



**Entwicklung eines autonomen Low-Cost-Messclients für geodätische
Überwachungsmessungen**

Masterarbeit

im Studiengang Geodäsie & Geoinformatik

zur Erlangung des akademischen Grades
Master of Engineering

eingereicht von:
Philipp Engel

Erstprüfer: Prof. Dr.-Ing. Karl Foppe
Zweitprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig

urn:nbn:de:gbv:519-thesis2013-0128-9

August 2013

Kurzzusammenfassung

Ein wichtiges Aufgabenfeld der Ingenieurvermessung ist die Überwachung von Bauwerken und Geländeabschnitten mit Hilfe geodätischer und geotechnischer Sensoren. An der Hochschule Neubrandenburg wurde dazu das „Datenbank-orientierte Monitoring- und Analyse-System“ (DABAMOS) entwickelt, das als freie und herstellerunabhängige Lösung konzipiert ist. Diese Arbeit befasst sich unter Einsatz des Monitoring-Systems mit der hard- und softwareseitigen Entwicklung eines autonom agierenden Messclients, der die Ansteuerung von Sensoren und die Übertragung von Messdaten auf Internet-Server übernehmen soll.

Abstract

The structural monitoring of buildings and terrain by means of geodetical and geotechnical sensors is an imported task field of modern surveying engineering. The free and open-source monitoring system DABAMOS has been developed at the Hochschule Neubrandenburg, and is neither limited to specific sensors nor to particular manufacturers. This master's thesis deals with the development and implementation of an autonomous working client computer for measuring and data processing purposes, including the control of sensors and the transmission of measurement result to internet servers, using the DABAMOS software.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Problemstellung	2
1.3. Typografische Konventionen	2
2. Geodätische Überwachungsmessungen	3
3. Sensorik	6
3.1. Geodätische Sensoren	6
3.1.1. GNSS-Empfänger	6
3.1.2. Digitalnivelliere	7
3.1.3. Tachymeter	8
3.2. Geotechnische Sensoren	9
3.2.1. Beschleunigungssensoren	9
3.2.2. Dehnungsmessstreifen	10
3.2.3. Extensometer	11
3.2.4. Faseroptische Sensoren	11
3.2.5. Inklinometer	11
3.2.6. Schlauchwaagen	12
3.3. Meteorologische Sensoren	12
4. Systeme für geodätische Überwachungsmessungen	13
4.1. Datenbank-orientiertes Monitoring- und Analyse-System	13
4.1.1. Software	14
4.1.2. Hardware	15
4.2. Weitere Monitoring-Systeme	15
4.2.1. GNSS/LPS/LS-based Online Control and Alarm System	17
4.2.2. Leica GeoMoS	17
4.2.3. Monitoring System der Technischen Universität München	18
4.2.4. Trimble 4D Control	18
5. Autonomer Low-Cost-Messclient	19
5.1. Anforderungen	19
5.1.1. Mechanik und Widerstandsfähigkeit	20
5.1.2. Leistung	20
5.1.3. Speichermedien	21
5.1.4. Echtzeitfähigkeit	21
5.1.5. Stromversorgung	22
5.1.6. Kosten	22

5.2. Hardware-Plattformen	23
5.2.1. Industrie-Computer	23
5.2.2. Mikrocontroller	23
5.2.3. Einplatinen-Computer	24
6. Hardware	25
6.1. Einplatinen-Computer	25
6.1.1. Arduino	26
6.1.2. BeagleBoard und BeagleBone	28
6.1.3. Carambola	29
6.1.4. Cubieboard	30
6.1.5. GNUBLIN	30
6.1.6. PandaBoard	31
6.1.7. Raspberry Pi	32
6.2. Kommunikation	33
6.2.1. Bluetooth	33
6.2.2. Ethernet	35
6.2.3. Free Space Optics	35
6.2.4. General Purpose Input/Output	36
6.2.5. Mobiles Internet	37
6.2.6. RS-232	39
6.2.7. USB	40
6.2.8. WLAN	41
6.2.9. ZigBee	42
6.3. Stromversorgung	43
6.3.1. Netzgerät	43
6.3.2. Unterbrechungsfreie Stromversorgung	44
6.3.3. Photovoltaik	47
6.3.4. Power over Ethernet	49
6.4. Gehäuse	49
7. Software	52
7.1. Betriebssystem	52
7.1.1. Linux	52
7.1.2. FreeBSD	54
7.1.3. NetBSD	54
7.1.4. RISC OS	55
7.2. Programmiersprache	56
7.3. Datenbank	56
8. Praktische Umsetzung	58
8.1. Hardware	58
8.1.1. Einplatinen-Computer	58
8.1.2. Kommunikation	58
8.1.3. Stromversorgung	59
8.1.4. Gehäuse	60

8.2. Software	61
8.2.1. Betriebssystem	61
8.2.2. Messclient-Software	64
8.3. Sensoransteuerung	66
8.3.1. Neigungssensor	66
8.3.2. Tachymeter	70
8.3.3. Virtueller Sensor	71
9. Überwachungsmessungen	76
9.1. Neigungssensor	76
9.2. Tachymeter	77
9.2.1. Überwachung von Messpfeilern im Labor	77
9.2.2. Überwachung von Bauwerken auf dem Campus der Hochschule	78
10. Auswertung und Ausblick	79
10.1. Hardware	79
10.1.1. Einplatinen-Computer	79
10.1.2. Unterbrechungsfreie Stromversorgung	80
10.1.3. Gehäuse	80
10.2. Software	81
10.2.1. Programmiersprache	81
10.2.2. Datenbank	82
10.3. Überwachungsmessungen	82
10.3.1. Neigungsmessung	82
10.3.2. Überwachung von Messpfeilern im Labor	83
10.3.3. Überwachung von Bauwerken auf dem Campus der Hochschule	83
A. Anhang	85
A.1. Abbildungen	85
A.2. Diagramme	86
A.3. MATLAB-Quelltext	88
B. Eidesstattliche Erklärung	90
Abbildungsverzeichnis	91
Tabellenverzeichnis	93
Literaturverzeichnis	94

1. Einleitung

Der Mensch errichtet seit mehr als dreitausend Jahren große Bauwerke. Bereits in der Antike wurden mit dem Bau der sieben Weltwunder beeindruckende Konstruktionen erschaffen. Viele von ihnen sind schon vor langer Zeit durch Naturkatastrophen zerstört worden, weil sie z. B. in tektonisch ungünstiger Lage oder in direkter Umgebung von fließenden Gewässern gebaut wurden. In der Gegenwart nehmen die Größe moderner Ingenieurbauwerke und der Eingriff in die Natur immer weiter zu. Riesige Brücken, mit mehr als 200 Meter hohen Pylonen, gewaltige Staumauern oder Tagebaue, die sich über mehrere Quadratkilometer erstrecken, sind nur einige Beispiele. Die Bauwerke werden aber nicht nur größer, auch ihr Alter und die Dichte ihrer Bebauung nimmt zu. Diese Faktoren führen zu erwarteten aber auch ungeahnten Folgen für Mensch und Natur. In Bergbauregionen sinkt das Grundwasser und der Boden setzt sich, aufgeschüttete Inseln versinken unter dem Druck der auf ihnen errichteten Gebäude und historische Bauwerke werden durch Bauarbeiten in der nahen Umgebung beschädigt und stürzen ein. Die Ingenieurgeodäsie ist heute in der Lage, nahezu alle Arten von Bewegungen zu erfassen und Objekte über längere Zeiträume hinweg zu beobachten, um Vorhersagen über das weitere Verhalten zu treffen.

Die Kipprutschungen im Braunkohlewerk Nachterstedt, der Einsturz des Kölner Stadtarchivs oder die Steilküstenabbrüche auf Rügen haben in den letzten Jahren gezeigt, dass ein Bedarf an fortschrittlichen ingenieurgeodätischen Monitoring-Lösungen besteht, die kleinste Bewegungen, die zu Beschädigungen oder Einstürzen führen können, aufdecken und das zukünftige Verhalten von Bauwerken vorhersagen können. Moderne Überwachungssysteme werden von verschiedenen Herstellern angeboten. Sie können nicht nur Sensoren ansteuern und Messwerte speichern, sondern auch Daten verarbeiten, analysieren und auf andere Rechnersysteme übertragen. Im Gefahrenfall versenden sie automatisch Alarmmeldungen und warnen den Benutzer. Etablierte Lösungen sind aber kostenintensiv und nutzen aktuelle wissenschaftlich-technische Entwicklungen, wie Cloud-Speicherdiensste oder Low-Cost-Sensoren, noch nicht in vollem Umfang. Es ist absehbar, dass fortschrittliche Technologien im Bereich automatischer Messsysteme in den nächsten Jahren auch in der Ingenieurgeodäsie Verbreitung finden werden.

1.1. Motivation

Das „Datenbank-orientierte Monitoring- und Analyse-System“ (DABAMOS) wird seit 2009 im Studiengang Geoinformatik der Hochschule Neubrandenburg entwickelt. Mit DABAMOS können geodätische Überwachungsmessungen an Gebäuden und Topografie durchgeführt werden, um Höhen- und Lageveränderungen im Submillimeter- bis Zentimeterbereich festzustellen und zu analysieren. Das System kann überall dort eingesetzt werden, wo Bewegungen zu erwarten sind, darunter Großbaustellen, Kirchen, Brücken oder Rutschhänge.

Die gemessenen Daten werden gespeichert, auf Internet-Server übertragen und anschließend automatisch analysiert, um Aussagen über das zukünftige Verhalten der Objekte treffen zu können. Dieses System soll im Rahmen dieser Arbeit um einen autonom agierenden Mess- und Steuerrechner erweitert werden.

1.2. Problemstellung

Das Überwachungssystem DABAMOS wurde in der Vergangenheit beim Bau des neuen Schiffshebewerks Niederfinow Nord ersten Praxistests unterzogen. Für die Ansteuerung der Sensoren, der lokalen Zwischenspeicherung der Messdaten und deren Übertragung zu Internet-Server kamen dabei konventionelle Personal Computer zum Einsatz. Diese eignen sich nur bedingt für den Betrieb im Außenbereich und weisen viele Nachteile auf, darunter ihre Größe und ihr Gewicht, ihre hohe Leistungsaufnahme und ihr fehlender Schutz gegenüber äußeren Einflüssen. Ein autonomer Betrieb des Monitoring-Systems über einen längeren Zeitraum wird dadurch erschwert oder unmöglich gemacht.

Aus diesem Grund soll ein speziell für das Aufgabenfeld der ingenieurgeodätischen Überwachungsmessungen ausgelegter Messclient entwickelt werden, der einerseits die gestellten technischen Anforderungen erfüllt, und andererseits bei möglichst geringen Kosten produziert und betrieben werden kann. Dazu sind die erforderlichen Komponenten zu evaluieren und die Software auf diese abzustimmen. Im Anschluss ist das System bei experimentellen Versuchen und Testmessungen auf Funktionalität zu prüfen.

1.3. Typografische Konventionen

In dieser Masterarbeit werden einige typografische Konventionen angewendet, die das Verständnis des Inhalts unterstützen sollen und nachfolgend aufgelistet sind.

- *Kursiv* gesetzt werden
 - Begriffe, die definiert werden,
 - Produktbezeichnungen,
 - Software-Applikationen und -Bibliotheken sowie
 - fremdsprachige Wörter.
- In **nichtproportionaler Schrift** gesetzt werden
 - Internetadressen,
 - Dateinamen,
 - Quelltext und
 - Befehle.

2. Geodätische Überwachungsmessungen

Die Ingenieurgeodäsie begann schon frühzeitig mit dem Nachweis geometrischer Veränderungen an Bauwerken. Erste Deformationsmessungen wurden bei der Errichtung von Staudämmen zu Beginn des 20. Jahrhunderts durchgeführt [WELSCH U. A., 2000]. Bei der Konstruktion konnte noch nicht auf Vergleichswerte zurückgegriffen werden, weshalb diese Objekte eine genaue Beobachtung und Analyse erforderten. Automatisch arbeitende Messsysteme waren damals ebenfalls noch nicht verfügbar. Die Messungen wurden dementsprechend in großen Intervallen von Wochen bis Jahren durchgeführt. Die Erfassung der geometrischen Veränderungen am Bauwerk erfolgte durch den Vergleich der Koordinaten verschiedener Epochen. Dieser klassische Ansatz ist auch heute noch verbreitet. Im Zuge der Entwicklung automatisierter Messsysteme und moderner Sensoren mit hohen Frequenzen kam es aber zu einem Übergang von der epochenweisen zur zeitkontinuierlichen Erfassung von Veränderungen. [EICHHORN, 2012]

Bauwerke werden heute aber nicht nur während der Bauphase sondern auch nach ihrer Errichtung überwacht. Der Einsturz eines Gebäudes kündigt sich häufig schon lange vorher durch Verformungen, Risse oder Setzungen an. Diese geometrischen Veränderungen können als Grundlage für Analysen und Prognosen zum frühzeitigen Erkennen und Beurteilen von Gefahrensituationen dienen. Darüber hinaus existieren noch weitere Aufgaben geodätischer Überwachungsmessungen. Deren Zielsetzung lässt sich nach [WELSCH U. A., 2000] wie folgt zusammenfassen:

- Bestimmung von geometrischen Veränderungen;
- Nachweis der Funktions- und Standsicherheit des Messobjektes und die Gewährleistung eines störungsfreien Betriebes;
- rechtzeitige Erfassung von Veränderungen, um Gefährdungen für das Messobjekt und seine Umgebung zu vermeiden oder zu mindern;
- Schadensdokumentation und die Klärung der Ursache von Schäden;
- Prognose des anzunehmenden Verhaltens in der näheren Zukunft und des Verhaltens unter bestimmten Bedingungen;
- Überprüfung von Konstruktions- und Materialeigenschaften;
- Erkenntnisgewinn bei vergleichbaren Messobjekten.

Bei Überwachungsmessungen wird grundsätzlich zwischen den Bereichen Bauwerks- und Geo-Monitoring unterschieden. Das Bauwerks-Monitoring umfasst vor allem Ingenieurbauwerke; aber auch historische Sakralbauten, wie Kirchen oder Klöster, sind häufig Gegenstand überwachender Messungen. Unter Geo-Monitoring werden hingegen alle jene Messverfahren

Zweck der Messung	Beispiele	Art der Bewegung	Größenordnung der Bewegung	Anforderung an ... ¹				
				Latenzzeit	Messzeit (Epoche)	Synchronisation	Wiederholungsrate	Genaugigkeit
Beschreibung von Veränderungen	Geotektonik	kriechend, abrupt	0–20 mm pro Jahr	0	1	1	1	5
	Gletscherbewegung	gleichförmig	0–10 m pro Jahr	0	1	1	1	2–4
	Gezeiten	kontinuierlich	bekannt	0	3	2	3	1–5
	Materialtests, Instrumentenuntersuchung	kriechend – abrupt plastische Verformung	< 100 mm/s	0	3–5	3–5	3–5	4–5
Stand- und Funktionssicherheit von Bauwerken und Geländeabschnitten	Talsperren, Brücken, Hochhäuser, usw.	elastisch; abhängig von Zeit, Füllgrad, ...	0–20 mm pro Jahr	3	2	2	2	2–5
	Hangrutschungen	kriechend – Rutschung, Felsabbruch, usw.	0–50 mm pro Jahr	4	3	3	3	2–5
Prozesssteuerung	Steuerung einer Baumaschine	gleichförmig	0–10 m/s	5	5	4	5	3–5

¹ Grad der Anforderung: Skalierung reicht von keine Anforderung (0) bis höchste Anforderung (5) an das jeweilige Kriterium.

Tabelle 2.1.: Kategorien der Messaufgaben [STAIGER, 2009]

verstanden, die sich mit Veränderungen der Umwelt beschäftigen. Dazu zählen die Atmosphäre, die Lithosphäre und die Hydrosphäre. Zu den verbreiteten Objekten, die auf Deformationen untersucht werden, gehören:

- Ingenieurbauwerke
 - Brücken
 - Hochhäuser
 - Stauanlage, Talsperren
 - Tunnel
- Geländeabschnitte
 - Rutschhänge
 - Steilküsten
 - Netze
 - Baustellen
 - Tagebaue

Die Aufgaben der Messverfahren ergeben sich bei beiden Bereichen aus den Anforderungen der beobachteten Objekte. Die Aufgaben können nach [STAIGER, 2009] in drei Kategori-

en unterteilt werden: Beschreibung, Warnung und Steuerung. Jeder Kategorie lassen sich verschiedene Anwendungen zuordnen, die in Tabelle 2.1 dargestellt sind.

3. Sensorik

Für die Überwachung von Bauwerken und Geländeabschnitten werden viele geodätische, geotechnische und meteorologische Sensoren eingesetzt. In zunehmenden Maße erfolgt eine Kombination der Sensoren miteinander. Bei Erfassung von Bauwerksbewegungen sind sowohl zeitliche als auch räumliche Veränderungen von Interesse. Die zeitlichen Veränderungen umfassen Vibrationen und Schwingungen im Bereich einiger hundertstel Sekunden bis mehreren Sekunden bei Brückenbauwerken sowie langfristige Bewegungen über Monate und Jahre bei Bauwerkssetzungen. Die von Sensoren gemessenen räumlichen Veränderungen reichen von kleinen lokalen Bereichen (Fugen, Risse), über einzelne Bauabschnitte (Wände, Pfeiler) bis hin zu gesamten Bauwerken. Darauf hinaus werden mit Tachymetern und GNSS-Empfängern aber auch großräumige Netze überwacht. [EICHHORN, 2012]

In diesem Kapitel wird eine Auswahl von häufig bei Überwachungsmessungen zum Einsatz kommende geodätische, geotechnischen und meteorologische Sensoren betrachtet. Die Ausführungen sind als eine Einleitung zu verstehen und decken bei weitem nicht das ganze Spektrum an verfügbaren und praktisch verwendeten Messverfahren ab. Zu den weiteren Verfahren, die auch für Deformationsanalysen und Überwachungsmessungen genutzt werden, gehören u. a. Laserscanning, Radarinterferometrie und Photogrammetrie.

