

„Dezentrale mikrocontrollerbasierende Messtechnik in der Leittechnik historischer Gebäude.“

1. Einführung und Hintergrund

Im Hintergrund dieses Themas stehen historische Bibliotheksgebäude, deren Bestände durch ungünstige klimatische Verhältnisse in den Innenräumen in der Erhaltung gefährdet sind.

Damit Maßnahmen der Gebäudeleittechnik geplant werden können, ist die Aufnahme der Umweltgrößen über einen längeren Zeitraum erforderlich. Im speziellen zielen die Maßnahmen auf die Klimatechnik mit ihren Aufgaben der Heizungsregelung, Belüftungsanlagen.

Ein Messsystem kann mit dem Mikrocontroller Beck „IPC@CHIP“ realisiert werden (Bild rechts).



Bild: IPC@CHIP

2. Ziele und Schritte dieses Projekts

Somit ergeben sich 2 primäre Ziele dieses Projekts, die mit den folgenden Arbeitsschritten erreicht werden.

Messsystem zur Klimabeurteilung in den Innenräumen historischer Gebäude entwickeln:

- Physikalische Größen aufnehmen und digitalisieren
- Signale zum Mikrocontroller übertragen
- Sensorspezifische Messsignalverarbeitung und
- dezentrale Archivierung der Messdaten auf dem Mikrocontroller

Messwertverläufe visualisieren:

- Anbindung des Mikrocontrollers an das Bussystem Ethernet
- Plattformunabhängige grafische Darstellung der Klimaverläufe

Das Ethernet erweist sich als günstig, da speziell auf der Visualisierungsseite auf Standard-Büro-PC-Hardware zurückgegriffen werden kann, und die dezentrale Datenverarbeitung und Archivierung in Zukunft auch im Industriebereich an Bedeutung gewinnen wird, wie das Ethernet auch selbst.

Über das Ethernet kann mit Softwaretechnologien eine dezentrale Archivierung der Messdaten auf dem Mikrocontroller erreicht werden.

3. Relevante Messgrößen und Sensoren

Die Beschaffenheit des historischen Inventars beeinflussen diese vier physikalischen Größen:

- Temperatur: Infineon KTY 10-6
- Luftfeuchte: Honeywell HIH-3610
- Luftdruck: Motorola MPX-4100
- UV-Intensität: RLT UVD-39 (Bild rechts)
(Ultraviolettes Licht)

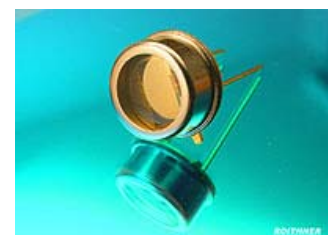


Bild: UV-Sensor

Grundsätzlich ist die *Temperatur* ein Einflussfaktor auf die Erhaltung der Materialien. Ist sie zu hoch, trocknen die Stoffe zu stark aus, sinkt sie zu stark ab, hat das einen Einfluss auf den Feuchtegehalt der Luft.

Die Luft kann bei einer bestimmten Temperatur nur eine bestimmte maximale Menge Wasser (Feuchtigkeit) aufnehmen. Die relative Feuchte zeigt, wie viel von dieser maximalen Feuchte (100 %) die Luft aufgenommen hat. Bei sinkenden Temperaturen und gleichbleibender Menge an Wasser steigt die relative Feuchte. Sinkt die Temperatur soweit, dass 100 % relative Feuchte erreicht wird, scheidet die Luft Wasser aus, der Taupunkt ist erreicht.

Dieser Taupunkt muss im Umfeld historischer organischer Materialien vermieden werden. Das herunterlaufende Wasser schädigt durch seine mechanische und adhäsive Wirkung insbesondere Farben und zellulosehaltige Werkstoffe.

Dadurch erkennt man bereits die zweite zu messende Größe: die *Luftfeuchte*. Sie bestimmt neben der Temperatur maßgeblich das Raumklima. Bei zu hoher Luftfeuchtigkeit besteht die Gefahr der Schimmelbildung auf den zu schützenden Objekten, hingegen eine zu tiefe Luftfeuchte z.B. Bücher austrocknen und ihre Papierbeschaffenheit verändern lässt. Die Luftfeuchte lässt sich durch analoge Sensoren zunächst prinzipbedingt nur als relative Feuchte bestimmen, mit der Einheit „Prozent relativer Feuchte“ (%rF). Sie gibt somit nur einen Faktor zur aktuell maximal möglichen Luftfeuchte an. Diese ist vom aktuellen Luftdruck abhängig.

Die absolute Feuchte kann mit der relativen Feuchte und dem *Luftdruck* errechnet werden. Es ist daher sinnvoll, auch ihn zu messen. Zudem gibt der Luftdruck Aufschluss über die derzeitige Wetterlage.

Die vierte physikalische Größe, die Einfluss auf die Alterung von historischen organischen Stoffen nimmt, ist ultraviolettes Licht (*UV-Licht*).

