster Zeitstempel: ".$from_time1."
    </tr></td>";
    echo "<tr><td>
    Letzter Zeitstempel: ".$to_time1."
    </td></tr>";
    // Get number of values
    $result = mysql_query("SELECT COUNT(*) FROM ‘tmp‘ WHERE
      timestamp>=$from_time AND timestamp<=$to_time", $db);
    $row = mysql_fetch_row($result);
    $rownumber = $row[0];
    echo "<tr><td>
    Anzahl der Werte pro Sensor: $rownumber
    </td> </tr>";
    echo "<tr><td>Glättung über $average Werte</td></tr>";
    echo "</table>";
    echo "<table border=1>";
    echo "<tr><td></td>";
    for ($a = 0; $a < sizeof($sensor); $a++) {
      echo "<td width=70>Sensor $sensor[$a]</td>";
    }
```

```
echo "</tr><tr>";
echo "<td>Minimalwert</td>";
for ($a = 0; $a < sizeof($sensor); $a++) {
  $result = mysql_query("SELECT
    MIN(sensor".$sensor[$a].")
  FROM ‘tmp‘ WHERE timestamp>=$from_time AND
    timestamp<=$to_time", $db);
  $row = mysql_fetch_row($result);
  $min_sensor = $row[0];
  echo "<td>
  ".$min_sensor."
  </td>";
}
echo "</tr><tr>";
echo "<td>Maximalwert</td>";
for ($a = 0; $a < sizeof($sensor); $a++) {
  $result = mysql_query("SELECT
    MAX(sensor".$sensor[$a].")
  FROM ‘tmp‘ WHERE timestamp>=$from_time AND
    timestamp<=$to_time", $db);
  $row = mysql_fetch_row($result);
  $min_sensor = $row[0];
  echo "<td>
  ".$min_sensor."
  </td>";
}
echo "</tr><tr>";
echo "<td>Durchschnitt</td>";
for ($a = 0; $a < sizeof($sensor); $a++) {
  $result = mysql_query("SELECT
    AVG(sensor".$sensor[$a].")
  FROM ‘tmp‘ WHERE timestamp>=$from_time AND
    timestamp<=$to_time", $db);
  $row = mysql_fetch_row($result);
  $min_sensor = $row[0];
  echo "<td>
  ".$min_sensor."
```

```
      </td>";
    }
    echo "</tr>";
    echo "</table>";
  }
  mysql_close($db);
?>
</body>
</html>
```

A.13. graph.php4

```
<?php
  /*****************************************
  /* Programm graph.php4
  /* displays temperature
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  error_reporting(63);
  set_time_limit(60);
  header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
  // Date in the past
  header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "
    GMT");
  // always modified
  header("Cache-Control: no-store, no-cache,
    must-revalidate");
  // HTTP/1.1
  header("Cache-Control: post-check=0, pre-check=0", false);
  header("Pragma: public");
  $total = time();
  include ("jpgraph-1.6.1/src/jpgraph.php");
  include ("jpgraph-1.6.1/src/jpgraph_line.php");
```

```
include ("/home/schtorch/public_html/diplom/config.php4");
// Check variables and initialize
if (!isset($table))
Header("Location: index.php4"); // Forward user to main
  page to reselect table
if (!isset($average)) $average = 10;
if ($average > 999) $average = 999;
if (!(isset($width_graph) || isset($height_graph))) {
  $width_graph = 600;
  $height_graph = 400;
};
if ($width_graph < 600 || $height_graph < 400) {
  $width_graph = 600;
  $height_graph = 400;
};
// check if only max and mins with refering sensor
  selected
if (isset($sendmax)) {
  $b = 0;
  for ($i = 0; $i < sizeof($sendmax); $i++) {
    for ($a = 0; $a < sizeof($sensor); $a++) {
      if ($sendmax[$i] == $sensor[$a]) {
        $max[$b] = $sendmax[$i];
        $b++;
      }
    }
  }
}
if (isset($sendmin)) {
  $b = 0;
  for ($i = 0; $i < sizeof($sendmin); $i++) {
    for ($a = 0; $a < sizeof($sensor); $a++) {
      if ($sendmin[$i] == $sensor[$a]) {
        $min[$b] = $sendmin[$i];
        $b++;
      }
    }
```