3.1. Geodätische Sensoren

Unter geodätischen Sensoren werden diejenigen Instrumente zur Messung von Richtungen, Winkel, Strecken und Höhenunterschieden verstanden, die auch in anderen Bereichen der geodätischen Praxis, wie der Landesvermessung oder der Liegenschaftsvermessung, eingesetzt werden [PELZER U. A., 1988]. Einige von ihnen eignen sich besonders für periodische und für Schwingungsmessungen (Tabelle 3.1). Im Folgenden soll näher auf sie eingegangen werden.

3.1.1. GNSS-Empfänger

Globale Navigationssatellitensysteme (GNSS) sind eine Standardmessmethode in der Geodäsie und kommen auch bei Überwachungsmessungen zum Einsatz. Aufgrund der Genauigkeitsanforderungen werden nur differentielle Verfahren eingesetzt, die Koordinatendifferenzen zwischen zwei oder mehr Stationen liefern. Ein Großteil der wirksamen Fehlereinflüsse lässt sich dadurch eliminieren.

Auf einer Referenzstation mit bekannten Koordinaten im WGS 84 wird dazu ein *GNSS-Empfänger* installiert. Alternativ kann auf Satellitenreferenzdienste zurückgegriffen werden.

Instrument	Messgenauigkeit	Messfrequenz	Periodische Messungen	Schwingungsmessungen
Digitalnivellier	< 1 mm (h)	< 0,3 Hz	++	○
Tachymeter	2 – 3 mm (X, Y, h)	< 10 Hz	++	+
GNSS-Empfänger	2 – 10 mm (X, Y, h)	< 20 Hz	++	+
Neigungssensor	< 0,01 mrad (γ)	< 10 Hz	++	+
Beschleunigungs- sensor	< 10 $\mu\text{m}/\text{s}^2$ (a)	< 500 Hz	○	++

Tabelle 3.1.: Beim Monitoring von Bauwerken eingesetzte Verfahren mit ihren Eigenschaften [RESNIK, 2010]

Aus der Differenz zwischen bekannten und berechneten Koordinaten lassen sich Entfernungs korrekturen bestimmen, die anschließend bei der Positionsbestimmung der überwachten Punkte verwendet werden. Dadurch können Genauigkeiten im Zentimeterbereich realisiert werden. Durch die Auswertung der Trägerphase sind aber auch Genauigkeiten von unter einem Millimeter möglich. Einflussgrößen, darunter Mehrwegeeffekte, Signalqualität und Antennenoffset, wirken sich auf das Ergebnis der Positionsbestimmung aus und beeinflussen sowohl die Referenzstation als auch einzelne Überwachungsstationen. Bei der Auswertung langer Zeitreihen können diese Effekte analysiert werden. [ZEIMETZ U. A., 2009]

Im Vergleich zum Tachymeter sind keine Sichtverbindungen zwischen den Punkten erforderlich. Ein weiterer Vorteil ist die Auswertung sehr langer Basislinien, wodurch die Überwachung größerer Flächen ermöglicht wird. Jeder Punkt benötigt aber einen eigenen GNSS- Empfänger mit Antenne (Abbildung 3.1).

Für die Integration in Monitoring-Systeme wird überwiegend auf die NMEA-Protokolle gesetzt. GNSS-Netzwerklösungen, z.B. Leica GNSS Spider, ermöglichen ein automatisches Post-Processing von RINEX-Rohdaten. Die Auswertungsergebnisse sind dann in Echtzeit verfügbar. Externe Monitoring-Systeme können über NMEA-Streams eingebunden werden.

In den letzten Jahren wurden auch Untersuchung hinsichtlich der Integration von Low-Cost-GNSS-Lösungen durchgeführt, um die Kosten bei Überwachungsmessungen zu senken. [STEMPFHUBER U. ALBERDING, 2012] Einfache Low-Cost-Empfänger, wie sie auch in Navigationsgeräten und Mobiltelefonen zum Einsatz kommen, sind bereits seit einiger Zeit in Form einzelner Steckmodule erhältlich. Diese erreichen aber, je nach Anzahl der verfügbaren Satelliten, nur eine Lagegenauigkeit von 5 bis 15 m. Für ingenieurgeodätische Aufgabenstellungen ist diese Genauigkeit zu niedrig. Mit Piksi wird gegenwärtig ein Low-Cost-Empfänger für die Echtzeitkinematik entwickelt, der nach Herstellerangaben eine Lagegenauigkeit von 1 cm erreichen kann (Abbildung 3.2). [SWIFT NAVIGATION INC., 2013]

3.1.2. Digitalnivelliere

Nivelliere werden für die Messung von Höhenunterschieden zwischen Punkten und Bezugsflächen eingesetzt. Nach der Erfindung von CCD-Sensoren kamen Anfang der 1990er Jahre



Abbildung 3.1.: Die GNSS-Antenne Leica AS10 unterstützt GPS, GLONASS, Galileo, Compass und SBAS und wird für einzelne Referenzstationen oder RTK- und Monitoringnetze verwendet

erste *Digitalnivelliere* auf den Markt, die optisch codierte Nivellierlatten automatisch ablesen können. Die im Instrument integrierte CCD-Zeilenkamera registriert das horizontal angezielte Code-Fragment der Nivellierlatte. Ein Prozessor berechnet anschließend den binärcodierten Höhenwert durch Korrelation mit der gespeicherten vollständigen Codefolge.

Aktuelle Digitalnivelliere bieten keine Möglichkeit, Messungen fortlaufend und autonom durchzuführen. Viele Instrumente verfügen aber über eine serielle Schnittstelle für die Ansteuerung und den Datenaustausch. Die Messungen können so von einem angeschlossenen Rechner ausgelöst werden, wodurch sich Höhenüberwachungen in festen Zeitabständen realisieren lassen. [DEUMLICH U. STAIGER, 2002] Nivelliere haben den Nachteil, dass die Messungen auf eine Latte beschränkt sind. Es wurden in der Vergangenheit aber Versuche unternommen, durch Motorisierung des Instruments auch mehrere Latten anzuzielen.

3.1.3. Tachymeter

Tachymeter sind Sensoren mit elektronischer Strecken- und Richtungsmesseinrichtung und dienen der gleichzeitigen Erfassung von Lage und Höhe dreidimensionaler Punkte im Raum. Von einem Standpunkt aus werden vom Instrument Horizontalrichtung, Vertikalwinkel und Schrägstrecke zum Zielpunkt gemessen. Aus diesen können anschließend Polarkoordinaten berechnet werden.

Die Standardabweichung der Punktkoordinaten ist von der Genauigkeit der Strecken- und Richtungsmessung sowie von meteorologischen Einflussgrößen (Temperatur, Luftdruck und Luftfeuchte) abhängig. Auch die Art der Zielmarkierung, ob Prisma, Reflexfolie oder reflektorlos, beeinflusst die Genauigkeit. Präzisionstachymeter für Monitoring-Aufgaben erreichen



Abbildung 3.2.: Der Low-Cost-Einfrequenz-Empfänger „Piksi“ soll durch Echtzeitkinematik eine Lagegenauigkeit von bis zu 1 cm erreichen [SWIFT NAVIGATION INC., 2013]

eine Streckenmessgenauigkeit von $\sigma_s = 0,6 \text{ mm} + 1 \text{ ppm}$ und eine Richtungsmessgenauigkeit von $\sigma_r = 0,5''$ (0,15 mgon).

Für den Einsatz im Rahmen von Überwachungsmessungen eignen sich nur motorisierte Tachymeter. Die mit Servomotoren ausgestatteten Instrumente können Punkte automatisch anzielen und ggf. auch verfolgen. Spezielle Instrumente für den Dauereinsatz sind besonders verschleißarm und mit einem Schutz gegen Wind, Staub und Regen ausgestattet. Im Tachymeter integrierte Programme oder externe Steuerungsapplikationen erlauben eine kontinuierliche Messung vordefinierter Punkte. Für diesen Zweck bieten die Hersteller abgestimmte Hard- und Software-Systeme an.

3.2. Geotechnische Sensoren

Bei Überwachungsmessungen werden je nach Messgröße und erforderlicher Genauigkeit neben den geodätischen Sensoren auch eine Vielzahl geotechnischer Sensoren eingesetzt. Geotechnische Messverfahren umfassen die direkte Feststellung von Verdrehungen, Verformungen, Dehnungen, Kräften, Spannungen, Wasserdrücken, Temperatur, Beschleunigung, Schwingungen uvm. [MARTE U. A., 2012] Sie finden auch in anderen Ingenieurdisziplinen Anwendung.

3.2.1. Beschleunigungssensoren

Beschleunigungssensoren sind Aufnehmer für Beschleunigungen und Neigungen. Sie liefern meist Signale für verschiedene Richtungsänderungen. Im Bauwesen werden Beschleunigungssensoren bei der Überwachung von Brücken, Staudämmen und Gebäuden eingesetzt, um die

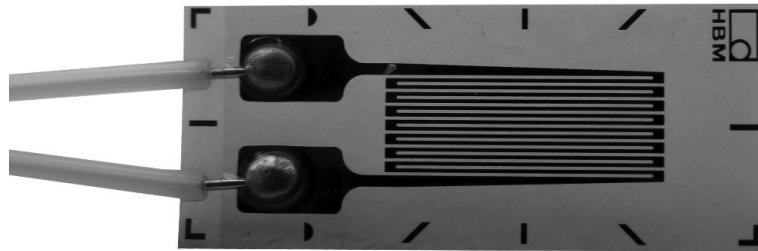


Abbildung 3.3.: Linearer Dehnungsmesstreifen mit einem Messgitter aus Konstantan

Schwinggeschwindigkeit zu bestimmen. Für die Bestimmung von Beschleunigungen steht eine Reihe von Messprinzipien zur Verfügung. Die am häufigsten verwendeten Sensoren messen die Änderung von Kapazitäten, Induktivitäten und Widerständen. [PAUL, 1998]

3.2.2. Dehnungsmesstreifen

Zur Messung von Dehnungen haben sich *Dehnungsmesstreifen* (DMS) durchgesetzt. Es handelt es sich dabei um passive Sensoren, die auf der Messung von Widerstandsänderungen von Metallen basieren. Treten an einem Messobjekt infolge äußerer oder innerer Krafteinwirkungen Verformungen auf, führen diese an der Oberfläche zu Dehnungen. Wird an der Oberfläche ein DMS befestigt, folgt dieser der Dehnung und verändert seinen elektrischen Widerstand. Die Änderung des Widerstands ist ein Maß für die aufgetretene Dehnung. Mit nachgeschalteten Geräten lässt sich diese messen und registrieren. [HOTTINGER U. KEIL, 1994]

Dehnungsmesstreifen lassen sich in verschiedene Kategorien einteilen. Sie unterscheiden sich in Materialien und Bauform:

Folien-DMS: Das messende Element dieser DMS ist ein mäanderförmiges Messgitter aus einer metallischen Widerstandslegierung. Es wird aus einer Metallfolie geätzt, anschließend auf eine Trägerfolie aufgebracht und mit einer Schutzfolie abgedeckt (Abb. 3.3). Folien-DMS werden am häufigsten verwendet. [SCHLEMMER, 1996]

Dünnfilm-DMS: Bei dieser Art werden die Legierungsbestandteile des messenden Elements unter Vakuum verdampft und damit direkt auf das Messobjekt aufgebracht. Die Technik ist nur in wenigen Bereichen anwendbar. Das Messprinzip entspricht dem der Folien-DMS.

Dickfilm-DMS: Bei den Dickfilm-DMS werden die einzelnen Schichten im Siebdruckverfahren auf einen keramischen Druckaufnehmer in Pastenform aufgebracht.

Halbleiter-DMS: Das Messprinzip der Halbleiter-DMS basiert auf dem piezoresistiven Effekt von Halbleitermaterialien. Sie sind empfindlicher als metallische DMS, aber auch temperaturabhängiger. [KOLLAR, 2000]

3.2.3. Extensometer

Mit *Extensometern* werden Relativverschiebungen und Dehnungen von Oberflächen über Distanzen von einigen Zentimetern bis wenigen Metern erfasst. Sie sind für festes Gestein und Mauerwerk ausgelegt und kommen bei der Überwachung von Bauwerken und Felswänden zum Einsatz. Es wird zwischen Draht-, Stangen- und Sondenextensometern unterschieden, die unterschiedliche Messmittel verwenden. Die Messung erfolgt entweder mit einer mechanischen Messuhr oder mit einem elektrischen Wegaufnehmer oder Linearpotentiometer. Letztere können auch an ein Überwachungssystem angeschlossen und automatisiert ausgelesen werden. [MARTE U. A., 2012]

3.2.4. Faseroptische Sensoren

Mit konventionellen geodätischen und geotechnischen Messverfahren kann nur die Oberfläche von Objekten überwacht werden. Um kontinuierlich Informationen aus dem Inneren von Objekten gewinnen zu können, wird u. a. auf *faseroptische Sensoren* (FOS) zurückgegriffen. Diese Form eingebetteter Sensoren basiert auf optischen Messverfahren und wird direkt in Bauwerke integriert. Grundlage sind Lichtwellenleiter aus Quarzglas und optische Polymerfasern aus Kunststoff. Die Wellenleiter reagieren auf Biegung, Zug, Druck und Torsion und erlauben dadurch Rückschlüsse auf geometrische Veränderungen. Dazu wird das eingesetzte Lichtsignal auf Laufzeit, Intensität, Wellenlänge und Polarisation untersucht, um Änderungen von Temperatur, Druck oder Länge des Objektes zu erfassen.

Faseroptische Sensoren sind immun gegenüber elektromagnetischen Feldern und kommen daher ohne Abschirmung aus. Sie können auch in blitz- und explosionsgefährdeten Umgebung eingesetzt werden, weil für ihren Betrieb keine elektrische Energie erforderlich ist. Sie sind widerstandsfähig gegenüber Erschütterungen und extremen Umwelteinflüssen, haben nur geringe Abmessungen ($d < 0,5$ mm) und ein geringes Gewicht und können große Entfernung zwischen Messort und Auswertetechnik überbrücken. Die opto-elektrische Signalwandlung ist aber aufwendig und in vielen Fällen müssen Temperatureinflüsse kompensiert werden. Je nach physikalischem Messprinzip sind anderen optische Ausgangssignale erforderlich. [SCHWARZ, 2013]

3.2.5. Inklinometer

Inklinometer sind Sensoren zur Feststellung von Neigungen. Sie können für die Überwachung einzelner Punkte oder für Messungen entlang einer Linie eingesetzt werden. Die Messung einzelner Punkte dient der Feststellung von Schiefstellungen oder Rotationen von Bauwerken. Messungen entlang einer Linie finden bei der Überprüfung von Achsen und Bohrlöchern Anwendung. Das Messprinzip basiert auf Flüssigkeits- oder Pendelsystemen. Das Flüssigkeitsniveau richtet sich immer rechtwinklig zur Lotrichtung aus und reflektiert auf treffende Lichtstrahlen. Bei einer Schiefstellung lässt sich so der Winkel des reflektierten Strahls messen. Mechanische Pendel erfassen die Neigung hingegen über Bezugspunkte. [DEUMLICH U. STAIGER, 2002] Kontinuierliche und automatisierte Messungen können mit festinstallierten Sensoren durchgeführt werden, während mobile Geräte flexibler, aber nicht ohne Weiteres für den dauerhaften Betrieb ausgelegt sind (Abbildung 3.4).



Abbildung 3.4.: Bluetooth-fähiger Low-Cost-Zweiachsneigungssensor DXL-360S mit einer Genauigkeit von $\pm(0,08^\circ + 1\%)$, bei einer Auflösung von $0,01^\circ$

3.2.6. Schlauchwaagen

Schlauchwaagen kommen beim hydrostatischen Nivellement zum Einsatz, das auf dem Prinzip der kommunizierenden Röhren beruht und bei dem der Flüssigkeitsspiegel einer ruhenden Flüssigkeit eine Horizontalebene bildet. Der relative Höhenunterschied zwischen zwei Punkten kann damit auch über große Entfernungen bestimmt werden. [DEUMLICH U. STAIGER, 2002] Elektronische Druckschlauchwaagensysteme erfassen Höhenänderungen an Bauwerken kontinuierlich. Diese Systeme werden häufig bei der Überwachung von Industriebauwerken, Staudämmen oder Turbinen eingesetzt. Die erreichbare Messgenauigkeit beträgt $\pm 0,01 \text{ mm}$.

Bei Deformationsmessungen mit Schlauchwaagensystemen werden diese in einer Ebene am zu überwachenden Objekt angebracht. Ein Luftschauch verbindet diese mit einer Schlauchwaage an einem Referenzmesspunkt. Ein zweiter Schlauch mit Messflüssigkeit verbindet Schlauchwaagen, Nachfüll- und Auffangbehälter miteinander. Der zeitliche Verlauf der Deformation des Messobjektes lässt sich dann relativ zum Referenzmesspunkt aufzeichnen.

3.3. Meteorologische Sensoren

Meteorologische Sensoren messen physikalische Zustände in der Erdatmosphäre, insbesondere Wetterereignisse. Zu den erfassten Größen gehören:

- Lufttemperatur
- Luftfeuchtigkeit
- Luftdruck
- Windstärke und -richtung
- Niederschlagsmenge
- Sonnenstrahlung

Mit Multisensor-Systemen lassen sich mehrere Messgrößen gleichzeitig bestimmen. Meteorologische Sensoren werden bei Überwachungsmessungen in Kombination mit geodätischen und geotechnischen Beobachtungen eingesetzt, um weitere Informationen über Ursachen und Wirkungen von geometrischen Veränderungen zu sammeln. Der Übergang zu den geotechnischen Sensoren ist oftmals fließend.