Der Sensor hierfür, rechts im Bild, hat seine spektrale Empfindlichkeit bei 220 bis 380nm Lichtwellenlänge.

Um eine gezielte Beeinflussung dieser vier Größen in der Klimatechnik vornehmen zu können, muss im ersten Schritt die Bestandsaufnahme des Raumklimas durchgeführt werden. Da sich die Tendenzen der Messgrößen und ihre Veränderung durch Beeinflussung nur im Vergleich zum Vor-Zeitraum erkennen lassen, ist es sinnvoll, den Klimaverlauf über längere Zeit zu verfolgen. Dabei kann ein Zeitraum von ein bis zwei Jahren als Richtwert dienen.

Nach den erfolgten Langzeitmessungen kann darüber entschieden werden, welchen Beitrag eine Gebäudeleittechnik mit integrierter Klimaregelung zur Werterhaltung der Gegenstände leisten kann. Es ist weiterhin zu überlegen, ob es möglich ist, das bestehende Mess-System nun vor Ort in die Leittechnik einzufügen oder den Klimaverlauf weiterer historischer Gebäude damit zu analysieren.

4. Sensorsignale aufnehmen, anpassen und digitalisieren

Aufgaben der Elektronischen Schaltung:

- Analoge Sensorsignale (Spannungen) dem Eingangsbereich des Analog-Digital-Wandlers anpassen
- Außenbeschaltung des A/D-Wandlers

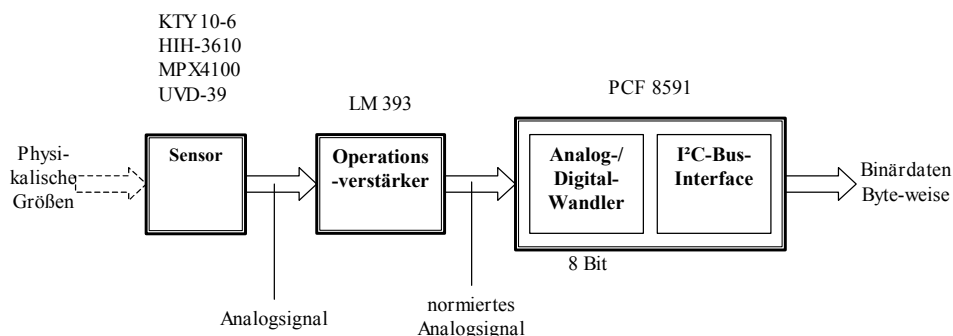


Bild: Signalfuss Sensormodul

(Bild oben) Die physikalischen Größen lösen in den Sensoren einen Stromfluss entsprechender Höhe aus. Diese analogen Signale werden nun in je einer Operationsverstärkerschaltung auf ein Spannungssignal mit den Grenzen 0V und 5V normiert. Die Veränderung des Offset und der Verstärkung ist erforderlich, um den Eingangsspannungsbereich des Analog-/Digital-Wandlers voll zu nutzen.

Der 4-Kanal-A/D-Wandler ist das nächste Glied in der Signal-Fluss-Kette. In ihm wird die analoge Eingangsspannung in einen äquivalenten Binärwert nach dem Verfahren der sukzessiven Approximation überführt. Ein Wandlungs-Zyklus des 8-Bit-A/D-Wandlers PCF8591 dauert maximal 90 μ s. Nach Abschluss der Wandlung wird ein Datenbyte mit dem Binärwert aus dem Datenregister des Wandlers auf den I²C-Bus gelegt. Dieses Byte wird dann vom Mikrocontroller IPC@CHIP eingelesen.

5. Entwicklung und Anfertigung von Sensormodulen

- Erstellung des Schaltplans
- Projektierung des Platinenlayouts
- Anfertigung der Platinen durch Fräsen
- Bestückung, der Platinen (Bild rechts)
- Test und Abgleich der Module

Es galt also, ein Sensormodul zu entwickeln, das die Sensoren und den Analog-Digital-Wandler integriert und die Signale zwischen ihnen konvertiert. Dazu wurden Schaltplan und Leiterplattenlayout erstellt.

Die Platinen wurden auf einer Portalfräsanlage angefertigt. Nach dem Bestücken der Platinen, erfolgten Test und Grobabweich der Module zusammen mit der Messsoftware.

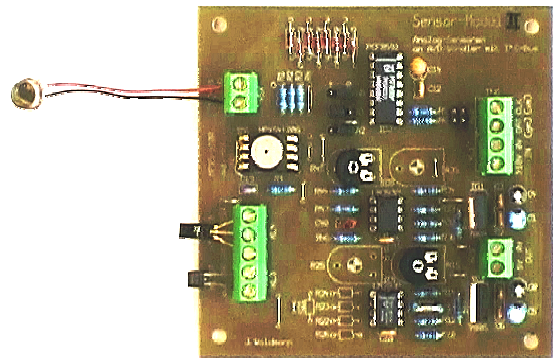


Bild: Sensormodul

6. Mikrocontrollersystem IPC@CHIP

Das eingesetzte Mikrocontrollersystem besteht aus drei wesentlichen Komponenten IPC@CHIP SC12, IPC@CHIP DK40 und Software @CHIP-RTOS.