```
    }
  }
  // Initialize connection to DB
  $db = mysql_connect($hostname, $username, $password);
  mysql_select_db($database, $db);
  echo mysql_error();
  // Get number of rows
  // Needed to limit lines to draw
  $result = mysql_query("SELECT COUNT(*) FROM ‘tmp‘", $db);
  $row = mysql_fetch_row($result);
  $rownumber = $row[0];
  //echo $rownumber." ";
  if ($rownumber/$width_graph < 1) $rownumber =
    $width_graph;
  $result = mysql_query("SELECT MIN(id) FROM ‘tmp‘", $db);
  $row = mysql_fetch_row($result);
  $min_id = $row[0];
  // Select data
  $result = "SELECT id, UNIX_TIMESTAMP(timestamp), mictime,
    ";
  for ($a = 0; $a < sizeof($sensor); $a++) {
    $result .= "AVG(‘sensor".$sensor[$a]."‘),";
  }
  if (isset($max)) {
    for ($a = 0; $a < sizeof($max); $a++) {
      $result .= "MAX(‘sensor".$max[$a]."‘),";
    }
  }
  if (isset($min)) {
    for ($a = 0; $a < sizeof($min); $a++) {
      $result .= "MIN(‘sensor".$min[$a]."‘),";
    }
  }
  $result = substr($result, 0, strlen($result)-1);
  $row_step = ceil($rownumber/$width_graph);
  $result .= "FROM ‘tmp‘ GROUP BY
    (id-id%($row_step))/$row_step ORDER BY id ASC";
```

```
$result = mysql_query($result, $db);
echo mysql_error();
//echo $rownumber/$width_graph;
$i = 0;
while($row = mysql_fetch_row($result)) {
  //echo $row[0]."<br>";
  while ($min_id+$i*$row_step < $row[0]) {
    $mictime = $min_id+$i*$row_step;
    $time = date ("H:i:s", substr($mictime, 0,
      strlen($mictime)-1));
    $date = date ("d.M.Y", substr($mictime, 0,
      strlen($mictime)-1));
    $datax[$i] = $time.".".substr($mictime,
      strlen($mictime)-1, strlen($mictime))."\n".$date;
    for ($a = 0; $a < sizeof($sensor); $a++) {
      $ydata[$a][$i] = "";
      $maxdata[$a][$i] = "";
      $mindata[$a][$i] = "";
    }
    //echo $datax[$i]."--<br>";
    $i++;
  }
  $last_id = $row[0];
  $mictime = $row[2];
  $time = date ("H:i:s", $row[1]);
  $date = date ("d.M.Y", $row[1]);
  $datax[$i] = $time.".".($mictime*10)."\n".$date;
  for ($a = 0; $a < sizeof($sensor); $a++) {
    $ydata[$a][$i] = $row[$a+3];
  }
  //echo $datax[$i]." ".$ydata[0][$i]."<br>";
  if (isset($max)) {
    for ($a = 0; $a < sizeof($max); $a++) {
      $maxdata[$a][$i] = $row[sizeof($sensor)+$a+3];
    }
  }
  if (isset($min)) {
```