4. Systeme für geodätische Überwachungsmessungen

Geodätische Monitoring-Systeme dienen der Beobachtung, Erfassung und Auswertung von geometrischen Objektveränderungen unter Einsatz geodätischer und geotechnischer Messverfahren. Die verfügbaren Systeme ähneln sich in Struktur und Aufbau, wobei die individuelle Zusammensetzung durch die Komplexität der Überwachungsmessung definiert wird: Ein Monitoring-System zur Erfassung von jährlichen Bauwerkssetzungen unterscheidet sich maßgeblich von einem Tsunami-Frühwarnsystem. [STEMPFHUBER, 2009] Im Zentrum stehen aber bei allen Umsetzungen Mess- und Auswerteaufgaben.

Erste Steuerprogramme wurden schon frühzeitig auf Servo-Tachymetern installiert. Sie erlauben eine kontinuierliche Messung festgelegter Punkte und eine Speicherung der Daten auf dem Instrument oder einer Speicherkarte. Die Durchführung einer solchen Überwachungsmessungen ist nicht autonom, weil der Benutzer die Daten manuell auslesen und auswerten muss. Eine Benachrichtigung im Fehlerfall ist ebenfalls nicht möglich. In den letzten Jahren wurden Tachymeter daher mit Schnittstellen ausgerüstet (RS-232, USB, Bluetooth, WLAN), die eine Kommunikation mit externen Hard- und Software-Lösungen ermöglichen.

Ein autonomer Betrieb ist nur möglich, wenn der Benutzer jederzeit auf die Messdaten zugreifen und das Monitoring-System fernsteuern kann. Die dazu nötige Internetanbindung ist heute flächendeckend verfügbar. Im einfachsten Fall können die erfassten Messdaten per E-Mail verschickt oder per FTP auf einen zentralen Server kopiert werden. Viele Systeme besitzen bereits eine Weboberfläche, um das Monitoring-System über den Browser zu verwalten und Messdaten abzurufen. Alarmfunktionen zur Benachrichtigung des Benutzers ist ebenfalls verbreitet. Die Sensorsteuerung übernehmen dabei konventionelle Desktop-Rechner, Industriecomputer oder Individuallösungen auf Basis integrierter Schaltkreise.

4.1. Datenbank-orientiertes Monitoring- und Analyse-System

Das *Datenbank-orientierte Monitoring- und Analyse-System* (DABAMOS) wird seit 2009 an der Hochschule Neubrandenburg entwickelt. Das System erlaubt Überwachungsmessungen mit geodätischen und geotechnischen Sensoren unabhängig von ihrem Hersteller, darunter Tachymeter, Neigungssensoren und Digitalnivelliere. Zusätzlich dazu können weitere Größen, wie Temperatur, Luftfeuchte oder Wasserstand, überwacht werden, um Zusammenhänge und Auswirkungen der Umwelteinflüsse zu ermitteln.

Neben der reinen Messdatenerfassung und -speicherung gehören Auswerte- und Alarmierungsfunktionen zum Funktionsumfang. Diese ermöglichen eine Untersuchung des Datenbestandes und Benachrichtigungen bei Gefahren. Die Interaktion mit dem System erfolgt

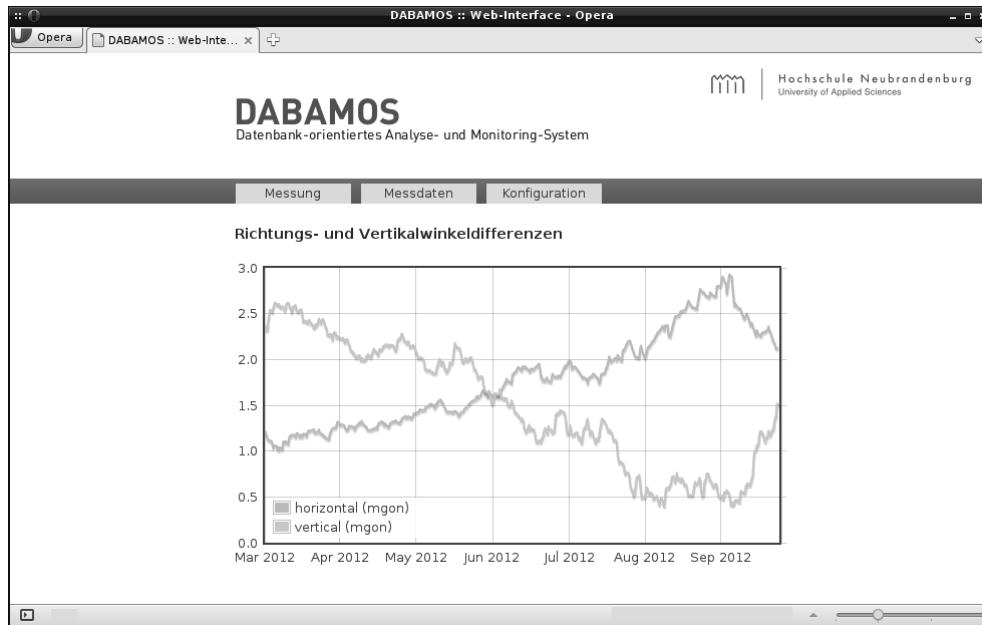


Abbildung 4.1.: Bildschirmfoto der Weboberfläche von DABAMOS

vollständig webbasiert (Abbildung 4.1). DABAMOS besteht aus mehreren Komponenten, zu denen Client- und Server-Software sowie ein Steuercomputer gehören (Abbildung 4.3).

4.1.1. Software

Die DABAMOS-Software besteht aus zwei Komponenten: einer Client- und einer Server-Applikation. Der Client wird auf dem Steuercomputer betrieben und hat die Aufgabe, die von den Sensoren gelieferten Messdaten an den Server zu schicken. Die Anwendung ist in Java SE geschrieben und damit plattformunabhängig einsetzbar. Die Server-Komponente ist hingegen eine Enterprise-Applikation auf Basis von Java. Sie speichert die von mehreren Clients gesendeten Messdaten und wertet sie aus. Die verwendete Datenbank ist die objektorientierte *db4o*. Zu den bereits implementierten Analysefunktionen gehören:

- Korrelation
- Fast Fourier Transformation
- Filter (Gauß, Moving Average)
- Regression (Grad 1 – 10)
- Transformation (Helmert)
- Kollokation
- Normierung
- Mittelwert

Die vom Server gespeicherten Messdaten können über eine Webschnittstelle direkt im Browser betrachtet und visualisiert werden. Bei Überschreitung von Grenzwerten ist darüber hinaus eine Alarmierung per E-Mail, SMS und Kurznachrichtendienste möglich.



Abbildung 4.2.: Testmessungen mit DABAMOS an einem Brückenmodell (Bild: THOMAS MANDT, HOCHSCHULE NEUBRANDENBURG)

4.1.2. Hardware

Während der Entwicklung und Erprobung von DABAMOS kamen bisher handelsübliche Personal Computer mit einem Linux-Betriebssystem zum Einsatz (Abbildung 4.2). Diese stellen für den autonomen Messbetrieb keine geeignete Grundlage dar. Ein platzsparende und wetterfeste Installation ist aufgrund der Größe des Gehäuses und der aktiven Kühlung nur schwer zu realisieren. Eine unterbrechungsfreie Stromversorgung zum Schutz vor Netzausfällen muss darüber hinaus an die hohe Leistungsaufnahme des Rechners angepasst sein.

4.2. Weitere Monitoring-Systeme

Der zunehmende Bedarf an geodätischen und geotechnischen Überwachungsmessungen hat dazu geführt, dass sich einige Hersteller auf entsprechende Soft- und Hardwarelösungen spezialisierten. Insbesondere die großen Hersteller geodätischer Sensoren – Leica Geosystems, Trimble Navigation und Sokkia – bieten auf ihre Geräte abgestimmte Monitoring-Systeme an. Viele kleinere Anbieter sind nur lokal im Ausland vertreten oder konzentrieren sich auf ausgesuchte Anwendungsfälle (Bergbau, Tunnelvortriebe, Hydrographie).

Auf dem Markt sind eine Vielzahl verschiedener Monitoring-Lösungen verfügbar, zu der die folgende Auswahl gehört:

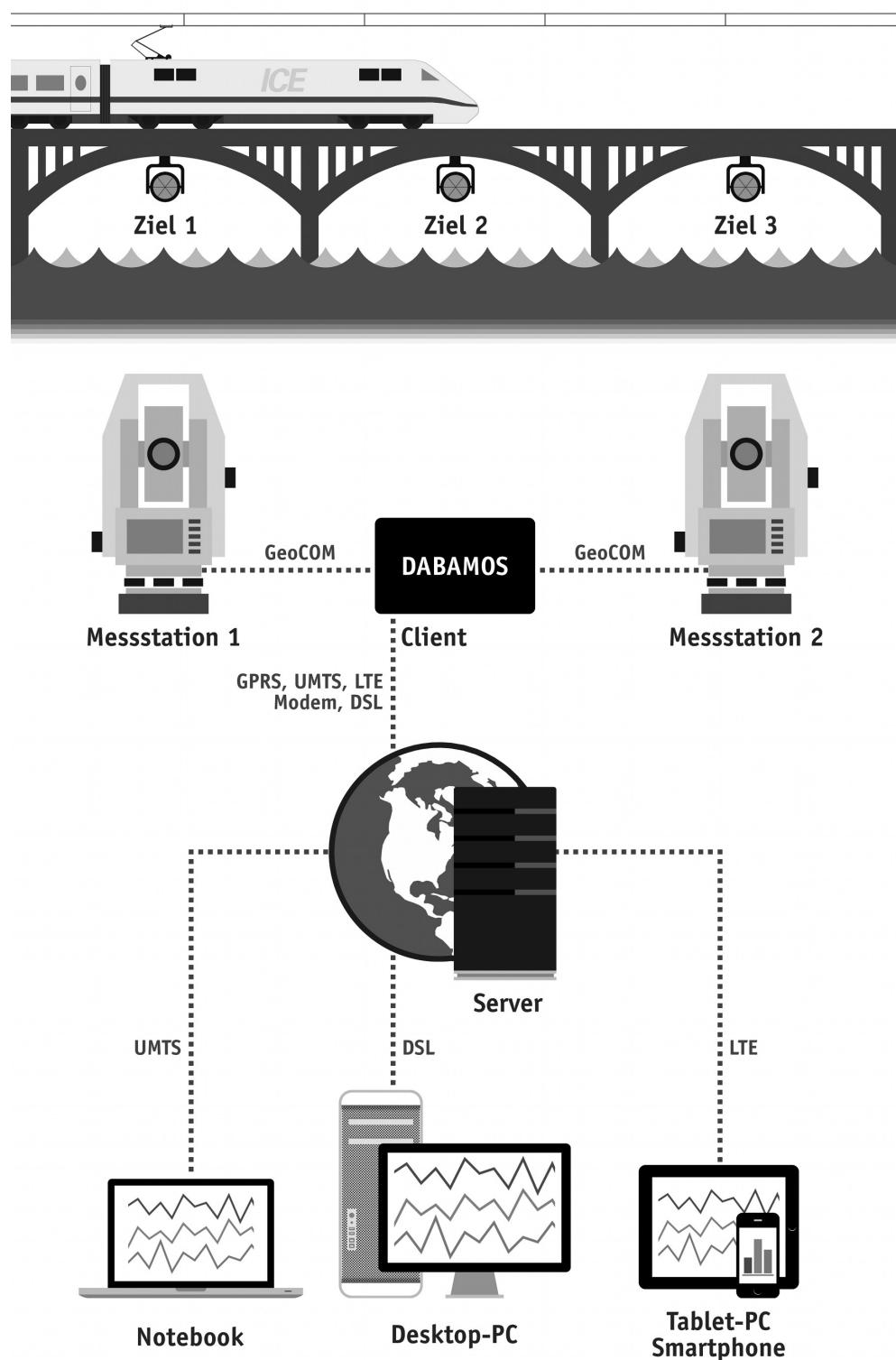


Abbildung 4.3.: Schematischer Aufbau eines Monitorings mit DABAMOS bei Nutzung von zwei Tachymetern

- Amberg *GeoMonitoring*
- BSF Swissphoto *DeTras*
- Geodata *Eupalinos*
- Hochschule Karlsruhe *GOCA*
- Leica *GeoMoS*
- Pöyry *2Doc*
- Sokkia/Topcon *RAPID ADMS*
- Solexperts *GeoMonitor*
- TU München *MoSTUM*
- Terra Monitoring *swissMon*
- Trigonos *Loxia*
- Trimble *4D Control*

Von diesen Systemen sollen GOCA, Leica GeoMoS, MoSTUM und Trimble 4D Control kurz beschrieben werden.

4.2.1. GNSS/LPS/LS-based Online Control and Alarm System

Das *GNSS/LPS/LS-based Online Control and Alarm System* (GOCA) ist ein Forschungs- und Entwicklungsprojekt des Instituts für Angewandte Forschung der Hochschule Karlsruhe. Neben globalen Satellitennavigationssystemen können auch lokale Positionierungssysteme und Sensoren für Echtzeit- und Near-Online-Deformationsanalysen genutzt werden. Das System arbeitet dabei herstellerunabhängig und bietet eine zentrale Schnittstelle zur Anbindung von Sensoren und Software-Produkten, wie Leica GeoMoS oder Leica GNSS Spider.

Die Funktionen von GOCA umfassen eine Online-Modellierung klassischer Deformationsnetze, Aufzeichnung, Visualisierung und Analyse der Zeitreihendaten auf einem zentralen Rechner vor Ort und die automatisierte Alarmierung beim Eintreten kritischer Zustände am Objekt. Der Steuerungsrechner lässt sich über ein Fernwartungssystem verwalten. [GOCA, 2013]

4.2.2. Leica GeoMoS

Das Unternehmen Leica Geosystems bietet mit *GeoMoS* eine verbreitete Lösung für Überwachungsmessungen an. Das System besteht aus den Software-Komponenten *Monitor*, *Analyzer* und *Adjustment*. Die onlinefähige Anwendung *Monitor* dient der Sensorsteuerung, Datensammlung, Toleranzkontrolle und Ereignis-Verwaltung (Abbildung 4.4). *Analyzer* ist eine reine Offline-Anwendung zur Analyse und Visualisierung der Messdaten und *Adjustment* wird u. a. für Netzausgleichungen und Deformationsanalysen eingesetzt.

GeoMoS konzentriert sich auf herstellereigene Sensoren, darunter Tachymeter, Nivelliere und Inklinometer. Die GNSS-Empfänger werden über das externe System Leica GNSS Spider eingebunden. Die Anzahl der nutzbaren Sensoren ist abhängig von den erworbenen Lizenzen. Die Sensoransteuerung und Datenerfassung unterstützt kabelgebundene und kabellose Übertragungsstandards. Bei der Überschreitung von Grenzwerten oder vordefinierten Ereignissen ist eine Benachrichtigung des Benutzers über E-Mail, SMS, Pager und andere externe Geräte möglich. Die erfassten Messdaten werden in einer SQL-Datenbank gespeichert.

Mit Leica GeoMoS Web wird vom Hersteller ein zusätzlicher Host-Service angeboten. Der Benutzer hat mit einem Webbrowser Zugriff auf die Messdaten und kann sie online visualisieren und analysieren.

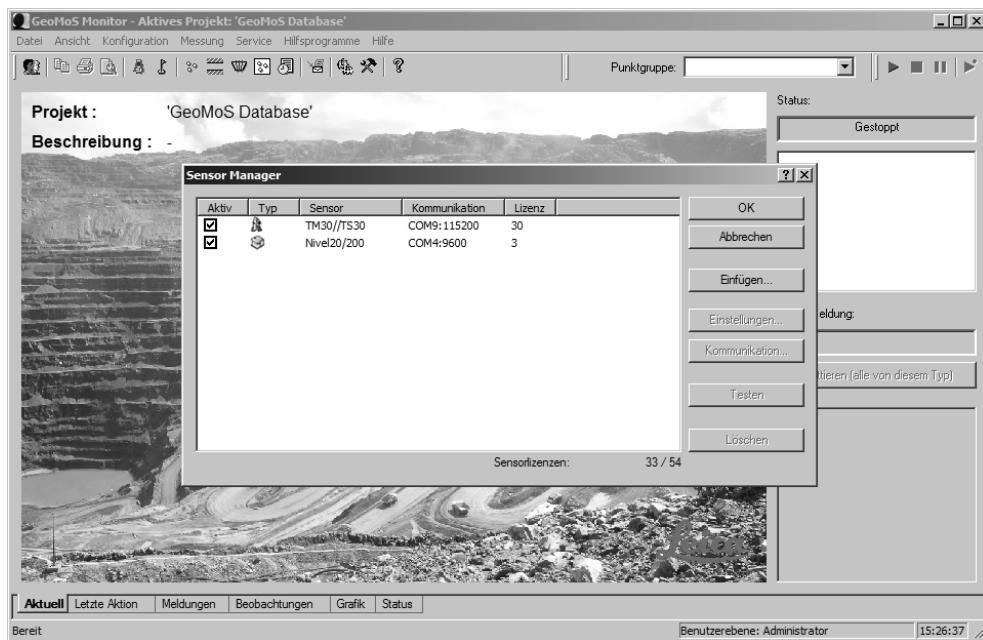


Abbildung 4.4.: Bildschirmfoto des Monitoring-Systems Leica GeoMoS

4.2.3. Monitoring System der Technischen Universität München

An der TU München wird für geodätischen Überwachungsmessungen im Innenbereich das *Monitoring System der Technischen Universität München*, kurz MoSTUM, entwickelt. Einsatzfeld ist insbesondere die Überwachung von Sakralbauten, wie Kirchen oder Klöster. Das System erlaubt die Integration beliebiger Sensoren, darunter Tachymeter von Leica und Trimble, Inklinometer, elektronische Distanzmesser und meteorologische Sensoren. Über weitere Programme lassen sich die erfassten Messdaten an einen zentralen Rechner senden, auswerten und darstellen. [REITH U. WASMEIER, 2009]

4.2.4. Trimble 4D Control

Der amerikanische Hersteller von Vermessungs- und Navigationsinstrumenten Trimble bietet mit *4D Control* eine eigene Monitoring-Lösung an, die nur mit herstellereigenen geodätischen Sensoren zusammenarbeitet. Unterstützt werden die Tachymeter Trimble S6 und Trimble S8 sowie GNSS-Empfänger, wie der NetR9. Eine Reihe von externen geotechnischen und meteorologischen Sensoren ist ebenfalls einsetzbar.