Der SC12 vereint in einem 32-poligen Gehäuse:

- CPU / 20 MHz Takt
- je 500 kByte RAM / Flash-ROM
- parallele, serielle, I²C- und Ethernet – Schnittstellen

Das DK40 ist das Entwicklungsboard.

Es realisiert die Schnittstellenbeschaltung für den SC12.

Das ms-dos-ähnliche Betriebssystem @CHIP-RTOS komplettiert das System, so dass die im Blockschaltbild gezeigten Funktionen über ein z.B. in C geschriebenes Programm vom Entwickler benutzt werden können.

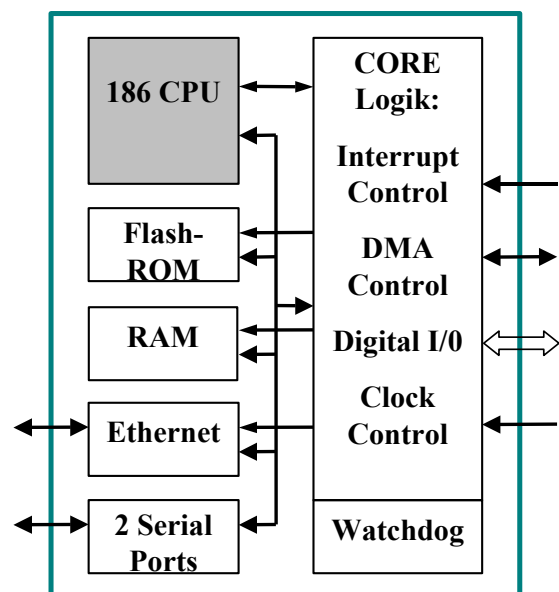


Bild: Chip-Architektur

7. Messprogramm des IPC@CHIP

Das in C geschriebene Messprogramm für den IPC@CHIP leistet eine sensorspezifische Verarbeitung der Messsignale in diesen Schritten:

- Anforderung der Wandlerwerte über den I²C-Bus
- Mittelwertbildung
- Plausibilitätsprüfung
- Polygonzug-Interpolation der Messwerte
- Dezentrale Archivierung
- Speicherung der Messdaten auf internem Flash-ROM

Die Mittelwertbildung wird über 20 Einzelmessungen in sehr kurzem Zeitabstand vorgenommen, damit der Quantisierungsfehler des Wandlers minimiert wird. Im Anschluss wird die Plausibilität geprüft. Um nun den Messwert zu erhalten, wird durch Polygonzug-Interpolation mit der Sensorkennlinie der Wert errechnet. Die Sensorkennlinie wird, sofern möglich, für jeden Sensor einzeln aufgenommen, und im Messprogramm verwendet. Wie eine solche Stützpunktkennlinie für den Temperatursensor aussieht, zeigt das Diagramm. Das Programm ordnet einem Eingangswert eine Temperatur entsprechend dieser Kennlinie zu. Des weiteren archiviert das Messprogramm die Messdaten auf internem oder externem Flash-ROM.