```
      for ($a = 0; $a < sizeof($min); $a++) {
        if (isset($max))
        $mindata[$a][$i] =
          $row[sizeof($sensor)+sizeof($max)+$a+3];
        else
        $mindata[$a][$i] = $row[sizeof($sensor)+$a+3];
      }
    }
    $i++;
  }
  //echo $i;
  $x_tick_diff = 1;
  if (isset($datax[0])) {
    $x_temp = round(($width_graph-120-50)/100);
    $x_tick_diff = (sizeof($datax)-1)/$x_temp;
    //echo $x_tick_diff;
  }
  if ($x_tick_diff < 10) $x_tick_diff = round($x_tick_diff);
  // Create the graph. These two calls are always required
  $graph = new Graph($width_graph, $height_graph, "auto");
  if ($scale != "man") {
    $graph->SetScale("textlin");
  } else {
    $graph->SetScale("textlin", $scale_min, $scale_max);
  }
  //if ($x_tick_diff<10) $graph->img->SetAntiAliasing();
  // Create the linear plots
  $color = array("blue", "green", "orange", "magenta",
    "black", "red");
  for ($a = 0; $a < sizeof($sensor); $a++) {
    if (isset($sensor[$a])) {
      $lineplot[$a] = new LinePlot($ydata[$a]);
      $lineplot[$a]->SetColor($color[$a]);
      $lineplot[$a]->SetWeight(1);
      $lineplot[$a]->SetLegend("Sensor $sensor[$a]");
      $graph->Add($lineplot[$a]);
    }
```

```
}
if (isset($max)) {
  $color = array("lightblue", "lightgreen",
    "lightsalmon", "magenta", "black", "lightred");
  for ($a = 0; $a < sizeof($max); $a++) {
    if (isset($max[$a])) {
      $maxplot[$a] = new LinePlot($maxdata[$a]);
      $maxplot[$a]->SetColor($color[$a]);
      $maxplot[$a]->SetWeight(1);
      $maxplot[$a]->SetLegend("Max Sensor $max[$a]");
      $graph->Add($maxplot[$a]);
    }
  }
}
if (isset($min)) {
  $color = array("darkblue", "darkgreen", "darkorange",
    "magenta", "black", "darkred");
  for ($a = 0; $a < sizeof($min); $a++) {
    if (isset($min[$a])) {
      $minplot[$a] = new LinePlot($mindata[$a]);
      $minplot[$a]->SetColor($color[$a]);
      $minplot[$a]->SetWeight(1);
      $minplot[$a]->SetLegend("Min Sensor $min[$a]");
      $graph->Add($minplot[$a]);
    }
  }
}
$graph->legend->Pos(0.03, 0.4, "right", "center");
$graph->xaxis->SetTickLabels($datax);
$graph->xaxis->SetTextTickInterval($x_tick_diff);
//$graph->xaxis->SetTextLabelInterval(2);
$graph->img->SetMargin(50, 120, 30, 50);
if (isset($computer)) $graph->title->Set($computer);
//$graph->xaxis->title->Set("\nDate");
//$graph->xaxis->SetLabelAngle(90);
$graph->yaxis->title->Set("°C");
$graph->title->SetFont(FF_FONT1, FS_BOLD);
```

```
  $graph->yaxis->title->SetFont(FF_FONT1, FS_BOLD);
  $graph->xaxis->title->SetFont(FF_FONT1, FS_BOLD);
  $graph->yaxis->SetColor("red");
  $graph->yaxis->SetWeight(2);
  $graph->SetShadow();
  //echo "Total: ".(time()-$total)."\n";
  // Display the graph
  $graph->Stroke();
  //echo "Total +graph: ".(time()-$total)."\n";
?>
```

A.14. diff_count.php4

```
<?php
  /*****************************************
  /* Programm diff_count.php4
  /* displays differences of temperature
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  error_reporting(63);
  set_time_limit(60);
  $total = time();
  include ("jpgraph-1.6.1/src/jpgraph.php");
  include ("jpgraph-1.6.1/src/jpgraph_line.php");
  include ("/home/schtorch/public_html/diplom/config.php4");
  // Check variables and initialize
  if (!isset($table))
  Header("Location: index.php4"); // Forward user to main
    page to reselect table
  if (!isset($average)) $average = 10;
  if ($average > 999) $average = 999;
  if (!isset($diff_min)) $diff_min = 5; // minimum time in
    seconds of a temperature change in one direction
```