Das System besteht aus mehreren Hard- und Software-Komponenten, die teilweise auf den Instrumenten selbst betrieben werden. *Trimble Access* dient dabei auf Tachymetern der Reihen S6 und S8 als Konfigurations-, Ansteuerungs- und Speicher-Software. Der Datenaustausch mit anderen Computern erfolgt mit dem *AccessSync*-Programm. Die Speicherung und Verarbeitung von Messdaten wird auf einem lokalen Rechner-Verbund durchgeführt. Der Fernzugriff über das Internet ist durch eine Remote-Verbindung zum darunterliegenden Windows-Betriebssystem möglich.

5. Autonomer Low-Cost-Messclient

Elektronische Steuersysteme für geodätische und geotechnische Sensoren sind bereits seit längerer Zeit verfügbar. Insbesondere die Sensorhersteller bieten auf ihre Produkte abgestimmte Hard- und Software-Lösungen an, um automatisiert Messdaten zu erfassen. Bei der Hardware handelt es sich entweder um konventionelle Computertechnik oder um speziell gefertigte integrierte Schaltkreise, die jeweils als Messcomputer dienen.

Die notwendigen finanziellen Aufwendungen für Computertechnik sinken stetig, weil immer neuere und leistungsfähigere Hardware auf den Markt drängt. Personal Computer sind aber für den Einsatz im Umfeld von Überwachungsmessungen häufig ungeeignet, weil sie den Anforderungen einer rauen Umgebung nicht gewachsen sind. Aufgrund ihrer hohen Leistungsaufnahme ist eine Absicherung gegen Stromausfall aufwendig und teuer. Auch der Schutz gegen das Eindringen von Wasser und Fremdkörpern ist nur schwer zu gewährleisten, weil die aktive Kühlung der Systemkomponenten eine vollständige Abdichtung verhindert. Die verbaute Hardware ist unter Umständen nicht für den Dauereinsatz ausgelegt.

Dagegen sind speziell entwickelte integrierte Schaltkreise den hohen Beanspruchungen zwar gewachsen, sie benötigen aber abgestimmte Software, bearbeiten nur einen fest umrissenen Aufgabenbereich und sind nicht und nur im geringen Maße erweiterbar. Die geringe Rechenleistung beschränkt darüber hinaus die Funktionen auf das reine Erfassen und Weiterleiten von Messdaten.

Bei den verfügbaren Systemen handelt es sich oft um historisch gewachsene Lösungen. Viele der heutigen technischen Möglichkeiten standen zum damaligen Entwicklungszeitpunkt noch nicht zur Verfügung. Ein vollständige autonomer Betrieb ist bei vielen Überwachungssystemen nicht vorgesehen. Die eingesetzte Hard- und Software erfordert eine regelmäßige Interaktion mit dem Benutzer. Dieser muss Daten manuell sichern, das System auf Fehler prüfen und eine kontinuierliche Stromversorgung gewährleisten.

Der zu entwickelnde Low-Cost-Messclient soll diese Nachteile kompensieren und eine herstellerunabhängige Lösung darstellen, um Sensoren zu steuern, Messdaten zu speichern und diese auf entfernte Rechnersysteme zu übertragen. Als Steuersoftware kommt das Datenbankorientierte Monitoring- und Analyse-System zum Einsatz. Der Messclient soll aber nicht auf die Nutzung dieser Anwendung beschränkt sein, sondern auch eine universelle Grundlage für andere Systeme sein.

5.1. Anforderungen

An den zu entwickelnden Messclient werden eine Reihe von Anforderungen hinsichtlich Aufbau, äußeren Einflüssen, technischen Spezifikationen und Kosten gestellt. Das System muss

für den Außeneinsatz unter widrigen Bedingungen geeignet sein, genügend Rechenleistung für die gestellten Aufgaben aufbringen und gleichzeitig nur niedrige Produktionskosten aufweisen. Die einzelnen Anforderungen werden im Folgenden weiter ausgeführt.

5.1.1. Mechanik und Widerstandsfähigkeit

Der Messclient soll sich für den Einsatz im Außenbereich und in der rauen Umgebung einer Baustelle eignen. Daher darf er keine mechanisch-beweglichen Komponenten besitzen, da diese zu vorzeitigen Ausfällen führen und die Lebensdauer des Systems verkürzen würden. Die Kühlung des Systems muss passiv und ohne Einsatz von Lüftern erfolgen. Dies gilt insbesondere für den Hauptprozessor und das Netzteil. Das Gehäuse kann nur gegen Wasser und Staub abgedichtet werden, wenn keine Lufteinlassöffnungen nötig sind (vgl. auch Abschnitt 6.4). Der Verzicht auf eine aktive Kühlung führt darüber hinaus zu einem niedrigen Geräuschpegel. Je nach Einsatzort des Messclients kann diese Eigenschaft gefordert sein, so z. B. bei der Überwachung von Sakralbauten.

Der Messclient muss nicht nur Staub und Feuchtigkeit widerstehen, sondern auch Erschütterungen, hohe und niedrige Temperaturen sowie elektromagnetischer Strahlung. Der Temperaturarbeitsbereich des gesamten Systems ist durch die Komponente mit den restriktivsten Werten vorgegeben. Soll der Messclient zu jeder Jahreszeit auch im Außenbereich zum Einsatz kommen, ist ggf. eine zusätzliche Kühlung oder Heizung nötig. Zwischen der Widerstandsfähigkeit gegenüber sehr hohen und sehr niedrigen Temperaturen und dem damit verbundenen Aufwand zur Absicherung des Betriebs muss ein Mittelweg gefunden werden – insbesondere in Hinblick auf die geforderte Kostenbeschränkung. Die Abschirmung vor elektromagnetischer Strahlung erfolgt durch ein Gehäuse aus Metall oder durch aufgeklebte Metallfolien im Inneren des Gehäuses. Weiterhin müssen auch Kabel, die zum Messclient führen, geschirmt sein, da sie sonst als Antennen arbeiten.

5.1.2. Leistung

Die Rechenleistung moderner Computersysteme übersteigt in den meisten Fällen die Anforderung, die die Sensoransteuerung stellt. Der autonome Low-Cost-Messclient muss aber auf leistungsschwacher Hardware basieren, um den Ansprüchen an Kosten, Größe, Leistungsaufnahme und Stabilität gerecht zu werden.

Der rechnerische Aufwand bei der Ansteuerung, Verarbeitung und Speicherung von Messdaten ist nur gering. Bereits einfache Mikrocontroller können für diese Aufgaben eingesetzt werden. Sollen aber weitere Dienste Zusatzfunktionen anbieten, darunter eine Weboberfläche zur lokalen Administration, die Ansteuerung von GNSS-Receivern und Kamerasystemen oder aufwendige statistische Berechnungen, ist eine höhere Rechengeschwindigkeit notwendig, als sie Mikrocontroller bereitstellen. Abschnitt 5.2 soll diesen Aspekt und die verschiedenen Computersysteme näher beleuchten.

5.1.3. Speichermedien

Der Messclient ist auf ein geeignetes Speichermedium angewiesen. Zum einen müssen Betriebssystem und Messclient-Software auf einem solchen installiert werden, zum anderen erfordern auch die erfassten Messwerte einen Speicherort. Eine Möglichkeit sind handelsübliche Festplatte, die auch in kleinen Baugrößen von 2,5 " und 1,8 " erhältlich sind. Die Speicherkapazität in diesen Baugrößen liegt in der Regel zwischen 60 GB und 2 TB. Festplatten unterliegen einem Verschleiß, der von vielen Faktoren abhängig ist, darunter Laufzeit, Temperatur und mechanischer Beanspruchung.

Ein weiteres Medium sind Flash-Speicher. Diese haben keine beweglichen Teile, sind gegenüber mechanischer Einflüssen unempfindlich und haben einen größeren Temperaturarbeitsbereich als Festplatten. Flash-Speicher sind in verschiedenen Varianten verfügbar. *Solid-State Drives* sollen herkömmliche Festplatten ersetzen und sind für den Einsatz in Desktop- und Server-Computern ausgelegt. Sie werden über eine Serial-ATA-Schnittstelle angeschlossen. SD- und CF-Speicherkarten kommen ebenfalls zum Einsatz. Sie benötigen aber passende Controller, die direkt auf der Platine verbaut sind oder als externe Lesegeräte per USB angebunden werden. Die Anzahl der Schreibzugriffe auf Flash-Speichermedien ist beschränkt. Mit Techniken wie *Block Journaling* und *Wear Leveling* kann die Lebensdauer zwar verlängert werden, sie liegt aber immer noch unter der moderner Festplatten. Je nach eingesetzten Speicherbausteinen werden von den Herstellern 10.000 bis 100.000 Schreibzyklen angegeben. [SCHRÖDER U. A., 2009] Flash-Speicher der *Industrial-Grade*-Klasse weisen darüber hinaus auch eine garantierter Mindestlebensdauer auf.

5.1.4. Echtzeitfähigkeit

Bei der Interaktion mit Sensoren müssen Zeitbedingungen beachtet und eingehalten werden. Bei Überwachungsmessungen wird immer auch ein Zustand zu einem bestimmten Zeitpunkt festgehalten. Der Messclient muss sich dieser Anforderung anpassen und von Sensoren vorgegebene Messfrequenzen übernehmen. Diese Eigenschaft wird durch den Begriff „Echtzeit“ oder „Realzeit“ beschrieben. Die DIN 44300 (Teil 9) beschreibt die „Realzeitverarbeitung“ wie folgt:

Eine Verarbeitungsart, bei der Programme zur Verarbeitung anfallender Daten ständig ablaufbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu bestimmten Zeitpunkten anfallen.

Nach [BÄHRING, 2010] lassen sich Echtzeitsysteme abhängig von der Zeitbedingung in drei Klassen einteilen:

Harte Echtzeitsysteme erfordern die unbedingte Einhaltung von Zeitbedingungen. Sollte die Zeitbedingung nicht erfüllt werden, können die Folgen sehr schwerwiegend sein, z. B. bei der Kollisionserkennung von Fahrzeugen.

Feste Echtzeitsysteme zeichnen sich dadurch aus, dass die Überschreitung fester Zeitgrenzen das Ergebnis der Berechnung wertlos macht. Als Beispiel sei die Positionserkennung von Fahrzeugen genannt. Eine verspätete Berechnung ist unbrauchbar, weil sich die Position des Fahrzeugs zwischenzeitlich geändert hat.

Weiche Echtzeitsysteme sind durch Zeitschränken gekennzeichnet, die nur eine grobe Richtlinie darstellen. Deren Überschreiten kann daher toleriert werden. Bei der regelmäßigen Erfassung von Temperaturwerten durch einen Sensor ist eine geringe Abweichung vom Messzeitpunkt hinnehmbar, weil der Temperaturunterschied nicht signifikant ist.

Bei geodätischen Überwachungsmessungen werden überwiegend Langszeitbeobachtungen zum Verhalten von Bauwerken und topografischen Gebieten durchgeführt, um mögliche Bewegungen zu erfassen und Aussagen über zukünftige Entwicklungen treffen zu können. Bei baubegleitenden Messungen ist auch die frühzeitige Erkennung von Bewegungen gefordert, um ggf. einen Baustopp und eine Evakuierung der Baustelle im Gefahrenfall einzuleiten. Diese Anforderungen implizieren lediglich eine weiche Echtzeitfähigkeit, bei der Überschreitungen hinnehmbar sind, weil die Messfrequenz in den meisten Fällen im Sekunden- bis Stundenbereich liegt. Ob höhere Frequenzen von 50 Hz und mehr, wie sie z. B. bei Schwingungsmessungen anfallen, erfasst werden können, ist aber gesondert zu prüfen.

5.1.5. Stromversorgung

Die Stromversorgung des Messclients ist möglichst einfach aufzubauen. Auf die Entwicklung eines eigenen Netzteils soll, auch in Hinblick auf eine CE-Zertifizierung, verzichtet und stattdessen auf im Handel erhältliche Lösungen zurückgegriffen werden. Der Messclient soll in jedem Fall durch eine unterbrechungsfreie Stromversorgung abgesichert werden, die nach Möglichkeit den Anschluss eines Solarmoduls erlaubt. Bei einem Netzausfall wird der Messclient dann mit der gespeicherten elektrischen Energie eines Akkumulators versorgt.

5.1.6. Kosten

Die Kosten, die für die einzelnen Bauteile und bei der Herstellung anfallen, sollen so gering wie möglich sein, damit das System dem Anspruch an eine Low-Cost-Lösung gerecht wird. Die Anforderungen an Stabilität, Robustheit und Witterungsbeständigkeit sind ungeachtet dessen zu erfüllen, so dass der Messclient für den Außen- und Baustellen-Einsatz geeignet ist. Dieses Ziel ist erreichbar, wenn auf standardisierte und bereits im Handel erhältliche Komponenten zurückgegriffen und eine Eigenentwicklung, insbesondere von elektrischen und elektronischen Schaltungen, vermieden wird. Die Kosten für den Steuerungscomputer, die unterbrechungsfreie Stromversorgung und das Gehäuse sollen nach ersten Überlegungen bei etwa 100 € liegen.

5.2. Hardware-Plattformen

Als Grundlage für einen autonom arbeitenden Client-Computer kommen verschiedene Hardware-Plattformen in Frage. Zu ihnen gehören Mikrocontroller, Industrie- und Einplatinen-Computer. Handelsübliche Desktop-Computer und Laptops bleiben wegen ihrer bereits ausführten Nachteile unberücksichtigt.

5.2.1. Industrie-Computer

Industrie-Computer werden im industriellen Bereich eingesetzt. Sie unterscheiden sich von herkömmlichen Desktop-Computern hinsichtlich des Formfaktors, der Zuverlässigkeit und der Widerstandsfähigkeit gegenüber möglichen Beanspruchungen. In der Automatisierungs-technik hat die Ausfallsicherheit einen hohen Stellenwert. Die Zuverlässigkeit von Industrie-Computern wird durch höherwertige Komponenten erreicht, die thermischer Beanspruchung oder mechanischer Belastung widerstehen. Die Lebensdauer soll jener industriellen Anlagen entsprechen, die durch die Industrie-PCs gesteuert werden. Die Gehäuse unterscheiden sich je nach Einsatzzweck und reichen von 19"-Baugruppenträgern bis zu Industrie-Bus-Systemen.

Kompaktere Ableger werden als *Embedded-PC* bezeichnet. Sie sind für einen bestimmten Einsatzzweck ausgelegt und haben nur unbedingt benötigte Komponenten integriert. Dadurch sind besonders kleine Bauformen realisierbar. Einige Systeme können auch auf Hutschienen in Schaltschränken montiert werden.

5.2.2. Mikrocontroller

Mikrocontroller sind Rechner, die alle Teile eines kleinen Computers in einem Chip vereinen, darunter Prozessorkern, Speicher und Eingabe-/Ausgabeschnittstellen. Sie weisen nur eine geringe Größe auf und eignen sich insbesondere für Steuerungs- und Kommunikations-aufgaben. Die Einsatzgebiete stellen sehr unterschiedliche Anforderungen an Leistung und Komplexität. Im Gegensatz zu universellen Prozessoren werden bei Mikrocontroller noch häufig 4-Bit, 8-Bit und 16-Bit-Prozessoren eingesetzt. Verbreitete Mikrocontroller-Familien sind Atmel AVR und PICmicro (Abbildung 5.1).

Zu den typischen Eigenschaften der Mikrocontroller gehören:

- fest installierte Programme für konkrete Anwendungen,
- nichtflüchtiger Programmspeicher (Flash-EEPROM) und
- geringer Stromverbrauch bei einer Betriebsspannung von 3 V und weniger.

Diese Eigenschaften führen zu einer kostengünstigen Herstellung von Bausteinen und Platinen sowie zu niedrigen Betriebskosten. Änderungen und Anpassungen in der Entwurfsphase lassen sich schnell in den Festspeicher schreiben und verkürzen so die Entwicklungszeit. Bei einer anschließenden Massenproduktion werden die Kosten durch die Programmierung zum Herstellungszeitpunkt weiter reduziert. [BÄHRING, 2010]

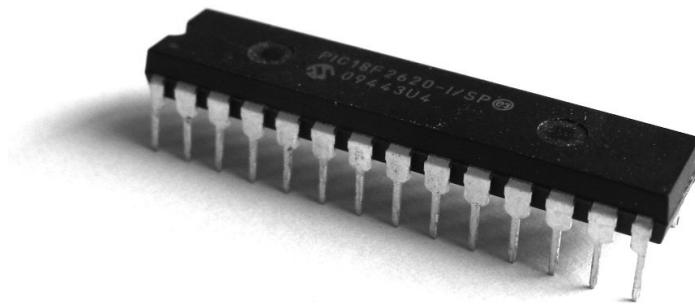


Abbildung 5.1.: Der PIC18F2620-I/SP ist ein einfacher 8-Bit-Mikrocontroller mit 40 MHz Systemtakt und 64 KB Programmspeicher.

Die verwendete Programmiersprache wird je nach Einsatzzweck des Mikrocontrollers gewählt und ist auch abhängig von der Architektur. In vielen Bereichen kommen Assembler oder C zum Einsatz. Andere Sprachen, darunter BASIC, Pascal und Ada, sind aber ebenfalls verbreitet.

5.2.3. Einplatinen-Computer

Ein *Einplatinen-Computer* ist in vollständiger Computer, der alle zum Betrieb nötigen Komponenten, darunter Prozessor, Speicher und Eingabe-/Ausgabe-Schnittstellen, auf einer Plakette vereint. Im Gegensatz zu Desktop-Computern fallen die Abmessungen der Einplatinen-Computer geringer aus. Sie eignen sich daher insbesondere für den Einsatz in eingebetteten Systemen.