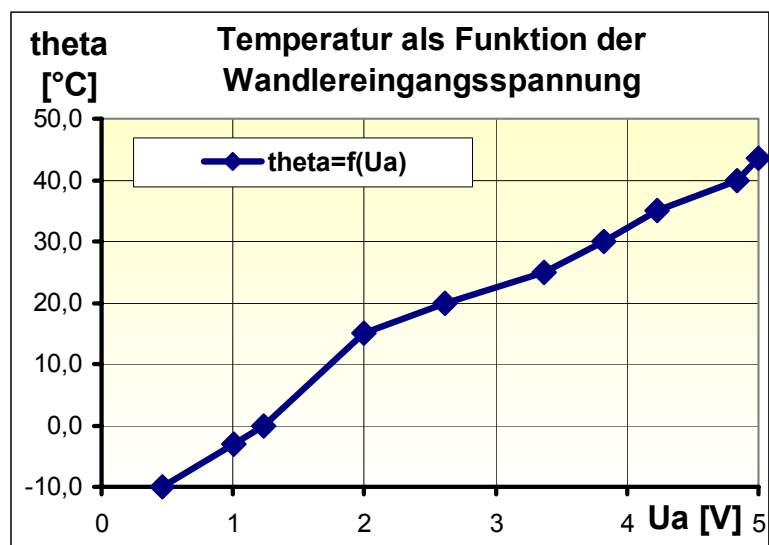
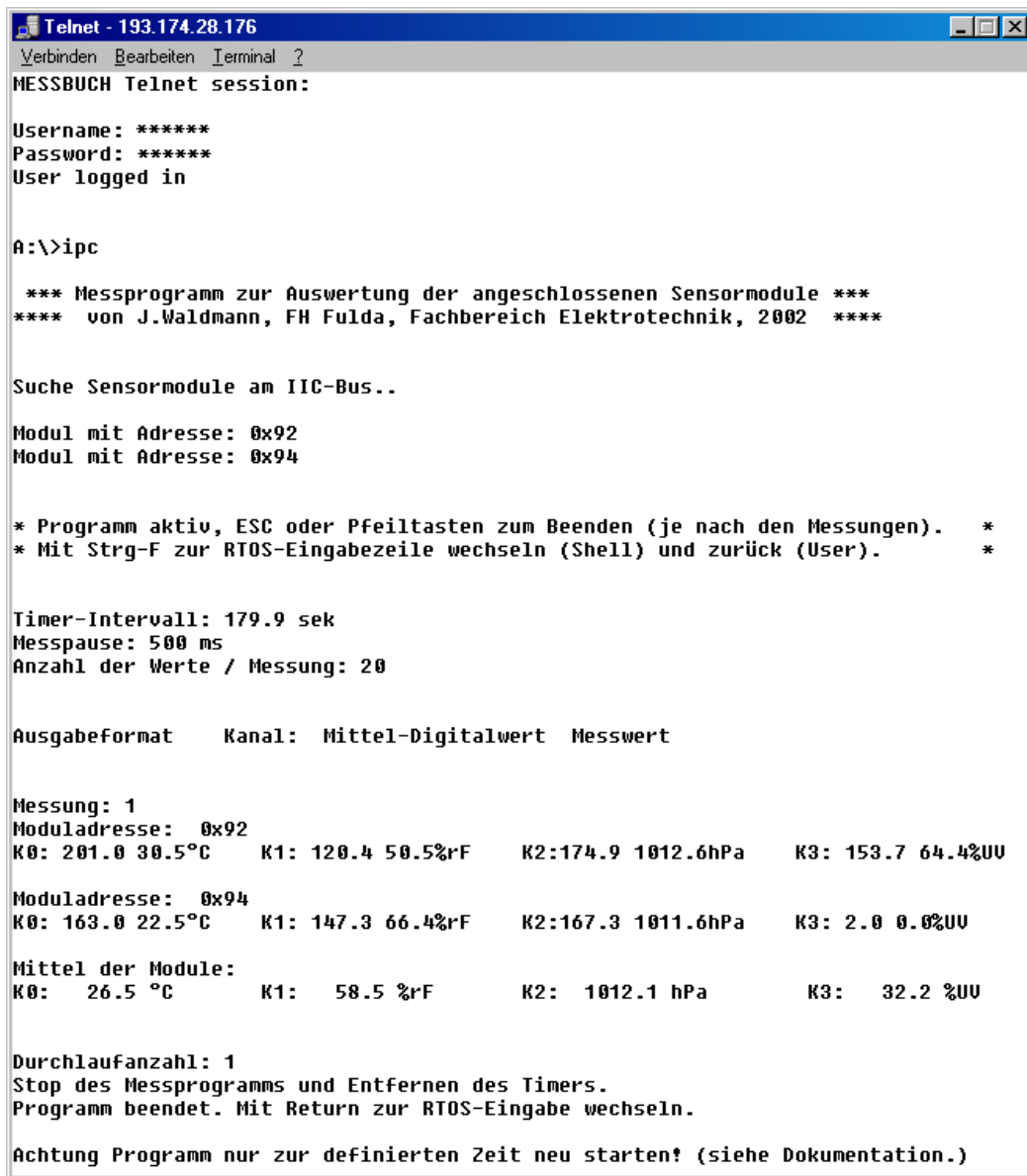


Bild: Sensor-Linearisierung

Zur effizienten Speicherausnutzung des Flash-ROM werden die Werte in je zwei Byte langen Ganzzahlen binär gespeichert. Pro Monat und Messgröße werden so rund 14600 Werte à 2 Byte Länge archiviert. Somit wächst das Archiv um 114 kByte pro Monat an.