```
if (!isset($diff_max)) $diff_max = 30;
if ($diff_min <= 0) $diff_min = 0.1;
if ($diff_max <= 0) $diff_max = 0.3;
if (!isset($start_diff)) $start_diff = 2;
if (!isset($temp_diff)) $temp_diff = 2;
if ($start_diff < 1) $start_diff = 2;
if ($temp_diff < 1) $temp_diff = 2;
$diff_min *= 10; // cause we stored the data in 1/10 sec
$diff_max *= 10;
// Initialize connection to DB
$db = mysql_connect($hostname, $username, $password);
mysql_select_db($database, $db);
echo mysql_error();
// Select data
for ($sens = 0; $sens < sizeof($sensor); $sens++) {
  $result = "SELECT ";
  $result .= "‘sensor".$sensor[$sens]."‘,";
  $result = substr($result, 0, strlen($result)-1);
  //$result.="timestamp ";
  $result .= "FROM ‘tmp‘ WHERE
    ‘sensor".$sensor[$sens]."‘!=’’ ORDER BY id";
  $result = mysql_query($result, $db);
  echo mysql_error();
  $row = mysql_fetch_row($result);
  $row[0] = round($row[0]);
  $inc = TRUE;
  $last_row = $row[0];
  $start_temp = $row[0];
  $row_diff = 0;
  $i = 1;
  while($row = mysql_fetch_row($result)) {
    $row[0] = round($row[0]);
    $i++;
    //echo $row[0]."<br>";
    // Check, if we still have increasing values
    if (($last_row <= $row[0]) && $inc == TRUE) {
      $row_diff = $row[0]-$start_temp;
```

```
    $last_row = $row[0];
    //echo "up<br>";
    continue;
  }
  // Check, if we still have decreasing values
  if (($last_row >= $row[0]) AND $inc == FALSE) {
    $row_diff = $start_temp-$row[0];
    $last_row = $row[0];
    //echo "down<br>";
    continue;
  }
  // The temperature changed direction
  // Did the last change of temperature direction took
    long enough?
  if ($i >= $diff_min AND $i <= $diff_max) {
    if ($inc) {
      // save incresing temperature changes
      if (!isset($datax[$start_temp][$row_diff]))
        $datax[$start_temp][$row_diff] = "0";
      $datax[$start_temp][$row_diff]++;
      //echo "Starttemp: $start_temp, Temperaturhub:
        $row_diff, Länge: $i, Anzahl:
        ".$datax[$start_temp][$row_diff]."<br>";
    } else {
      if (!isset($datax_min[$start_temp][$row_diff]))
        $datax_min[$start_temp][$row_diff] = "0";
      $datax_min[$start_temp][$row_diff]++;
      //echo "Starttemp: $start_temp, Temperaturhub:
        $row_diff, Länge: $i, Anzahl:
        ".$datax_min[$start_temp][$row_diff]."<br>";
    }
  }
  $start_temp = $last_row;
  $row_diff = abs($last_row-$row[0]);
  $last_row = $row[0];
  if ($inc == TRUE) $inc = FALSE;
  else $inc = TRUE;
```

```
  $i = 1;
}
// Get last change
$i++;
if ($i >= $diff_min AND $i <= $diff_max) {
  if ($inc) {
    // save incresing temperature changes
    if (!isset($datax[$start_temp][$row_diff]))
      $datax[$start_temp][$row_diff] = "0";
    $datax[$start_temp][$row_diff]++;
    //echo "Starttemp: $start_temp, Temperaturhub:
      $row_diff, Länge: $i, Anzahl:
      ".$datax[$start_temp][$row_diff]."<br>";
  } else {
    if (!isset($datax_min[$start_temp][$row_diff]))
      $datax_min[$start_temp][$row_diff] = "0";
    $datax_min[$start_temp][$row_diff]++;
    //echo "Starttemp: $start_temp, Temperaturhub:
      $row_diff, Länge: $i, Anzahl:
      ".$datax_min[$start_temp][$row_diff]."<br>";
  }
}
// Start output for positive temperature changes
// Get min and max of 3-dimensional array
$min_temp = 127;
$max_temp = 0;
$min_start = 127;
$max_start = 0;
for ($a = 1; $a < 127; $a++) {
  for ($i = 1; $i < 127; $i++) {
    if (!isset($datax[$a][$i])) continue;
    else
    {
      if ($min_temp > $i) $min_temp = $i;
      if ($max_temp < $i) $max_temp = $i;
      if ($min_start > $a) $min_start = $a;
      if ($max_start < $a) $max_start = $a;
```