Einplatinen-Computer setzen häufig nicht auf die im Desktop-Bereich verbreiteten *x86*- und *amd64*-Architekturen; stattdessen sind neben Mikrocontrollern vor allem ARM- und MIPS-basierte Prozessoren verbaut. In den letzten Jahren konnten sich *System on a Chip* (SoC) genannte integrierte Schaltkreise durchsetzen, die alle Komponenten eines Computers in einem einzigen Chip vereinen. Diese führten zu Kostensenkungen in der Produktion, die wiederum besonders kostengünstige Systeme möglich machten. Einige Einplatinen-Computer sind bereits für deutlich unter 50 € erhältlich.

6. Hardware

Die Hardware des Messclients besteht aus einzelnen Komponenten, die zu einem Gesamtsystem zusammengesetzt werden. Zu diesen Komponenten gehören

- Computer,
- unterbrechungsfreie Stromversorgung,
- Netzgerät und
- Gehäuse.

In diesem Kapitel sollen diese näher betrachtet und konkrete Produkte, die sich für den Einsatz im Messclient eignen, vorgestellt werden.

6.1. Einplatinen-Computer

Einplatinen-Computer eignen sich aufgrund ihrer hohen Leistung, geringen Abmessungen und niedrigen Kosten. Eine Reihe von Betriebssystemen, insbesondere aus dem Unix- und Linux-Umfeld, sind verfügbar. Anders als bei Mikrocontrollern stehen auch moderne Hochsprachen für die Programmierung bereit.

Auf dem Markt sind Einplatinen-Computer verschiedener Architekturen und Hersteller verfügbar. Am weitesten verbreitet sind ARM-basierte Systeme, wie folgende Liste zeigt:

- | | |
|--|---|
| <ul style="list-style-type: none"> • ARM <ul style="list-style-type: none"> – AllWinner A1x – ARMv6, ARMv7 – Freescale i.Mx – Qualcomm Snapdragon – Rabbit/Z80 – Samsung Exynos – ST-Ericsson NovaThor – Texas Instruments OMAP – Xilinx Zynq | <ul style="list-style-type: none"> • Atmel AVR <ul style="list-style-type: none"> – ATmega • MIPS • x86 <ul style="list-style-type: none"> – AMD Geode – Intel Atom – Vortex86 |
|--|---|

Eine umfangreiche und kontinuierlich erweiterte Liste von Einplatinen-Computern wird von der englischsprachigen Wikipedia geführt.¹

¹http://en.wikipedia.org/wiki/List_of_single_board_computers

Name	CPU	RAM	Flash	Schnittstellen*	Preis
Arduino Due	84 MHz	96 KB	512 KB	erweiterbar mit Shields	45,- €
Arduino Leonardo	16 MHz	2,5 KB	32 KB	erweiterbar mit Shields	20,- €
Arduino Uno	16 MHz	2 KB	32 KB	erweiterbar mit Shields	25,- €
Arduino Yún	16 MHz	32 KB	8 MB	erweiterbar mit Shields	20,- €
BeagleBoard†	720 MHz	256 MB	256 MB	RS-232, SD	125,- €
BeagleBoard-xM‡	1 GHz	512 MB	microSD	Ethernet, RS-232	150,- €
BeagleBone§	720 MHz	256 MB	microSD	Ethernet	80,- €
BeagleBone Black	1 GHz	512 MB	2 GB	Ethernet, microSD	50,- €
Carambola	350 MHz	32 MB	8 MB	Ethernet, RS-232, WLAN	40,- €
Carambola 2	400 MHz	64 MB	16 MB	Ethernet, RS-232, WLAN	40,- €
CubieBoard	1 GHz	1 GB	4 GB	Ethernet, microSD, SATA	65,- €
GNUBLIN-LAN	180 MHz	32 MB	microSD	Ethernet	55,- €
PandaBoard	1,2 GHz	1 GB	SD	Bluetooth, Ethernet, WLAN	150,- €
Raspberry Pi (A)	700 MHz	256 MB	SD	Ethernet	25,- €
Raspberry Pi (B)	700 MHz	512 MB	SD	Ethernet	35,- €

* bei allen auch GPIO, I²C, USB und SPI

† Revision D

‡ Revision C2

§ Revision A6A

|| Revision A5B

Tabelle 6.1.: Gegenüberstellung ausgesuchter Einplatinen-Computer mit ihren Einzelhandelspreisen

In diesem Abschnitt sollen Systeme vorgestellt und ihre technischen Spezifikationen näher ausgeführt werden, um eine Auswahl an geeigneten Einplatinen-Computern treffen zu können. Es werden dabei nur Platinen untersucht, die zum Zeitpunkt der Betrachtung bereits auf dem Markt verfügbar sind (Tabelle 6.1).

6.1.1. Arduino

Bei *Arduino* handelt es sich um eine sogenannte *Physical-Computing*-Plattform², die aus zwei Komponenten besteht. Zum einen die Arduino-Hardware, eine Mikrocontroller-Platine, die auf der 8-Bit-AVR-Familie von Atmel basiert. Zum anderen die Arduino IDE, ein plattformunabhängige integrierte Entwicklungsumgebung, die die Programmierung des Mikrocontrollers ermöglicht. Die Programmiersprache ist an Wiring angelehnt. Alternativ können Programme auch in reinem C geschrieben werden. Software und Hardware sind Open Source (LGPL/GPL), die technischen Spezifikationen und Schaltpläne sind frei verfügbar. Der

² *Physical Computing* ist eine Form des elektronischen *Rapid Prototyping*. Es umfasst den Entwurf interaktiver Systeme, die physikalische Objekte mit Hardware und Software verbinden. [ODENDAHL U. A., 2009; BANZI, 2011]

Quelltext der IDE ist ebenfalls offengelegt. Die Platine kann vorgefertigt gekauft oder aber von Hand aus den einzelnen Komponenten zusammengebaut werden. [BANZI, 2011]

Nachfolgend soll auf einige Arduino-Boards eingegangen werden. Unberücksichtigt bleiben sehr kleine Platinen, wie Fio, Micro oder Nano, und Platinen in ungewöhnlicher Bauform, wie Esplora, LilyPad oder Robot.

Arduino Due: Der erste Arduino, der auf einer 32-Bit-ARM-Architektur basiert. Der verbaute Mikrocontroller von Atmel basiert auf dem ARM Cortex-M3 und arbeitet mit 84 MHz. Der Due ist mit 54 digitalen I/O-Pins ausgestattet, von denen zwölf für Pulsweitenmodulation (PWM) nutzbar sind. Des Weiteren verfügt er über zwölf analoge Eingänge und vier serielle Schnittstellen. Für Programme stehen 512 KB Flash-Speicher und 96 KB Arbeitsspeicher zur Verfügung.

Arduino Leonardo: Das Leonardo-Board basiert auf einem ATmega32u4-Mikrocontroller. Die 16 MHz Taktgeschwindigkeit liefert ein Quarzoszillator. Als Speicher stehen 32 KB Flash, 2,5 KB SRAM und 1 KB EEPROM zur Verfügung. Die 20 digitalen Ein- und Ausgänge sind großzüig bemessen. Sieben sind für PWM einsetzbar und 12 dienen auch als analoge Eingänge. Zudem ist ein Micro-USB-Anschluss vorhanden.

Arduino Uno: Auf dem Mikrocontroller-Board ist ein ATmega328 von AVR verbaut, der mit einer Taktgeschwindigkeit von 16 MHz arbeitet. Der verfügbare Speicher beschränkt sich auf 32 KB Flash, 2 KB SRAM und 1 KB EEPROM. Von den 14 digitalen I/O-Pins sind sechs für PWM geeignet. Darüber hinaus stehen auch sechs analoge Eingänge zur Verfügung. Neben der USB-Schnittstelle ist auch ein 6-poliger ISP-Anschluss vorhanden.

Arduino Yún: Der Yún basiert auf dem Arduino Leonardo und ist sowohl mit einem ATmega32U4-Mikrocontroller mit 16 MHz als auch mit einem MIPS-basierten WLAN-SoC von Atheros (AR9331) ausgestattet, der mit OpenWrt betrieben wird. Als Speicher stehen für den Mikrocontroller 2,5 KB RAM und 32 KB Flash, für den WLAN-SoC 32 MB RAM und 8 MB Flash zur Verfügung. Yún ist das chinesische Wort für „Wolke“. Der Begriff bezieht sich auf das Cloud-Computing, für das der Einplatinen-Computer ausgelegt ist. Der Atheros AR9331 arbeitet mit 400 MHz und dient dem Aufbau von WLAN-Funkverbindungen. Power over Ethernet kann durch ein externes Modul nachgerüstet werden. Die Stromversorgung erfolgt dann durch das Ethernetkabel. Über 14 I/O-Pins lassen sich externe Geräte ansteuern. Sieben der Pins eignen sich für PWM, zwölf als analoge Eingänge. [ARDUINO BLOG, 2013]

Eine Besonderheit der Arduino-Plattform ist ihre einfache Erweiterung durch sogenannte *Shields*. Diese Erweiterungsplatinen werden einfach auf das Arduino-Board gesteckt und stellen anschließend weitere Schnittstellen oder Controller zur Verfügung. Es existieren Shields, die die Kommunikation über Funkstandards wie Bluetooth, GPRS, WLAN oder ZigBee erlauben (Abbildung 6.1). Mit anderen kann der Speicher durch SD-Karten erweitert oder eine Positionsbestimmung mittels Low-Cost-GPS-Modulen durchgeführt werden.



Abbildung 6.1.: Ein *GSM Shield* mit eingesetzter SIM-Karte, das auf einem Arduino Uno steckt [ARDUINO, 2013]

6.1.2. BeagleBoard und BeagleBone

Das *BeagleBoard* wurde von Texas Instruments entwickelt und basiert auf der herstellereigenen *Open Multimedia Applications Platform* (OMAP), einer Prozessorarchitektur für Einplatinen-Computer und mobile Endgeräte. Der SoC des BeagleBoard setzt sich aus einem ARM Cortex-A8-Prozessor mit 720 MHz, einem Grafikprozessor von PowerVR Technologies und einem digitalen Signalprozessor von Texas Instruments für beschleunigte Audio- und Video-Decodierung zusammen. Fest- und Arbeitsspeicher sind im Package-on-Package-Verfahren auf einem Chip vereint. Dieser besteht aus 256 MB NAND-Speicher und 256 MB RAM. Weiterer Festspeicher kann durch den integrierten SD-Kartenleser in Form von SD- und MMC-Karten zugeführt werden. Ein Expansion-Header ist ebenfalls verbaut verbaut, der Ports für SD/MMC, I²C, GPIO, SPI und UART bereitstellt. [COLEY, 2012a]

Eine erweiterte Version wird unter dem Namen *BeagleBoard-xM* vertrieben. Mit 1 GHz ist der integrierte Prozessor etwas schneller als der des BeagleBoard. Auch der Arbeitsspeicher ist mit 512 MB größer dimensioniert. Auf Festspeicher wurde verzichtet; ersatzweise befindet sich ein microSD-Kartenleser auf der Platine. Als externe Anschlüsse sind Ethernet, RS-232 und vier USB-Ports vorhanden, außerdem ein Expansion-Header und eine Kamera-Schnittstelle. Displays lassen sich wahlweise an die DVI-D- oder die S-Video-Buchse anschließen. [COLEY, 2012b]

Zwei weitere Nachkommen des BeagleBoard sind das *BeagleBone* und das *BeagleBone Black*. In beiden ist ein ARM Cortex-A8-Prozessor verbaut, der beim Beagle Bone, je nach Art der Stromversorgung, mit 500 oder 720 MHz arbeitet. Als Arbeitsspeicher kommen 256 MB RAM zum Einsatz. Festspeicher muss durch eine microSD-Karte extern zugeführt werden. Das BeagleBone Black unterscheidet sich vom Vorgänger durch einen schnelleren Prozessor mit 1 GHz, 512 MB Arbeitsspeicher und einem HDMI-Ausgang. Zusätzlich zum microSD-Kartenleser sind auch noch 2 GB Flash-Speicher verbaut. Auf das BeagleBone Black können

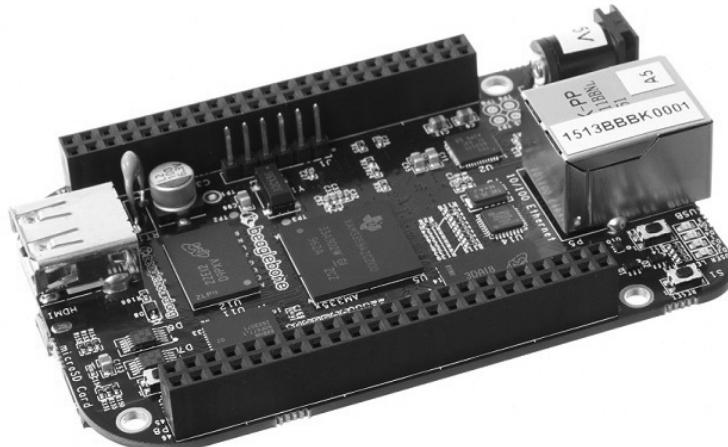


Abbildung 6.2.: Die kreditkartengroße Platine des BeagleBone Black [COLEY, 2013]

auch die Erweiterungsplatten der GNUBLIN-Plattform gesteckt werden. Für viele Anwendungsszenarien sind Module verfügbar, darunter Schnittstellen (z. B. AD-Wandler, RS-232 und RS-485) und Peripherie (z. B. Display, Relaiskarte oder Temperatursensor). [COLEY, 2013]

Für die verschiedenen BeagleBoard- und BeagleBone-Versionen wurden eine Reihe von Erweiterungsplatten entwickelt, die u. a. weitere Schnittstellen, wie Ethernet, RS-232 oder WLAN, hinzufügen. Auf allen Boards können verschiedene Linux-Distributionen betrieben werden.

6.1.3. Carambola

Das MIPS-basierte *Carambola* wird vom litauischen Herstellers 8devices angeboten. Der verbaute SoC ist ein Ralink RT3050 mit 320 MHz. Für das Betriebssystem und weitere Programme sind 8 MB Flash-Speicher und 32 MB Arbeitsspeicher vorgesehen.

Auf dem Board befinden sich drei Netzwerkadapter: ein WLAN-Modul (802.11b/g/n) und zwei Ethernet-Ports, von denen einer Power over Ethernet unterstützt. Die WLAN-Antenne ist bereits integriert. Die Transceiver und RJ45-Anschlüsse der Ethernet-Ports fehlen hingegen und müssen auf einer zweiten Platine nachgerüstet werden. Auch USB- und RS-232-Anschluss sind nicht auf dem Carambola-Board vorhanden und daher extern zu stellen. Die Abmessungen betragen nur 35×45 mm. Der Hersteller bietet ein Entwickler-Board an, auf dem die fehlenden Anschlüsse vorhanden sind. Externe Sensoren lassen sich auch an die GPIO-Ausgänge des RT3050 anschließen.

Carambola 2 ist eine leistungsstärkere Version mit Atheros AR9331 MIPS-SoC (400 MHz), 16 MB Flash-Speicher und 64 MB Arbeitsspeicher. Ein passendes Entwickler-Board ist ebenfalls verfügbar.

Die Linux-Distribution OpenWrt wird als Betriebssystem eingesetzt. Es ist speziell für eingebettete Systeme ausgelegt und verfügt über mehr als 2.000 Software-Pakete. Die Kon-

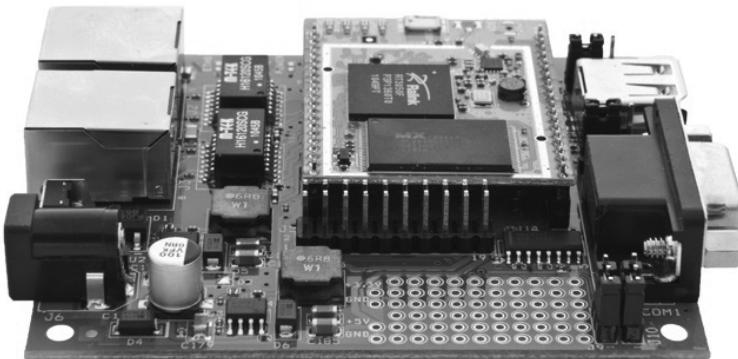


Abbildung 6.3.: Einplatinen-Computer Carambola (gelbe Platine) auf dem Entwickler-Board, das die Anschlüsse bereitstellt [8DEVICES, 2013]

figuration des Carambola-Boards kann mittels Web-Schnittstelle oder Telnet-Verbindung erfolgen. [8DEVICES, 2013]

6.1.4. Cubieboard

Das *Cubieboard* basiert wie viele andere Einplatinen-Computer auf der ARM-Architektur. Der SoC ist ein A10 von Allwinner, der einen Cortex-A8-Prozessor mit 1 GHz und 1 GB Arbeitsspeicher integriert. Neben einem microSD-Slot und einem SATA-Port für externen Festspeicher sind zusätzlich 4 GB NAND-Flash auf der Platine verbaut. Als weitere Anschlüsse stehen zwei USB-Ports, Ethernet, HDMI, Infrarot und ein Steckplatz mit 96 Pins zur Verfügung. Grundsätzlich können alle Betriebssysteme verwendet werden, die ARMv7 unterstützen. Praktisch sind dies vor allem Linux-Distributionen wie Ubuntu, Fedora oder Arch Linux. Der Nachfolger, Cubieboard 2, unterscheidet sich lediglich durch seinen Doppelkernprozessor.