Alle Ausgabe-Informationen des Programms für den Anwender werden auf der Standardausgabe in „Telnet“ ausgegeben. Das Bild zeigt die Bildschirmausgabe.



```
Telnet - 193.174.28.176
Verbinden Bearbeiten Terminal ?
MESSBUCH Telnet session:

Username: *****
Password: *****
User logged in

A:\>\ipc

*** Messprogramm zur Auswertung der angeschlossenen Sensormodule ***
**** von J.Waldmann, FH Fulda, Fachbereich Elektrotechnik, 2002 ****

Suche Sensormodule am IIC-Bus..

Modul mit Adresse: 0x92
Modul mit Adresse: 0x94

* Programm aktiv, ESC oder Pfeiltasten zum Beenden (je nach den Messungen). *
* Mit Strg-F zur RTOS-Eingabezeile wechseln (Shell) und zurück (User). *

Timer-Intervall: 179.9 sek
Messpause: 500 ms
Anzahl der Werte / Messung: 20

Ausgabeformat   Kanal: Mittel-Digitalwert  Messwert

Messung: 1
Moduladresse: 0x92
K0: 201.0 30.5°C   K1: 120.4 50.5%rF   K2:174.9 1012.6hPa   K3: 153.7 64.4%UU

Moduladresse: 0x94
K0: 163.0 22.5°C   K1: 147.3 66.4%rF   K2:167.3 1011.6hPa   K3: 2.0 0.0%UU

Mittel der Module:
K0: 26.5 °C       K1: 58.5 %rF      K2: 1012.1 hPa     K3: 32.2 %UU

Durchlaufanzahl: 1
Stop des Messprogramms und Entfernen des Timers.
Programm beendet. Mit Return zur RTOS-Eingabe wechseln.

Achtung Programm nur zur definierten Zeit neu starten! (siehe Dokumentation.)
```

Bild: Telnet-Ausgabe des Messprogramms

8. Messwertverläufe visualisieren

Der zweite große Teil dieses Projekts, war das Programmieren der Messwertvisualisierung in Java. Diese besitzt zum einen das Merkmal, dass der IPC@CHIP ein Webserver ist. Dadurch kann er die Webseite zur Verfügung stellen. Ebenso weitere Daten für ein Applet, hier die Messdaten, die ja auch dem IPC@CHIP gewonnen wurden.

Die Programmiersprache Java ist aus C++ entstanden, aber im Gegensatz dazu eine rein objektorientierte Sprache. In Java ist es nicht möglich, rein prozedurale Programme zu entwickeln, da es keine Abwärtskompatibilität zu einem Vorgänger geben muss, wie zum Beispiel von C++ auf C.

Ein zweites wesentliches Merkmal von Java ist Plattformunabhängigkeit. Java ist sowohl auf der Quelltextebene als auch auf Binärebene unabhängig von Rechnertyp und Betriebssystem. Beim Compilieren des Quelltextes einer herkömmlichen Programmiersprache wird ein Maschinencode nur für Hardware-Architektur-kompatible Prozessoren des jeweiligen Rechnertyps (z.B. Pentium) erzeugt. Für andere Rechner (z.B. Macintosh) muss der Quelltext mit einem dafür passenden Compiler erneut übersetzt werden, um ein ausführbares Programm zu erhalten. Ein Java-Programm durchläuft dagegen verschiedene Phasen, bevor es ein Anwender benutzen kann. Zuerst wird der Quelltext in einer Datei mit der Endung **.java** gespeichert. Diese Datei übersetzt der Java-Compiler in den plattformunabhängigen *Bytecode*. Jede seiner Code-Anweisungen ist genau ein Byte lang. Eine Bytecode-Datei trägt die Erweiterung **.class**, bzw.

.jar für komprimierte und zusammengefasste Dateien. Die **.class**-Dateien werden dann auf dem Anwender-Rechner von einem plattformabhängigen Bytecode-Interpreter ausgeführt. (Bild)

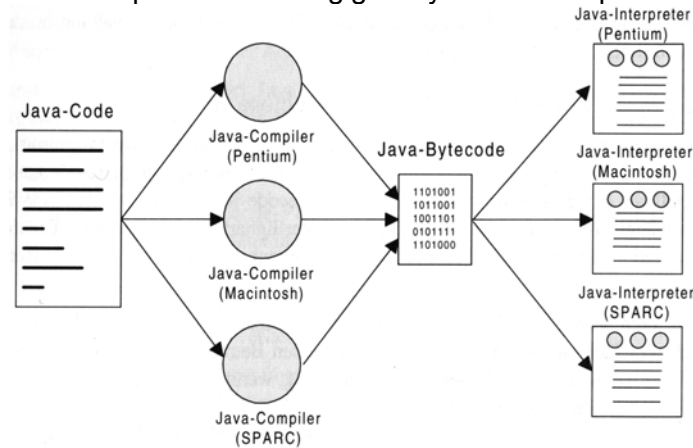


Bild: Plattformunabhängiger Bytecode in Java

Durch die Trennung von Compilierung und systemabhängiger Interpretierung ist eine Grundlage zur Verteilung von Java-Programmen über das Internet erfüllt. Der Nachteil der Trennung ist die geringere Ausführungsgeschwindigkeit. Der Download hingegen verläuft durch die geringere Größe der **.class**-Datei schneller ab.