```
    }
  }
}
echo "<h3>Positive Temperaturhübe für $computer, Sensor
  $sensor[$sens]<br>";
echo "Hübe zwischen ".($diff_min/10)." und
  ".($diff_max/10)." Sekunden</h3>";
echo "<table><tr><td>Starttemperatur in °C</td>";
for ($start_temp = $min_start; $start_temp <=
  $max_start; $start_temp = $start_temp+$start_diff) {
  echo "<td align=right width=30>$start_temp";
  if ($start_diff > 1) echo " bis
    ".($start_temp+$start_diff);
  echo "</td>";
}
echo "</tr>";
echo "<tr><td>Temperaturhub in K</td></tr>";
$i = 0;
for ($row_diff = $min_temp; $row_diff <= $max_temp;
  $row_diff = $row_diff+$temp_diff) {
  if ($i%2) echo "<tr>";
  else echo "<tr bgcolor=#FFFFCC>";
  echo "<td align=left>$row_diff";
  if ($temp_diff > 1) echo " bis
    ".($row_diff+$temp_diff);
  echo "</td>";
  for ($start_temp = $min_start; $start_temp <=
    $max_start; $start_temp = $start_temp+$start_diff) {
    $temp = "";
    for ($b = $start_temp; $b <
      $start_temp+$start_diff; $b++) {
      for ($c = $row_diff; $c < $row_diff+$temp_diff;
        $c++) {
        if (isset($datax[$b][$c])) {
          $temp += $datax[$b][$c];
        }
      }
```

```
    }
    if ($temp) {
      echo "<td align=right>".$temp."</td>";
    } else {
      echo "<td align=right>-</td>";
    }
  }
  echo "</tr>\n";
  $i++;
}
echo "</tr></table>";
// Start output for negative temperature changes
// Get min and max of 3-dimensional array
$min_temp = 127;
$max_temp = 0;
$min_start = 127;
$max_start = 0;
for ($a = 1; $a < 127; $a++) {
  for ($i = 1; $i < 127; $i++) {
    if (!isset($datax_min[$a][$i])) continue;
    else
    {
      if ($min_temp > $i) $min_temp = $i;
      if ($max_temp < $i) $max_temp = $i;
      if ($min_start > $a) $min_start = $a;
      if ($max_start < $a) $max_start = $a;
    }
  }
}
echo "<h3>Negative Temperaturhübe für $computer, Sensor
  $sensor[$sens]<br>";
echo "Hübe zwischen ".($diff_min/10)." und
  ".($diff_max/10)." Sekunden</h3>";
echo "<table><tr><td>Starttemperatur in °C</td>";
for ($start_temp = $min_start; $start_temp <=
  $max_start; $start_temp = $start_temp+$start_diff) {
  echo "<td align=right width=30>$start_temp";
```

```
  if ($start_diff > 1) echo " bis
    ".($start_temp+$start_diff);
  echo "</td>";
}
echo "</tr>";
echo "<tr><td>Temperaturhub in K</td></tr>";
$i = 0;
for ($row_diff = $min_temp; $row_diff <= $max_temp;
  $row_diff = $row_diff+$temp_diff) {
  if ($i%2) echo "<tr>";
  else echo "<tr bgcolor=#FFFFCC>";
  echo "<td align=left>-$row_diff";
  if ($temp_diff > 1) echo " bis
    -".($row_diff+$temp_diff);
  echo "</td>";
  for ($start_temp = $min_start; $start_temp <=
    $max_start; $start_temp = $start_temp+$start_diff) {
    $temp = "";
    for ($b = $start_temp; $b <
      $start_temp+$start_diff; $b++) {
      for ($c = $row_diff; $c < $row_diff+$temp_diff;
        $c++) {
        if (isset($datax_min[$b][$c])) {
          $temp += $datax_min[$b][$c];
        }
      }
    }
    if ($temp) {
      echo "<td align=right>".$temp."</td>";
    } else {
      echo "<td align=right>-</td>";
    }
  }
  echo "</tr>\n";
  $i++;
}
echo "</tr></table>";
```