6.1.5. GNUBLIN

Der Einplatinen-Computer *GNUBLIN* entstand 2007 an der Hochschule Augsburg als Lernplattform für Embedded Linux. Im Rahmen des Projektes wurden ein Reihe von Erweiterungsplatinen, Anwendungen und eine umfangreiche API für die Programmierung von Steuerungssystem, Datenloggern und Automatisierungen entwickelt. *GNUBLIN* wird in vier Varianten angeboten, die sich in Bauform sowie Art und Anzahl der Schnittstellen unterscheiden. Auf allen sind ARM-Prozessoren mit 180 MHz und 32 MB Arbeitsspeicher verbaut. Festspeicher kann in Form einer microSD-Karte zugeführt werden. Der USB-Bus dient auch als Erweiterungsschnittstelle für diverse LAN-, WLAN- und Bluetooth-Adapter. Die Erweiterungsplatinen können mit dem Raspberry Pi genutzt werden. Externe Sensoren und Schaltungen lassen sich an die I²C-, SPI- und UART-Schnittstellen anschließen. Auf der Variante *GNUBLIN-LAN* sind zusätzlich ein Ethernet-Anschluss und eine Echtzeituhr vorhanden. Als Betriebssystem wird eine angepasste Debian-Distribution verwendet. [EMBEDDED PROJECTS GMBH, 2013]

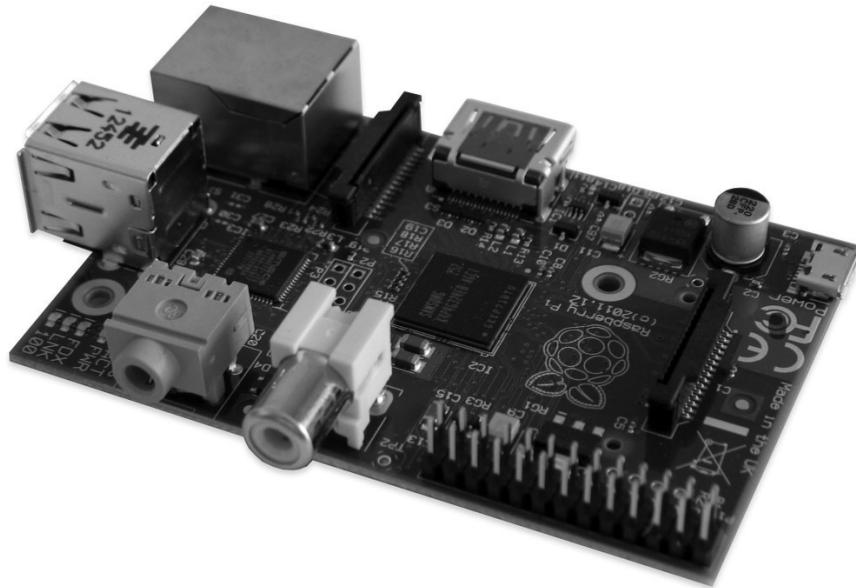


Abbildung 6.4.: Einplatinen-Computer Raspberry Pi (Modell B) mit Ethernet-Schnittstelle und zwei USB-Ports

6.1.6. PandaBoard

Das *PandaBoard* ist der Nachfolger des BeagleBoard und basiert auf dem OMAP4430-SoC von Texas Instruments, der mit einem 1 GHz Doppelkernprozessor (ARM Cortex-A9 MPCore) ausgestattet ist. Es ist 1 GB Arbeitsspeicher aber kein Festspeicher verbaut. Stattdessen muss auf den integrierten Kartenleser zurückgegriffen werden, der SDHC-Karten bis 32 GB unterstützt. Auf der Platine befinden sich auch noch ein 10/100-Mbit-Ethernet-Adapter sowie ein WLAN- und ein Bluetooth-Modul. Sensoren und andere externe Geräte lassen sich über zwei USB-Buchsen (Host und Device) und die GPIO-Pins an zwei Expansions-Ports anbinden. Viele Pins sind aber bereits für andere Funktionen reserviert, darunter den General Purpose Memory Controller zur Ansteuerung von Flash-Speicher oder für USB, I²C und UART. Für externe Displays stehen Anschlüsse für HDMI, DVI-D und eine LCD-Erweiterung zur Verfügung.

Mögliche Betriebssysteme sind Linux, OpenBSD und FreeBSD. Des Weiteren können auch Android und Firefox OS installiert werden, die für die Sensoransteuerung aber ungeeignet sind. RISCOS 5 ist unter OMAP4 lauffähig und damit auch auf dem PandaBoard; für das Echtzeitbetriebssystem QNX Neutrino kann die OMAP4-Unterstützung durch Pakete nachgerüstet werden.

Mit dem *PandaBoard ES* existiert noch eine leistungsstärkere Version des Einplatinen-Computers, die mit einem OMAP4460-SoC ausgestattet ist, dessen Haupt- und Grafikprozessor höhere Taktgeschwindigkeiten aufweisen.

	Model A	Model B
System on a Chip	Broadcom BCM2835	
CPU	700 MHz ARM1176JZF-S	
GPU	Broadcom VideoCore IV	
Arbeitsspeicher	256 MB SDRAM (shared)	512 MB SDRAM (shared) [*]
USB 2.0	1 Port (vom BCM2835)	2 Ports (integrierter Hub)
Nichtflüchtiger Speicher	SD-, MMC- und SDIO-Kartenleser	
Netzwerk	–	10/100 Mbit Ethernet
Schnittstellen	8 × GPIO, UART, I ² C, SPI, I ² S	
Video-Ausgänge	Composite Video, HDMI, DSI	
Audio-Ausgänge	3,5 mm Klinkenbuchse, HDMI, I ² S	
Leistungsaufnahme	5 V, 500 mA (2,5 Watt)	5 V, 700 mA (3,5 Watt)
Stromversorgung	5 V Micro-USB, 4 × AA-Batterien	
Abmessungen, Gewicht	8,6 cm × 5,4 cm × 1,5 cm, 45 g	
Preis	25,- €	35,- €

* 256 MB in der ersten Revision des Modells

Tabelle 6.2.: Spezifikationen der beiden Modellvarianten des Raspberry Pi [RASPBERRY PI FOUNDATION, 2013]

6.1.7. Raspberry Pi

Der *Raspberry Pi* wurde von der britischen Raspberry Pi Foundation entwickelt und ist in zwei Modellvarianten (A und B) erhältlich. Diese werden in Lizenz von drei Distributoren gefertigt. [RASPBERRY PI FOUNDATION, 2013] Auf der etwa kreditkartengroßen Platine ist ein ARM-basierter SoC von Broadcom verbaut, der Hauptprozessor, Grafikprozessor und Arbeitsspeicher vereint.

Der Prozessor arbeitet mit einer Taktgeschwindigkeit von 700 MHz, der durch Übertakten temporär auch 1 GHz erreichen kann. Als Festspeicher werden handelsübliche SD- und SDHC-Karten verwendet. Der Zugriff erfolgt durch den integrierten Kartenleser. Das günstigere Modell A ist mit 256 MB Arbeitsspeicher ausgestattet, das Modell B seit der zweiten Revision mit 512 MB. Im Modell B ist auch ein integrierter USB-Hub verbaut, an den zwei USB-Ports und ein Ethernet-Controller angeschlossen sind. Das Modell A besitzt nur einen einzigen USB-Port, der direkt mit dem SoC verbunden ist. Ein Hub oder ein Ethernet-Adapter kann über diesen Port nachgerüstet werden. WLAN- und Bluetooth-Unterstützung ist ebenfalls über USB erweiterbar. [SCHMIDT, 2012]

Externe Sensoren lassen sich an USB, Ethernet und GPIO an den Raspberry Pi anschließen. Die Kommunikation über GPIO kann auf verschiedenen Standards basieren, darunter Serial Peripheral Interface (SPI) und Inter-Integrated Circuit (I²C). Für externe Displays stehen drei Anschlüsse zur Verfügung: HDMI, MIPI DSI und Composite Video. Kameras können an USB und MIPI CSI angeschlossen werden. Ein 5-Megapixel-Modul für die CSI-Schnittstelle

Klasse	Max. Leistung	Reichweite
1	100 mW / 20 dBm	100 m
2	2,5 mW / 4 dBm	50 m
3	1 mW / 0 dBm	10 m

Tabelle 6.3.: Leistungsklassen von Bluetooth [GESSLER U. KRAUSE, 2009]

ist seit Mai 2013 erhältlich. Die Platine ist auch mit einem Audioausgang ausgestattet. Soll im Gefahrenfall ein akustisches Alarmsignal ausgelöst werden, kann der Ausgang mit einem Lautsprecher, ggf. mit Vorverstärker, verbunden werden. Die weiteren technischen Spezifikationen der beiden Modellvarianten sind in Tabelle 6.2 aufgelistet.

Eine Reihe von Betriebssystemen ist auf dem Raspberry Pi lauffähig, darunter mehrere Linux-Distributionen. Da der Raspberry Pi ohne BIOS und Echtzeituhr auskommt, muss entweder ein externer Zeitgeber angeschlossen oder auf Netzwerkdienste, wie dem Network Time Protocol (NTP), zurückgegriffen werden, um ein physikalisches Zeitsignal und damit die aktuelle Uhrzeit zu erhalten. Viele Betriebssysteme können bereits beim Systemstart mittels NTP die aktuelle Uhrzeit setzen. Dafür ist aber in jedem Fall eine Netzwerkverbindung zu einem NTP-Server nötig.

Die zahlreichen Erweiterungsplatinen (Shields) für die Arduino-Plattform lassen sich auch mit dem Raspberry Pi nutzen. Die dazu nötige Verbindungsplatine, z. B. Alemode oder Embedded Pi, wird auf die GPIO-Schnittstelle gesteckt. Anschließend können beispielsweise Messdaten auf dem Raspberry Pi ausgewertet werden, die die Verbindungsplatine von den angeschlossenen Shields erfasst. Das Alemode-Board stellt dem Raspberry Pi darüber hinaus auch eine Echtzeituhr zur Verfügung. Die Erweiterungsplatinen für das GNUBLIN sind mit Hilfe eines „Module-GnuPi“-Boards verwendbar.

6.2. Kommunikation

Der Messclient muss mit Schnittstellen und Adapter für die Kommunikation mit Sensoren und die Anbindung an ein Netzwerk zur Datenübertragung ausgestattet sein. Als Übertragungsmedien werden Kabel, Funk- oder Lichtwellen genutzt. Einige Schnittstellen dienen sowohl für die Sensoransteuerung als auch für die Netzwerkanbindung (z. B. Ethernet und Bluetooth). Eine nähere Betrachtung der einzelnen Standards umreißt die Eigenschaften und die Einsatzfelder bei geodätischen Überwachungsmessungen.

6.2.1. Bluetooth

Bluetooth ist ein standardisiertes Verfahren (IEEE 802.15.1) zur drahtlosen Datenübertragung auf kurzen Distanzen in Wireless Personal Area Networks (WPAN). Es arbeitet im lizenzzfreien 2,4 GHz ISM-Band. Das Frequenzband wird auch von anderen Geräten, darunter Garagentüröffnern, Autozentralverriegelungen oder Schnurlosemobiltelefonen, genutzt, so dass



Abbildung 6.5.: Das Neigungsmessgerät BlueCLIO von Wyler ist mit einer Bluetooth-Schnittstelle ausgerüstet

Störungen nicht ausgeschlossen sind. Um dem entgegenzuwirken werden Frequenzsprungverfahren eingesetzt, die für eine gleichmäßige Nutzung sorgen. Die Sendeleistung von Bluetooth ist beschränkt und wird in drei Klassen unterteilt (Tabelle 6.3).

Erlaubt sind Punkt-zu-Punkt- und Punkt-zu-Multipunkt-Verbindungen. Ein Gerät arbeitet als Master, zu dem sich als Slave agierende Geräte verbinden. Diese bilden zusammen ein Piconet. Mehrere Piconets lassen sich zu einem Scatternet zusammenfassen. Ad-hoc-Netzwerke werden ebenfalls unterstützt. Damit können sich Netze automatisch konfigurieren, ohne, dass die kommunizierenden Geräte einander bekannt sein müssen. Der Datenaustausch zwischen Geräten basiert auf Profilen. Diese definieren Regeln und Protokolle für verschiedene Einsatzmodelle. Für jeden Dienst wird ein separates Profil verwendet. Dazu gehören u. a. Druckanwendungen, die Übertragung von Audiodaten oder der Nachrichtenaustausch zwischen Geräten. [GESSLER U. KRAUSE, 2009]

Die Verbreitung von Bluetooth in der Industrie führte auch zur Integration von entsprechender Schnittstellen in einigen geodätischen und geotechnischen Sensoren. Beispiele für die Datenübertragung per Bluetooth in der Messtechnik sind der Laserdistanzmesser Leica DISTO D3a BT, der Neigungssensor BlueCLIO von Wyler (Abbildung 6.5) und meteorologische Sensoren für Temperatur, Druck oder Beschleunigung verschiedener Hersteller. Einige Tachymeter der großen Hersteller Leica Geosystems, Trimble, Sokkia und Topcon sind ebenfalls mit einer entsprechenden Schnittstelle ausgestattet, die zum Teil aber nur für die Geräte-Fernsteuerung und nicht für den Datenaustausch konzipiert ist.

6.2.2. Ethernet

Ethernet ist weltweit die am häufigsten verwendete kabelgebundene Local-Area-Network-Technologie und umfasst Schnittstellen, Kabel und Protokolle. Entwickelt wurde Ethernet bereits Anfang der 1970er Jahre. Es ist als IEEE 802.3 standardisiert. Im Zuge der Entwicklung entstanden verschiedene Varianten, wobei die Basistransporteinheit konstant blieb. Die Architektur setzt auf die beiden unteren Schichten des ISO-/OSI-Referenzmodells auf (Bitübertragungsschicht und Sicherungsschicht). Der Datenaustausch erfolgt über Datenframes. [RIGGERT, 2012]

Bei Ethernet kommen Kupfer- und Glasfaserkabel als Übertragungsmedium zum Einsatz. Je nach verwendetem Ethernet-Standard unterscheiden sich Medium, Topologie, Datenrate, Kabeltyp und Segmentlänge. Beim verbreiteten Fast-Ethernet-Standard mit Twisted-Pair-Kabeln beträgt die Datenrate 100 Mbit/s, bei einer Segmentlänge von maximal 100 m. [SPURGEON, 2000] Mit Power over Ethernet können Endgeräte über das Ethernet-Kabel auch mit Strom versorgt werden. Abschnitt 6.3.4 soll darauf genauer eingehen.

Vorhandene Stromleitungen lassen sich mit der *PowerLAN* genannten Technik ebenfalls als Medium nutzen. Diese Trägerfrequenzanlage nutzt spezielle Adapter, die in eine Steckdose gesteckt und über eine Ethernet-Schnittstelle mit dem Endgerät verbunden werden. Da das System nicht standardisiert ist, vertreiben mehrere Hersteller proprietäre Lösungen, die zum Teil zueinander inkompatibel sind. PowerLAN erlaubt eine maximale Datenrate von 500 Mbit/s, bei einer Reichweite von bis zu 300 m. [HORAK, 2007]

Es ist auch möglich, Ethernet-Netzwerke mit WLAN-Netzwerken zu verbinden. Für diesen Zweck sind externe WLAN-Adapter erhältlich, die durch ein Netzwerkabel mit der Ethernet-Schnittstelle verbunden oder direkt auf die RJ-45-Buchse gesteckt werden.

Ethernet eignet sich nicht nur für die Netzwerk- und Internet-Anbindung, sondern auch für die Sensorkommunikation. Einige meteorologische Sensoren, z. B. von ALLNET, Gude, Neol oder pro-ite, lassen sich in ein Ethernet-Netzwerk integrieren (Abbildung 6.6). Die Ansteuerung erfolgt über die IP-Adresse des Sensors.

6.2.3. Free Space Optics

Free Space Optics ist eine Technologie zur kabellosen Datenübertragung über mehrere hundert Meter bis einige Kilometer mit Hilfe von Licht. Die Technik ist auch als optischer Richtfunk bekannt, obwohl strenggenommen keine Radiowellen ausgesendet werden. Die vollduplex Punkt-zu-Punkt-Verbindungen werden stattdessen mit Lasern aufgebaut, die Datenraten von mehr als 1 Gbit/s erlauben; bei Nutzung von optischen Wellenlängenmultiplexverfahren auch deutlich mehr. Kommerzielle Systeme arbeiten im infraroten Spektrum, mit Wellenlängen zwischen 800 und 1550 nm.

Die Technik kann überall dort eingesetzt werden, wo Hochgeschwindigkeitsverbindungen gefordert sind, die Verlegung von Ethernet- oder Glasfaserkabeln aber nicht möglich ist. Anwendungsgebiete sind z. B. LAN-zu-LAN-Brücken, die zwei Ethernet-Netzwerke miteinander verbinden, oder Breitband-Zugänge in abgelegenen Gebieten. Nachteilig ist der hohe



Abbildung 6.6.: Das Triometer ALL3418 von ALLNET dient der netzwerkbasierten Erfassung von Temperatur und Luftfeuchte. Die Messwerte des Sensors werden als HTML-Seite über HTTP ausgegeben. Die Spannungsversorgung ist optional auch über Power over Ethernet möglich.

Einfluss von atmosphärischen und meteorologischen Störungen, die das Signal dämpfen und die Reichweite reduzieren. [KARTALOPOULOS, 2011]

Die Kosten sind abhängig von der geforderten Entfernung und Datenrate. Einfache Systeme sind ab etwa 2.000 € erhältlich. Mit „Reasonable Optical Near Joint Access“ (RONJA) existiert auch eine LED-basierte Low-Cost-Lösung mit einer maximalen Reichweite von 1,4 km. Die technischen Spezifikationen stehen unter einer Open-Source-Lizenz und die Materialkosten sollen sich lediglich auf etwa 100 € belaufen. [RONJA, 2013]

6.2.4. General Purpose Input/Output

Bei *General Purpose Input/Output* (GPIO) handelt es sich um die Schnittstelle eines Prozessors, die als digitaler Ein- oder Ausgang konfiguriert ist. Die genaue Verwendung ist nicht vordefiniert und wird von der Bauart bestimmt. GPIO kann z. B. Interrupts auslösen, direkten Speicherzugriff erlauben oder andere Bus-Systeme bereitstellen und wird häufig zu Ports mit 16 und mehr Pins zusammengefasst (Abbildung 6.7). Der Spannungsbereich entspricht üblicherweise dem des Prozessors (z. B. 0 – 3,3 V oder 0 – 5 V). [SCHRÖDER U. A., 2009]

Externe Geräte, die an GPIO angeschlossen werden, nutzen für die Kommunikation oft spezielle Bus-Systeme. Die folgenden seriellen Busse sind weit verbreitet und werden auch von vielen Einplatinen-Computern unterstützt:

Controller Area Network (CAN) ist ein asynchroner serieller Bus, der zu der Gruppe der Feldbusse gehört. Er wurde 1983 von Bosch für die Anbindung von Sensoren und Steuersystemen in der Automobiltechnik entwickelt und ist heute ein Industriestandard für die Vernetzung sicherheitskritischer Anlagen.