IPC@CHIP ist ein Webserver:

- HTML-Webseiten und Bilder
- JavaScript und Java-Applets
- Daten für Java-Applets (hier: Messdaten)

Ein Ethernet-Client, also ein beliebiger Standard-Rechner im Netzwerk, führt die Visualisierung aus:

- Web-Browser ruft Webseite des IPC@CHIP über Ethernet auf
- dadurch wird das Java-Applet mit der Visualisierung geladen und gestartet (ebenfalls über Ethernet und TCP/IP)

9. Konzept der Visualisierung mit Java-Applets

- Java-Compiler erstellt universellen Bytecode
- Bytecode (Applet) wird über das Netzwerk geladen
- Im Browser integrierte JVM führt Applet (Visualisierung) aus
- Vorteil: keine spezielle Softwareinstallation für die Visualisierung

Das Wesen der plattformunabhängigen Visualisierung mit Java veranschaulicht das Bild.

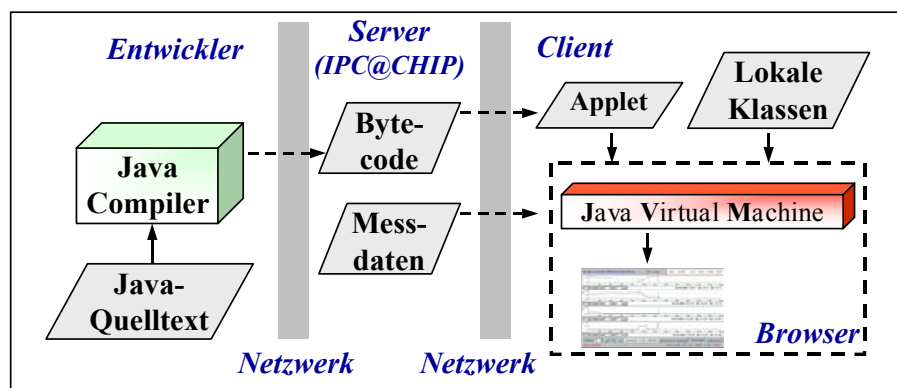


Bild: Konzept von Java

Links ist die Entwickler-Seite. Der Programmierer überträgt den kompilierten Quelltext auf den Web-Server. Der Server, in diesem Fall der IPC@CHIP enthält somit den Bytecode und die hier gewonnenen Messdaten. Beide gelangen über das Netzwerk Ethernet zum Rechner des Anwenders, dem Client. Zusammen mit lokalen Java-Bestandteilen wird das Applet durch die Java Virtual Machine im Internet-Browser des Anwenders ausgeführt.

Auf dem Anwenderrechner wird keine spezielle Software für dieses Applet benötigt. Dieser Vorteil wird durch geringere Ausführungsgeschwindigkeit innerhalb der Java Virtual Machine erkauft. Dies kommt hier aber nicht so sehr zum Tragen.

Alle Bestandteile, die zum Anzeigen der Webseite und Ausführen des Applets erforderlich sind lädt der Internet-Browser selbständig mit dem Protokoll HTTP. Dieses Protokoll der Anwendungsschicht des OSI-Modells hat eine einfache Struktur. Nach dem Aufbau der TCP-Verbindung durch den Web-Browser (HTTP-Client) mit dem HTTP-Web-Server, sendet der Client seine Anforderung an den Server. Diese Dateianforderung enthält typischerweise die Internet-Position und das Protokoll mit dem die Datei entgegengenommen werden soll. Nachdem der Web-Server seine Antwort in Form der Datei zurückgesendet hat, schließt einer der Teilnehmer die Verbindung.

10. Benutzeroberfläche der Visualisierung

Hier ist die vollständige Benutzeroberfläche des Visualisierungs-Applets gezeigt. Die zentralen Bestandteile sind die vier Messwertdiagramme von Temperatur, Feuchte, Luftdruck und UV-Intensität. Die Navigationsschaltflächen wirken auf alle vier Diagramme gleichermaßen. Die Skalierung der Wertebereiche kann dagegen für jede Messgröße separat eingestellt werden. Wichtige Kennzahlen sind in numerischer Form angezeigt. Dies sind Minimum, Maximum und Mittelwert des dargestellten Abschnitts. Der numerische Wert jedes Punktes der Graphen lässt sich über einen Mausklick aufrufen, mit der zugehörigen Zeit.