```
  }
?>
```

A.15. count.php4

```
<?php
  /*****************************************
  /* Programm count.php4
  /* displays data selected in menugraph.php4
  /* (c) Copyright 2002, Jens Bierkandt
  /* e-mail: jens@bierkandt.org
  /* Entstanden im Rahmen meiner Diplomarbeit
  /*****************************************
  */
  header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
  // Date in the past
  header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "
    GMT");
  // always modified
  header("Cache-Control: no-store, no-cache,
    must-revalidate");
  // HTTP/1.1
  header("Cache-Control: post-check=0, pre-check=0", false);
  header("Pragma: public");
  set_time_limit(60);
  include ("jpgraph-1.6.1/src/jpgraph.php");
  include ("jpgraph-1.6.1/src/jpgraph_line.php");
  include ("/home/schtorch/public_html/diplom/config.php4");
  // Check variables and initialize
  if (!isset($table))
  Header("Location: index.php4"); // Forward user to main
    page to reselect table
  //if (!isset($sensor[0])) $sensor[0]="1";
  if (!(isset($width_graph) || isset($height_graph))) {
    $width_graph = 600;
    $height_graph = 400;
```

```
};
if ($width_graph < 600 || $height_graph < 400) {
  $width_graph = 600;
  $height_graph = 400;
};
// Initialize connection to DB
$db = mysql_connect($hostname, $username, $password);
mysql_select_db($database, $db);
echo mysql_error();
// Prefill array to have continuous data
$count_min = 127;
$count_max = 0;
for ($b = 0; $b < sizeof($sensor); $b++) {
  $result = mysql_query("SELECT
    MIN(sensor".$sensor[$b]."),
    MAX(sensor".$sensor[$b].") FROM ‘tmp‘", $db);
  echo mysql_error($db);
  $row = mysql_fetch_row($result);
  if (round($row[0]) < $count_min) $count_min =
    round($row[0]);
  if (round($row[1]) > $count_max) $count_max =
    round($row[1]);
}
//echo $count_max."\n";
for ($i = 0; $i <= ($count_max-$count_min); $i++) {
  $datax[$i] = $i+$count_min;
  //echo $datax[$i]." ";
  for ($b = 0; $b < sizeof($sensor); $b++) {
    $ydata[$b][$i] = "";
  }
}
// Select data
for ($b = 0; $b < sizeof($sensor); $b++) {
  $query = "SELECT count(*), round(sensor".$sensor[$b]."
    FROM ‘tmp‘ WHERE sensor".$sensor[$b]."!=’’ GROUP BY
    ROUND(sensor".$sensor[$b].") ASC";
  $result = mysql_query($query, $db);
```