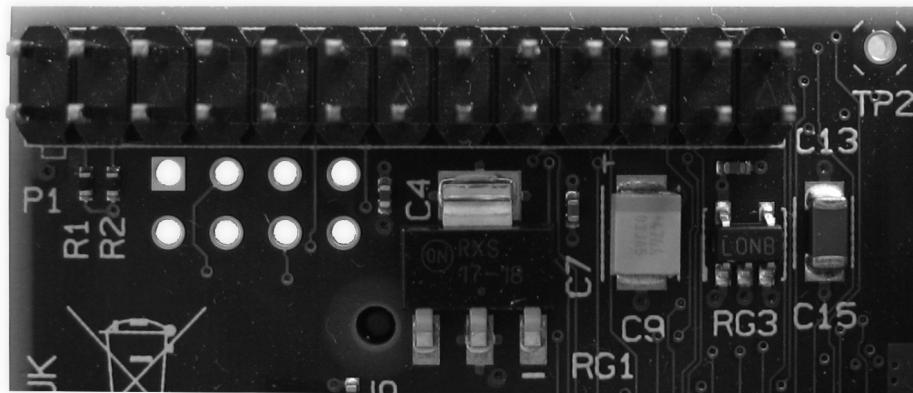


Abbildung 6.7.: Beispiel für eine GPIO-Schnittstelle mit 26 Pins

Inter-Integrated Circuit (I²C) ist ein von Philips Anfang der 1980er Jahre entwickelter serieller Datenbus, der für die Kommunikation zwischen verschiedenen digitalen Schaltungen, wie Mikrocontrollern und ICs, konzipiert ist.

Serial Peripheral Interface (SPI) ist ein synchrones unidirektionales serielles Bus-System, das von Motorola entwickelt wurde. Die Kommunikation erfolgt nach dem Master-Slave-Prinzip. Durch die Voll-Duplex-Übertragung und den hohen Takt sind Übertragungsraten von bis zu 15 Mbit/s möglich.

Universal Asynchronous Receiver/Transmitter (UART) ist eine asynchrone serielle Schnittstelle, die u. a. für Datenübertragungen benutzt wird. Sie dient auch als Grundlage für RS-232 und RS-485.

Einfache Sensoren aus dem Low-Cost-Bereich, die als integrierte Schaltung auf einer einzelnen Platine vertrieben werden, lassen sich problemlos über GPIO ansteuern. Erweiterungsboards für Einplatinen-Computer sind mit Pfostensteckverbindern ausgestattet, um sie direkt auf den GPIO-Port stecken zu können.

6.2.5. Mobiles Internet

Überwachungsmessungen werden auch an Orten durchgeführt, an denen keine Ethernet- und WLAN-basierten Internetzugänge vorhanden sind. Anbindungen mit hoher Übertragungsgeschwindigkeit sind seit der Einführung von UMTS aber in vielen Gebieten Europas verfügbar. Für die Übertragung von Messdaten zu einem Internet-Server kann bereits der GPRS-Standard ausreichend sein. Ein Zugang zum mobilen Internet ist nach [SAUTER, 2011] über folgende Systeme möglich:

Global System for Mobile Communication (GSM) ist der Nachfolger des C-Netzes und der am weitesten verbreitete Mobilfunkstandard. Mit der Übertragungsart Circuit Switched Data (CSD) sind auch Datenübertragungen möglich, wobei nur maximal 14,4 kbit/s erreicht werden. Höhere Datenraten leisten HSCSD und GRPS.

High Speed Circuit Switched Data (HSCSD) ist eine Erweiterung von CSD und erlaubt durch Kanalbündelung maximal 57,6 kbit/s. Für die Nutzung von HSCSD sind spezielle Endgeräte nötig. Deren Verbreitung nimmt seit der Einführung von HSDPA weiter ab, so dass der Standard in Zukunft obsolet wird.

General Packet Radio Service (GPRS) dient der paketorientierten Datenübertragung in GSM-Netzen. In der Praxis werden Übertragungsgeschwindigkeiten von maximal 50 kbit/s erreicht. GPRS ist heute flächendeckend verfügbar.

Enhanced Data Rates for GSM Evolution (EDGE) erweitert die GSM-Technik um eine neue Modulationsart. Der Datendienst Enhanced GPRS (E-GPRS) erreicht Datenraten von bis zu 270 kbit/s. Infolge des parallelen UMTS-Ausbau wird EDGE aber nicht von allen Netzbetreibern nachgerüstet.

High Speed Packet Access (HSPA) wird für Internetverbindungen in UMTS-Netzen eingesetzt. Die Übertragungsraten sind mit denen von DSL vergleichbar und erreichen mit dem aktuellen Standard HSPA+ in der Praxis maximal 42 Mbit/s (Downlink) und 5,76 Mbit/s (Uplink). UMTS ist oft nur in Ballungszentren verfügbar; in vielen ländlichen Regionen ist die Abdeckung hingegen sehr gering oder nicht vorhanden.

Long Term Evolution (LTE) ist ein Mobilfunkstandard der vierten Generation und unterstützt Bandbreiten zwischen 10 und 300 Mbit/s im Downlink und 5 bis 75 Mbit/s im Uplink. In vielen deutschen Großstädten ist LTE bereits verfügbar. Der Ausbau ist aber noch nicht abgeschlossen.

Um die Mobilfunkstandards für die Internetanbindung nutzen zu können, sind passende Endgeräte notwendig. Das Tethering-Verfahren kann genutzt werden, um eine Verbindung zwischen einem Rechner und einem Mobiltelefon aufzubauen. Das Mobiltelefon arbeitet dabei als Modem, das per USB-Kabel, IrDA oder Bluetooth mit dem Rechner verbunden ist. Alternativ lässt sich mit WLAN auch ein Ad-hoc-Netz aufbauen, bei dem das Mobiltelefon zum Hotspot wird und die Internetverbindung teilt. Eine weitere Möglichkeit ist die Nutzung von USB-Adaptoren. Sie werden nach dem Einsetzen der SIM-Karte an die USB-Schnittstelle des Rechners angeschlossen und dienen als Modem. Ähnlich funktionieren auch UMTS-Router, die mehreren Endgeräten über eine gemeinsame Verbindung den Zugriff auf das Internet ermöglichen. Für einige Einplatinen-Computer sind auch passende GPRS- und UMTS-Erweiterungsboards verfügbar (Abbildung 6.1).

Ein Problem bei der Nutzung des mobilen Internets stellen die Verbindungstrennungen dar, die von den Providern zwangsweise in regelmäßigen Abständen durchgeführt werden und zu der Vergabe einer neuen dynamischen IP-Adresse führen. Da der Messclient im Internet nur über die IP-Adresse und nicht über eine Domain erreichbar ist, wird ein Zugriff von Außen erschwert. Die Provider bieten zwar auch die Vergabe einer festen IP-Adresse an, die der SIM-Karte bei jeder Einwahl zugewiesen wird. Aufgrund der stetig abnehmenden Zahl freier IPv4-Adressen sind diese aber teuer. Dieser Umstand wird sich erst mit der flächendeckenden Einführung von IPv6-Adressen ändern, deren Adressraum erheblich größer ist. Ein Rückgriff auf DynDNS-Anbieter kann das Problem der wechselnden Adressen umgehen. Der Messclient teilt dem Anbieter nach jeder Trennung die neue IP-Adresse mit, die anschließend mit einer Domain verknüpft wird. Über die Domain ist der Messclient jederzeit und weltweit erreichbar.



Abbildung 6.8.: Ein *Serial Device Server* von Digi International. Der Adapter wird an die 9-polige D-Sub-Buchse einer RS-232/422/485-Schnittstelle angeschlossen und überträgt Daten kabellos per 802.11b auch über größere Distanzen. Die Konfiguration erfolgt über ein Web-Interface. [DIGI INTERNATIONAL INC., 2013]

Viele Provider setzen aber Proxy-Server ein, die den Zugriff von Außen grundsätzlich blockieren. Eine Lösung kann die Nutzung eines virtuellen privaten Netzwerkes (VPN) oder eines SSH-Tunnels sein. Mit beiden Systemen lassen sich die Proxy-Server der Provider überwinden und verschlüsselte bidirektionale Verbindungen zum Messclient aufzubauen. Router mit integriertem VPN-Client richten solche Verbindungen auch selbstständig ein. Mit OpenVPN steht darüber hinaus eine verbreitete und kostenfreie Software-Lösung zur Verfügung.

6.2.6. RS-232

RS-232 ist ein weit verbreiteter Standard für serielle Schnittstellen. Er wurde bereits 1962 eingeführt und ist auch heute noch aufgrund der niedrigen Anforderungen an Hard- und Software in industriellen Anwendungsbereichen verbreitet. Für die Datenübertragung wird ein einfaches asynchrones serielles Verfahren verwendet, bei dem die einzelnen Bits des zu übertragenden Bytes sequenziell über eine Datenleitung gesendet werden. Sender und Empfänger müssen durch den gleichen internen Takt und durch ein Startbit synchronisiert werden, weil keine Takteleitung vorhanden ist. [SCHLEMMER, 1996] Die maximale Entfernung zwischen zwei RS-232-Geräten ist wie bei allen seriellen Übertragungsverfahren überwiegend von der Baudrate und dem verwendeten Kabel abhängig. Bei RS-232 liegt sie zwischen weniger als 2 m (115.200 Baud) und 900 m (2.400 Baud). [SCHRÖDER U. A., 2009] Werden Bluetooth-, Ethernet- oder WLAN-Konverter (Serial Device Server) an die RS-232-Schnittstelle angeschlossen, ist eine Datenübertragung auch über größere Distanzen oder per Funk möglich (Abbildung 6.8).

Leica Geosystems setzt für die Kommunikation mit den herstellereigenen Sensoren auch auf die serielle Schnittstelle. Es finden aber keine 9-poligen D-Sub-Stecker sondern 5-polige und 8-polige Rundsteckverbinder von LEMO Verwendung. Mittels Konverterkabel ist eine Anbindung an den RS-232- oder USB-Port eines Computers möglich. Zum Aufbau von



Abbildung 6.9.: Beispiel für einen USB-Konverter mit zwei Ports für RS-232, RS-422 und RS-485 [EXSYS VERTRIEBS GMBH, 2013]

Sensornetzwerken mit dem Leica Nivel220 kommt hingegen die RS-485-Schnittstelle zum Einsatz, weil diese im Gegensatz zu RS-232 nicht nur Punkt-zu-Punkt-Verbindungen erlaubt, sondern auch einen gemeinsamen Bus für mehrere Geräte bereitstellt.

6.2.7. USB

Der *Universal Serial Bus* (USB) ist ein serieller Bus, der 1996 von Intel entwickelt wurde, um die wachsende Zahl externer Peripheriegeräte im PC-Bereich mit einer einheitlichen Schnittstelle zu versorgen. Die Verwendung von anderen Schnittstellen, wie RS-232, PS/2 oder IEEE 1284, ist nach Einführung in vielen Bereichen obsolet geworden.

Die Spezifikation USB 1.0 definiert eine Geschwindigkeit von 12 Mbit/s, die mit USB 2.0 auf 480 Mbit/s und mit USB 3.0 auf 4.000 Mbit/s erhöht wurde. Die Struktur des Busses ist ein Baum, an dessen Wurzel ein zentraler Host-Controller sitzt, der für die Koordination der angeschlossenen Geräte zuständig ist. Theoretisch lassen sich bis zu 127 verschiedene Geräte (Slaves) an einem Host-Controller betreiben. Der Host-Controller identifiziert angesteckte Geräte und lädt automatisch gerätespezifische Treiber. Es darf immer nur ein Gerät an einem USB-Port angeschlossen sein. Mehrere Geräte erfordern die Nutzung eines Hubs.

Die Datenübertragung erfolgt symmetrisch über zwei verdrillte Leitungen, um Gleichtaktstörungen zu unterdrücken und eine hohe Übertragungsrate zu ermöglichen. Endgeräte können über eine Versorgungsleitung mit einem Strom von bis zu 500 mA versorgt werden. Die maximale Leitungslänge ist auf 5 m begrenzt. [SCHRÖDER U. A., 2009] Wie bei RS-232 kann auch bei USB die Reichweite durch Serial Device Server auf mehrere hundert Meter erhöht werden.

Andere Schnittstellen, darunter RS-232, RS-422 und RS-485, lassen sich mittels passender Konverter an einem USB-Port betreiben (Abbildung 6.9). Daher eignet sich USB auch für die Sensoransteuerung, wenn an den Sensoren keine USB-Schnittstelle vorhanden ist. Für

Eigenschaft	WLAN (IEEE 802.11b)	Bluetooth (IEEE 802.15.1)	ZigBee (IEEE 802.15.4)
Verfahren	DSSS*	FHSS†	DSSS*
Datenrate	11 Mbit/s	1 Mbit/s	250 kbit/s
Reichweite	140 m	100 m	1500 m
Datengröße	unbeschränkt	unbeschränkt	kleine Datenpakete
Komplexität	komplex	sehr komplex	einfach

* Direct Sequence Spread Spectrum

† Frequency Hopping Spread Spectrum

Tabelle 6.4.: Vergleich zwischen WLAN, Bluetooth und ZigBee

viele Übertragungsstandards, wie Bluetooth, Ethernet und WLAN, sind auch USB-Adapter verfügbar.

6.2.8. WLAN

Unter *Wireless Local Area Network* (WLAN) wird im Allgemeinen ein Funkstandard für Ethernet-Netzwerke verstanden, der durch die Normen-Familie IEEE 802.11 beschrieben wird. Die Standards wurden immer wieder erweitert und weisen unterschiedliche Spezifikationen auf. Als Frequenzblöcke sind 2,4 GHz und 5 GHz aus dem ISM-Band vorgesehen.

Bei WLAN wird zwischen den beiden Netzwerk-Topologien Ad-hoc-Netze und Infrastruktur-Modus unterschieden:

Ad-hoc-Netzwerke sind die einfachste Art, ein WLAN aufzubauen. Dabei wird eine direkte Verbindung zwischen den Teilnehmern hergestellt. Die Teilnehmer kommunizieren nur untereinander und müssen sich daher auch gegenseitig erreichen können.

Infrastruktur-Netzwerke werden durch eine feste Basisstation, ein sogenannter Access Point, koordiniert. Die Teilnehmer müssen sich im Funkbereich eines Access Point befinden, der in der Regel ein WLAN-Router ist. Mehrere Access Points können per Ethernet verbunden werden, um die Reichweite des Netzes zu erhöhen.

Wireless Distribution System beschreibt einen Spezialfall des Infrastruktur-Modus, bei dem die Access Points nicht über Kabel sondern per Funk miteinander verbunden sind.

Die Reichweite von WLAN-Netzwerken lässt sich mit Repeatern weiter erhöhen. Diese arbeiten wie Signalverstärker, die eingehende Daten eines Clients an den nächsten Access Point weiterleiten und umgekehrt – mit der Einschränkung, dass sich die Datenrate halbiert.

Um den Datenverkehr abzusichern wurden verschiedene Standards zur Verschlüsselung entwickelt. Das älteste und unsicherste Verfahren ist Wire Equivalent Privacy (WEP). Aufgrund des schwachen Verschlüsselungsalgorithmus wird WEP nicht mehr als sicher angesehen und sollte gemieden werden. Der Nachfolger, Wi-Fi Protected Access (WPA), setzt auf

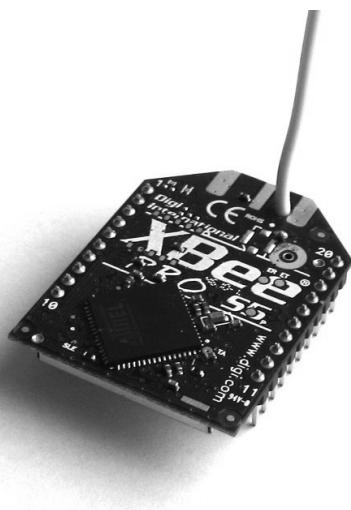


Abbildung 6.10.: Das ZigBee-Modul „XBee-PRO 868“ mit Drahtantenne von Digi International

Authentifizierungs-Server oder Pre Shared Keys. Der aktuelle Sicherheitsstandard ist IEEE 802.11i (WPA2), der AES als Verschlüsselungsverfahren nutzt. [METTER U. BUCHER, 2007]

WLAN eignet sich insbesondere für die funkbasierte Anbindung des Messclients an das Internet. Vor allem auf größeren Baustellen sind häufig WLAN-Netzwerke vorhanden. Diese arbeiten in der Regel mit Network Address Translation, weshalb für den Zugriff von Außen ein virtuelles privates Netzwerk oder ein SSH-Tunnel notwendig ist. Beide Verfahren verschlüsseln die Kommunikation und erhöhen damit die Sicherheit signifikant. Bei Nutzung von Verschlüsselungssystemen wird auch die Nutzung öffentlicher Access Points interessant, die vor allem in Großstädten immer häufiger zu finden sind und beim Bauwerksmonitoring im innerstädtischen Bereich eine einfache aber unsichere Möglichkeit der Internetanbindung darstellen können. Mit WLAN-basierten Serial Device Servern lassen sich auch serielle Verbindungen über Funk aufbauen. Indirekt eignet sich WLAN daher auch zur Sensoransteuerung. Erste Sensoren, beispielsweise das Tachymeter Leica Nova TM50, sind bereits nativ mit einer WLAN-Schnittstelle ausgestattet.