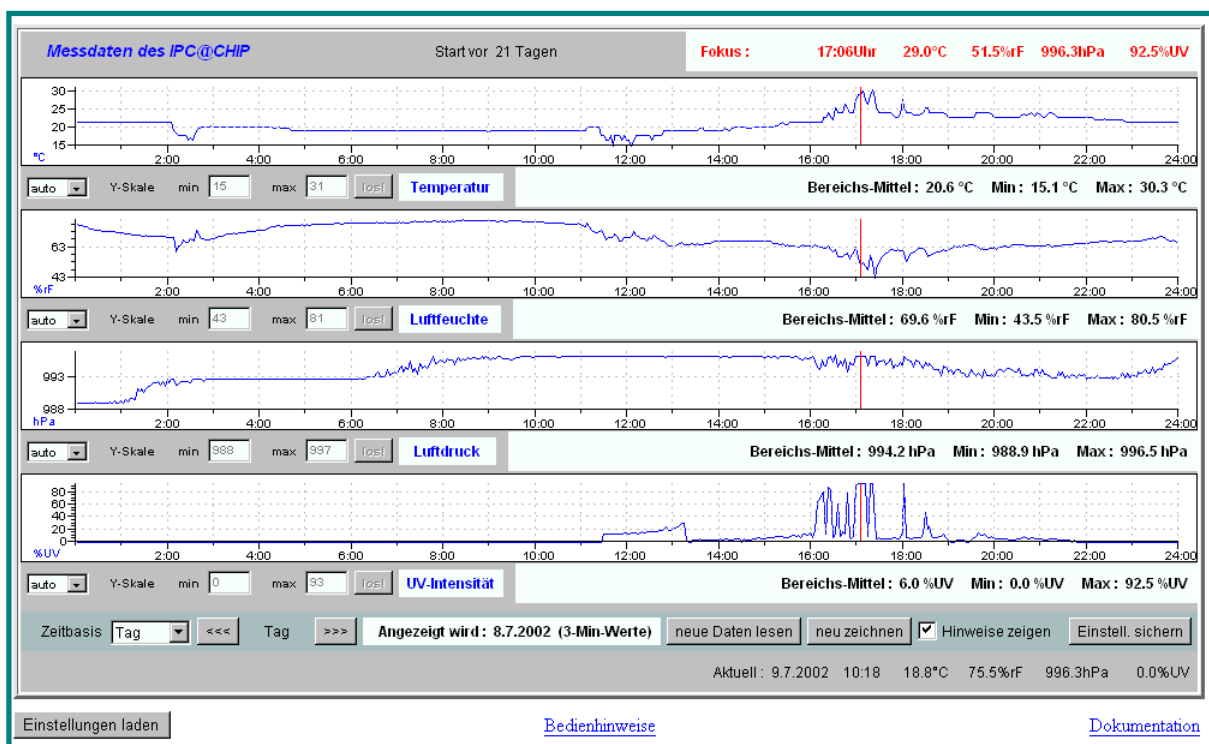


Bild: Visualisierungsoberfläche

Die Bedienoberfläche ist in drei Hauptbereiche untergliedert.

- Der obere Teil enthält nur weniger wichtige Anzeigen und keine Bedienelemente.
- Im mittleren Bereich, der weit mehr als $\frac{3}{4}$ der Gesamt-Bildschirmfläche einnimmt, sind die vier Wertverläufe angezeigt. Jedem Diagramm sind kanalbezogene Ein- und Ausgabeelemente zugeordnet.
- Kanalübergreifende Schaltflächen und Informationen sind schließlich im unteren Bildschirmteil platziert.

11. Das Messgerät im Überblick

Das Bild zeigt die Hardware des Messsystems, wie sie in ein Buch eingepasst ist. Oberhalb des IPC@CHIP-Mikrocontroller ist die Adapter-Platine der Compact-Flash-Karte zu sehen. Das untere Bild enthält das zusätzliche Sensormodul. Auf der linken Seite sind die Sensoren für *UV-Intensität*, *Luftdruck*, *Luftfeuchte* und *Temperatur* (von oben nach unten). Alle Sensoren sind steckbar oder schraubbar, um bei Bedarf auch außerhalb des Buches platziert werden zu können. (Bild)