```
  echo mysql_error($db);
  $i = 0;
  while($row = mysql_fetch_row($result)) {
    while ($i+$count_min < $row[1]) $i++;
    $ydata[$b][$i] = $row[0];
    //echo
      "Sensor".$b.":".$datax[$i].":".$ydata[$b][$i]." ";
    $i++;
  }
}
$x_tick_diff = 1;
if (isset($datax[0])) {
  $x_temp = round(($width_graph-120-50)/50);
  $x_tick_diff = (sizeof($datax)-1)/$x_temp;
  //echo $x_tick_diff;
}
if ($x_tick_diff < 10) $x_tick_diff = round($x_tick_diff);
// Create the graph. These two calls are always required
$graph = new Graph($width_graph, $height_graph, "auto");
$graph->SetScale("textlin");
// Create the linear plots
$color = array("blue", "green", "orange", "magenta",
  "black", "red");
for ($a = 0; $a < 6; $a++) {
  if (isset($sensor[$a])) {
    $lineplot[$a] = new LinePlot($ydata[$a]);
    $lineplot[$a]->SetFillColor($color[$a]);
    $lineplot[$a]->SetWeight(1);
    $lineplot[$a]->SetLegend("Sensor $sensor[$a]");
    $graph->Add($lineplot[$a]);
  }
}
$graph->legend->Pos(0.03, 0.3, "right", "center");
$graph->xaxis->SetTickLabels($datax);
$graph->xaxis->SetTextTickInterval($x_tick_diff);
//$graph->xaxis->SetTextLabelInterval(2);
$graph->img->SetMargin(50, 120, 30, 50);
```

```
  if (isset($computer)) $graph->title->Set($computer);
  $graph->xaxis->title->Set("°C");
  //$graph->xaxis->SetLabelAngle(90);
  $graph->title->SetFont(FF_FONT1, FS_BOLD);
  $graph->yaxis->title->SetFont(FF_FONT1, FS_BOLD);
  $graph->xaxis->title->SetFont(FF_FONT1, FS_BOLD);
  $graph->yaxis->SetColor("red");
  $graph->yaxis->SetWeight(2);
  $graph->SetShadow();
  // Display the graph
  $graph->Stroke();
?>
```

B. Erklärung

Erklärung:
Hiermit erkläre ich, dass ich diese Arbeit selbstständig durchgeführt und abgefasst habe. Quellen, Literatur und Hilfsmittel, die von mir benutzt wurden, sind als solche gekennzeichnet.

C. Thesen

1. In dieser Arbeit wird ein Konzept zur Übertragung von gemessenen Betriebsgrößen leistungselektronischer Bauteile über größere Strecken und deren anschließender grafischer Aufbereitung entwickelt. Die Werte dienen zur Bestimmung der zu erwartenden Lebensdauer der leistungselektronischen Baugruppen. Dabei wurde darauf Wert gelegt, die Messdaten wirtschaftlich zu übertragen und das System kostengünstig und stabil zu konzipieren. Weiterhin können entfernte Messrechner durch eine Fernadministration gewartet werden.

2. Ein Ausfall von elektronischen Bauteilen, Baugruppen und Systemen kann wirtschaftliche Folgen nach sich ziehen.

3. Eine Methode zur Vorhersage der Lebensdauer von elektronischen Bauteilen, Baugruppen und Systemen sind Tests im Rahmen der Qualitätssicherung, bei denen sie unter Zugrundelegung kritischer Einflussgrößen einer beschleunigten Alterung unterzogen werden.

4. Die kritischen Einflussgrößen sind in erster Linie Temperaturwechsel.

5. Zur Gewinnung realer Daten für eine beschleunigte Alterung ist eine Langzeituntersuchung der kritischen Einflussgrößen an den Bauteilen, Baugruppen und Systemen notwendig.

6. Können die Daten einer Langzeituntersuchung von einem entfernten Messrechner zur Auswertung nicht über eine fest installierte Leitung übertragen werden, so kann auf Telefonleitungen oder den Mobilfunk zurückgegriffen werden.

7. Die Untersuchungen zeigen, dass Linux als Serversystem zum Speichern von großen Datenmengen und zur Generierung der Grafiken kostengünstig ist und sehr stabil läuft.

8. MySQL als Datenbanksoftware ist für die Aufnahme von großen Datenmengen geeignet und arbeitet in unserem Fall mit ausreichender Geschwindigkeit.

9. Die Kommunikation eines Windows-Rechners mit wahlweise einem analogen Modem, einer ISDN-Karte oder einer GSM-Karte und einem Linuxserver mit einer ISDN-Karte läuft ohne nennenswerte Probleme.

10. Zur Vermeidung von Ausfällen der Messrechner müssen verschiedene Bauteile redundant ausgelegt werden.