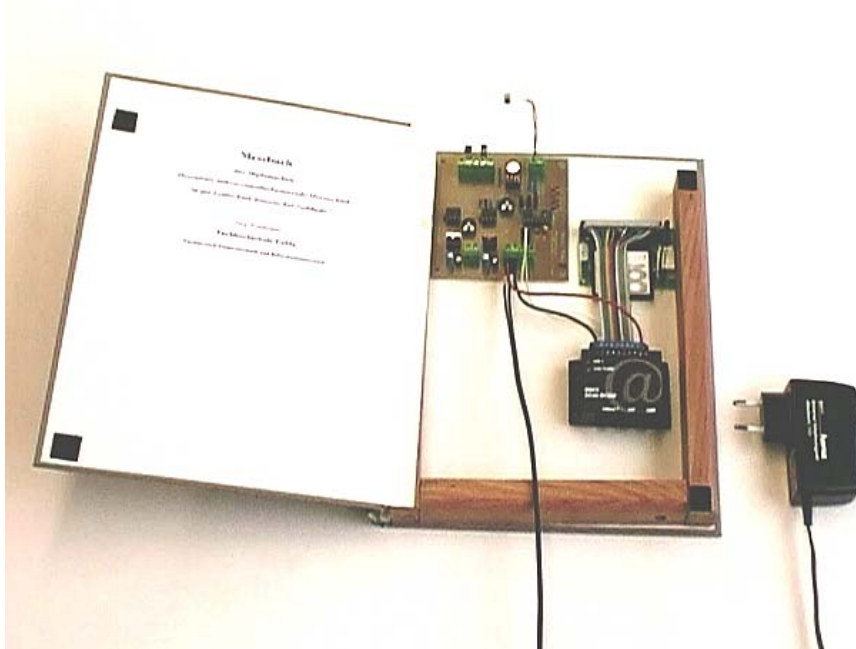


Bild: IPC@CHIP und Sensormodul im Buch integriert

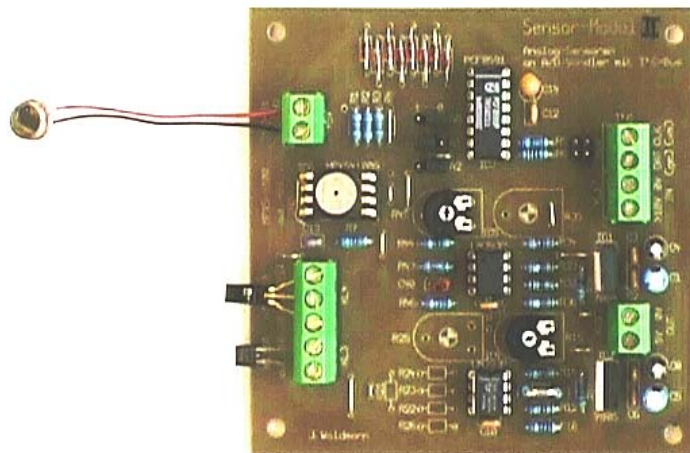


Bild: Bestücktes Sensormodul

12. Administration des Messsystems über Ethernet mit TCP/IP

Ein weiterer Gesichtspunkt für den Einsatz des Ethernet, ist die vollständige Administration des Messsystems über das Ethernet mit TCP/IP.

Es sind im Wesentlichen diese fünf Kommunikationsformen mit dem IPC@CHIP möglich.

- Starten, Beenden und Bildschirmausgaben des Messprogramms in *Telnet*
- Aktualisieren des Betriebssystems über *UDP*
- Datensicherung und –Wiederherstellung über *FTP*
- Prüfung der Sensormodule über Telnet mit Diagnoseprogramm
- Dokumentation des Messsystems im *Browser (HTTP)*

13. Verwendung des Messsystems in der Gebäudeleittechnik

In erster Linie ist das Messsystem als unabhängig von der Gebäudeleittechnik mit Temperatur- und Klimaregelungen sowie Belüftungssteuerungen konzipiert. Das bedeutet, dass kein Eingriff in eventuell vorhandene Steuerungs- und Regelungsaufgaben erfolgt. Somit handelt es sich zunächst um ein „Stand-alone“-System. Es wird davon ausgegangen, dass die zu analysierenden Räume bisher über keine derartigen Einrichtungen verfügen. Erst im zweiten Schritt nach der Klima-Untersuchung ist der Einsatz von Regelungssystemen im Sinne des Inventarschutzes abzuwägen.

Nach der Analyse bieten sich prinzipiell zwei Möglichkeiten, das Messsystem mit dem IPC@CHIP in die Gebäudeleittechnik einzubinden.

- Das Sensormodul ist so konzipiert, dass die aufbereiteten Sensor-Analogsignale von einem externen A/D-Wandler übernommen werden können. Der Mikrocontroller und die Visualisierung würden dann hier nicht mehr benötigt. Es ist ebenso möglich Sensormodule neu anzufertigen, weil alle notwendigen Daten, speziell Schaltplan und Platinenlayout vorhanden sind.
- Eine andere Form der Eingliederung in die Gebäudeleittechnik führt über das Ethernet. Die Regelungssysteme der Leittechnik werden in einer Software-SPS ausgeführt, die z.B. auf einem Industrie-PC läuft. Der Mikrocontroller kommuniziert über das Ethernet mit dem Industrie-PC und liefert so die Messwerte für die SPS. Das Messprogramm muss hierzu um die TCP/IP-Kommunikation erweitert werden.

Die favorisierte Nutzung des Messsystems nach Abschluss der Klima-Analyse ist aber die Weitergabe des gesamten Messsystems in andere historische Gebäude. Auch hier kann damit eine unkomplizierte Bestandsaufnahme der Umweltparameter durchgeführt werden.

14. Ergebnis und Weiterentwicklung

- Messsystem ist in Hard- und Software funktions- und einsatzfähig
- Exemplarisch wurde das Messsystem in ein Buch integriert, um einen unmittelbaren Einsatz in einer historischen Bibliothek zu ermöglichen.
- Erste Analyse der Klimaverhältnisse in der „[Herzogin-Anna-Amalia-Bibliothek](#)“ in Weimar seit 26. August 2002 mit diesem „Messbuch“.

Eine **Weiterentwicklung** kann mit folgenden Ansätzen erfolgen.

- Fehlerprotokollierung des Messprogramms sinnvoll
- Druck- und UV-Messkanäle noch genau abzugleichen
- Warnung bei Grenzwert-Verletzung per E-Mail etc.
- Messwert- und Systemstatus-Anzeige vor Ort auf Display