6.2.9. ZigBee

ZigBee ist ein Funknetzwerk für kurze Distanzen und geringe Übertragungsvolumen. Der Industriestandard wurde von der ZigBee Alliance entwickelt, einem Zusammenschluss von mehr als 200 Technologieunternehmen. Die Kernprotokolle werden durch IEEE 802.15.4 und die darin definierten Parameter, wie Frequenzen, Modulation und Kanalzugriff, beschrieben. ZigBee arbeitet in den Bändern 868 MHz, 915 MHz und 2,4 GHz. Die maximale Datenrate beträgt 250 kbit/s. [GESSLER U. KRAUSE, 2009]

Das ISM-Band muss sich ZigBee mit anderen Sendern, darunter Wireless LAN und Bluetooth, teilen. Diese leistungsstärkeren Sender können die Signalqualität von ZigBee-Geräten massiv stören. Ein Überschneiden der Frequenzbänder muss daher nach Möglichkeit verhindert werden. Tabelle 6.4 vergleicht die gängigen Funkstandards miteinander.

ZigBee zielt auf batteriebetriebene Anwendungsbereiche ab, bei denen niedrige Datenraten, niedrige Kosten und lange Batterielaufzeiten entscheidend sind. In vielen Fällen befinden sich die ZigBee-Module die meiste Zeit in einem Schlafmodus, aus dem sie nur aufwachen, wenn sie für einen kurzen Moment Daten übertragen. Dadurch sind Batterielaufzeiten von mehreren Jahren realisierbar. [FARAHANI, 2008] Die Anwendungsgebiete sind vor allem in der Industrie- und Automatisierungstechnik, der Medizintechnik und in Wireless Sensor Networks zur Gebäudeüberwachung zu finden.

Je nach Gerätetyp sind verschiedene Netzwerktopologien möglich. Die Netzwerke können in Stern-, Baum- oder Maschenstruktur aufgebaut sein. Es wird zwischen drei Gerätetypen unterschieden:

ZigBee Coordinator: Eröffnet ein Personal Area Network (PAN) mit einer Identifikationsnummer und verwaltet das Netzwerk. ZigBee Router und End Devices verbinden sich zum Coordinator.

ZigBee Router: Dient der Vergrößerung des PAN-Netzwerkes. Er schließt sich einem PAN an und bildet mit dem Coordinator den Kopf einer Baumstruktur.

ZigBee End Device: Verbindet sich mit einem Router oder direkt mit dem Coordinator und hat nur einen reduzierten Funktionsumfang.

Für die Nutzung von ZigBee ist spezielle Hardware erforderlich. Unter dem Namen XBee wird eine Sender-Familie des Unternehmens Digi International vertrieben, die sich in der Kombination von Hardwarekomponenten, Firmware, Reichweite und Antenne unterscheiden. Andere ZigBee-Module werden u. a. von Atmel, Freescale, Panasonic und Texas Instruments produziert. Von Drittherstellern sind auch USB-Adapter und Shields für die Arduino-Plattform erhältlich. [FALUDI, 2011]

6.3. Stromversorgung

Die Versorgung des Messclients mit elektrischem Strom kann nicht immer über das Verbundnetz der Elektrizitätsversorger garantiert werden. Nicht überall sind Netzanschlüsse verfügbar, mitunter ist auch mit Ausfällen zu rechnen. Der Einzelfall erfordert eine individuelle Konzeptionierung der Stromversorgung. Neben einem geeigneten Netzgerät soll der Messclient auch über eine unterbrechungsfreie Stromversorgung verfügen. Als weitere Versorgungssysteme sind Power over Ethernet und Photovoltaik-Anlagen zu betrachten.

6.3.1. Netzgerät

Die Leistungsaufnahme des Raspberry Pi ist mit 5 V und 700 mA (3,5 Watt) gering. Der Einplatinen-Computer besitzt kein eigenes Netzteil, so dass die Stromversorgung extern erfolgen muss. Dazu ist auf der Platine eine Micro-USB-Buchse vorhanden, an die ein Netzteil angeschlossen wird. Alternativ dazu können auch vier AA-Batterien oder der USB-Port eines Computers den nötigen Strom liefern. Im Handel sind passende Steckernetzteile für Kleinverbraucher bereits für 10 € verfügbar. Leistungsstärkere Netzteile mit 10 W und mehr besitzen mehrere Ausgänge und können so auch weitere Endgeräte mit Strom versorgen.

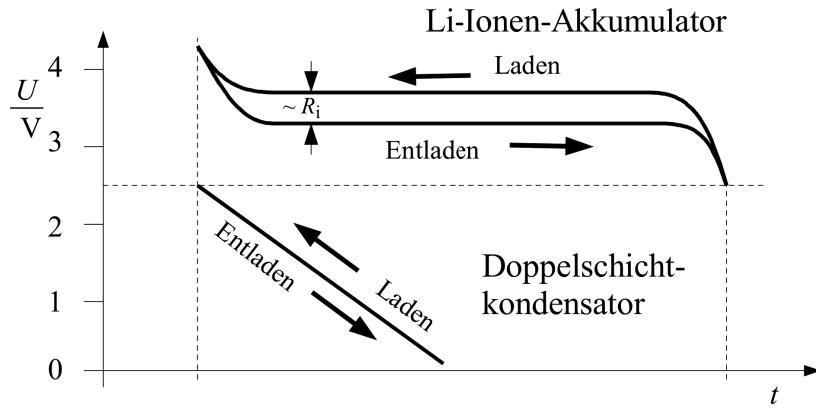


Abbildung 6.11.: Vergleich der Lade-Entlade-Spannung eines Lithium-Ionen-Akkumulators und eines Doppelschichtkondensators im Konstantstrombetrieb [SPECOVIUS, 2010]

6.3.2. Unterbrechungsfreie Stromversorgung

Eine *Unterbrechungsfreie Stromversorgung* (USV) ist eine Anlage zur Sicherstellung einer konstanten und dauerhaft verfügbaren Stromversorgung. Hauptaufgabe ist der Schutz der angeschlossenen Geräte vor Netzausfällen. Je nach Bauart werden auch

- Über- oder Unterspannung,
- Spannungsseinbrüche, Spannungsspitzen,
- Störspannungen,
- Frequenzänderungen und harmonische Oberschwingungen

gefiltert. [SPECOVIUS, 2010] Die detaillierte Klassifizierung erfolgt anhand des Betriebsverhaltens nach DIN EN 50091-3. Es existieren drei allgemeine USV-Typen: statische, rotierende und statisch-rotierende (hybride). Rotierende und hybride USV gewinnen ihren Strom aus der kinetischen Energie von Schwungscheiben und werden v. a. bei Großverbrauchern eingesetzt. [EMADI U. A., 2004] Statische Systeme lassen sich hinsichtlich ihrer Arbeitsweise in Online-USV, Offline-USV und netzinteraktive USV unterteilen. Bei einem Netzausfall muss die Offline-USV auf den Sekundärstromkreis umschalten, wodurch eine Latenz von wenigen Millisekunden auftritt („Schaltpause“). Netzinteraktive USV funktionieren ähnlich wie Offline-USV, sie leiten den Netzstrom direkt an den Verbraucher weiter. Filter erkennen Spannungsschwankungen und schalten falls nötig auf Batteriebetrieb um. Die Online-USV versorgt dagegen die Verbraucher ständig mit Strom, so dass bei einer Störung keine Beeinträchtigung auftritt; mit dem Nachteil einer höheren Verlustleistung.

An die zu verwendende USV werden eine Reihe von Anforderungen gestellt, die bei der Wahl des Systems zu berücksichtigen sind. Zum einen muss die Kapazität des Energiespeichers dem Verbrauch der angeschlossenen Endgeräte angepasst sein. Sollen nur kurze Ausfallzeiten überbrückt werden, um beispielsweise ein Rechnersystem ordnungsgemäß herunterzufahren, sind unter Umständen kleine Kapazitäten ausreichend. Längere Zeiten machen dagegen den Einsatz einer Notstromversorgung erforderlich. Zum anderen ist bei der Leistungsbemessung

der Stromversorgung die Nennleistung und der Einschaltstrom der angeschlossenen Verbraucher zu beachten. [SPECOVIOUS, 2010] Die weiteren Anforderungen an die USV richten sich nach den Vorgaben des Einsatzzweckes. Für die Integration in einem Messclient sind

- hohe Effizienz,
- geringe Kosten und kleine Abmessungen,
- niedriger Wartungsaufwand,
- lange Lebensdauer des verwendeten Akkumulators,
- Anschlussmöglichkeit für Photovoltaik-Anlagen und
- leichte Verfügbarkeit im Handel

ausschlaggebend. Darüber hinaus sollte die Ausgangsspannung der USV der Betriebsspannung der Verbraucher entsprechen. Bei abweichender Ausgangsspannung muss vor die Verbraucher noch ein Spannungswandler geschaltet werden. Die damit verbundene Verlustleistung macht den Aufbau aber ineffizienter.

Die USV sollte außerdem erkennen können, ob der Primärstromkreis unterbrochen ist. Besteht die Möglichkeit, die Anlage mit der GPIO-Schnittstelle des Einplatinen-Computers zu verbinden, kann ein Interrupt gesendet und auf den Netzausfall aufmerksam gemacht werden. Ein Programm auf dem Rechner reagiert auf das Signal und sendet eine Nachricht (E-Mail, SMS, XMPP o. a.) an den Nutzer.

Als Pufferspeicher kommen sowohl Akkumulatoren als auch Kondensatoren in Frage.

Batteriespeicher: Systeme zur unterbrechungsfreien Stromversorgung nutzen in der Regel Akkumulatoren als Pufferspeicher. Insbesondere Blei- und Blei-Gel-Akkumulatoren sind aufgrund ihrer niedrigen Kosten verbreitet. Sie weisen eine vergleichsweise geringe Energiedichte von 35 Wh/kg auf und müssen in regelmäßigen Abständen ausgetauscht werden. Im Gegensatz zu anderen Akkumulator-Technologien, wie Lithium-Ionen oder Lithium-Polymer, besteht aber eine geringere Gefahr von thermischem Durchgehen. Die Energiedichte von Lithiumzellen ist mit 120 Wh/kg hingegen höher. Die Entwicklung keramischer Elektrolyte wird in den nächsten Jahren die Energiedichte von Akkumulatoren weiter ansteigen lassen. [SPECOVIOUS, 2010] Da jeder Akkumulatorentyp andere Ladecharakteristiken aufweist, müssen diese von der Ladeschaltung berücksichtigt werden.

Kondensatoren: Zur Überbrückung kurzer Ausfallzeiten eignen sich auch Doppelschichtkondensatoren als Energiespeicher in unterbrechungsfreien Stromversorgungen. Anders als bei Akkumulatoren wird elektrische Energie nicht indirekt in Form von chemischer Energie gespeichert, sondern direkt als elektrisches Feld. Es kann sowohl eine kurze Aufladezeit von wenigen Sekunden, als auch eine hohe Lebensdauer von vielen Jahren ermöglicht werden. Für den Einsatz in USV-Anlagen sind ggf. noch Gleich- und Wechselrichter, Transformatoren und passende Regel- und Steuereinrichtungen erforderlich. [ALLELEIN, 2013] Die Pufferzeit ist abhängig vom Ladezustand der Kondensatoren und von der Stromaufnahme der Verbraucher, beträgt je nach Kapazität aber nur einige Sekunden bis Minuten (Abbildung 6.11).

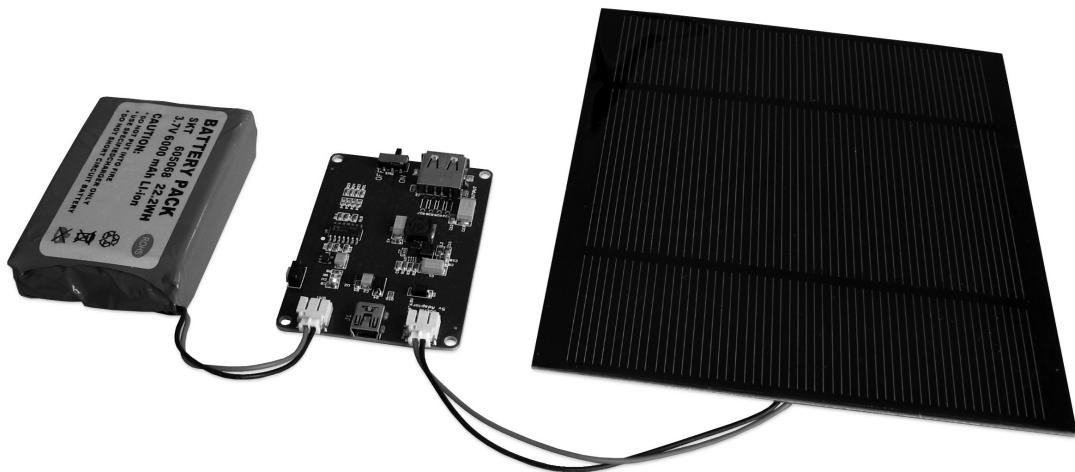


Abbildung 6.12.: USV-Platine „LiPo Rider Pro“ mit Lithium-Polymer-Akkumulator und Solarmodul

Im Folgenden sollen konkrete USV-Anlagen beschrieben und verglichen werden. Sie nutzen Akkumulatoren als Pufferspeicher und besitzen zum Teil eine integrierte Ladeschaltung.

Bausatz: Im Handel sind fertige Bausätze von USV-Anlagen für Verbraucher mit Kleinspannung erhältlich. Sie basieren auf Schaltreglern, die bei einem Netzausfall eine Betriebsspannung von z. B. 12 V bereitstellen, die mit mehreren hundert Milliampere belastbar ist. Energiespeicher sind u. a. konventionelle NiMH-Mignon-Akkumulatoren. Im Netzbetrieb werden diese geladen und im geladenen Zustand gehalten. Tritt ein Spannungsauftreten auf, übernimmt die USV die Versorgung des angeschlossenen Gerätes. Die Kosten für einen Bausatz belaufen sich auf etwa 20 €.

Externer USB-Akkumulator: Im Consumer-Bereich werden diese Akkumulatoren eingesetzt, um mobile Endgeräte, wie Smartphones, Tablet-PCs oder MP3-Player, unterwegs und bei fehlendem Netzanschluss aufzuladen. Über den USB-Port werden sie mit dem Verbraucher verbunden und geben anschließend ihre Ladung an diesen ab. Die Kapazität der Lithium-Ionen- und Lithium-Polymer-Akkumulatoren liegt bei den meisten Systemen zwischen 200 und 20.000 mAh. Die Ausgangsspannung beträgt immer 5 V, bei einem Strom von bis zu 3 A. In einige Modelle lassen sich handelsübliche AA-NiMh-Akkumulatoren einsetzen, die nach Überschreiten ihrer Lebensdauer ausgetauscht werden.

Ein USB-Akkumulator kann auch als einfache und günstige USV dienen, wenn er zwischen Verbraucher und Netzstrom geschaltet wird, und dabei Stromausfälle von einem Tag und länger überbrücken. Qualitativ hochwertige Produkte sind mit einem Schutz gegen Überladung, Überspannung, Tiefentladung und Kurzschluss gesichert. Die Preisspanne der USB-Akkumulatoren erstreckt sich zwischen 10 und 100 €.

LiPo Rider (Pro): Die beiden USV-Platinen LiPo Rider und LiPo Rider Pro werden vom chinesischen Auftragsfertiger Seeed Studio produziert und sind für die Stromversorgung von Einplatinen-Computern konzipiert (Abbildung 6.12). Sie können aber auch andere Endgeräte mit einer Betriebsspannung von 5 V versorgen. Als Pufferspeicher

kommen Lithium-Ionen- und Lithium-Polymer-Akkumulatoren zum Einsatz, die von einem integrierten Schaltkreis geladen werden. Hervorstechend ist die Möglichkeit, Solarzellen anzuschließen. Diese können sowohl das Endgerät versorgen, als auch den Akkumulator laden.

Für den Netzanschluss ist eine Mini-USB-Buchse vorhanden, an die ein Netzgerät mit einer Ausgangsspannung von 5 V angeschlossen wird. Eine USB-Buchse versorgt den angeschlossenen Verbraucher. LiPo Rider und LiPo Rider Pro unterscheiden sich hinsichtlich des Ausgangstroms, der 350 mA bzw. 1.000 mA beträgt. Die Platinen sind für etwa 10 € und 15 € erhältlich.

Mini UPS & Power Sensor: Das Unternehmen AVTECH bietet für die eigenen Überwachungssysteme und andere Anwendungen eine USV-Anlage für Verbraucher mit einer Betriebsspannung von 5 V an. Drei austauschbare Mignonzellen können Strom für maximal 15 min liefern. Ein integriertes Sensor registriert Netzausfälle und aktiviert angeschlossene Überwachungstechnik. Die Kombination kostet umgerechnet 65 €.

UPiS: Das Modul „Uninterruptible Power intelligent Supply“ von Pimodules dient als unterbrechungsfreie Stromversorgung speziell für den Raspberry Pi. Als Pufferspeicher wird je nach Variante ein Lithium-Polymer-Akkumulator mit 1200 oder 2600 mAh verwendet, der eine Laufzeit von 2,2 h bzw. 5 h garantiert. Externe Solarmodule können mit der erweiterten Version genutzt werden. Die Schaltzeit zwischen Netz- und Batteriebetrieb soll laut Hersteller nur 100 µs betragen. Des Weiteren steht dem Raspberry Pi auch eine Echtzeituhr, eine Zeitsteuerung und ein RS-232-Anschluss zur Überwachung von Temperatur und Spannung zur Verfügung. Bei einem Netzausfall fährt eine integrierte Software den Raspberry Pi sicher herunter. Die beiden Varianten der USV-Platine sind gegenwärtig noch nicht erhältlich und sollen noch dieses Jahr durch Distributoren in Griechenland und Großbritannien vertrieben werden.

picoUPS-100/120: Das amerikanische Unternehmen Mini-Box.com vertreibt zwei USV-Platinen für den Einsatz in Fahrzeugen und als Absicherung von Personal Computern. Die Eingangsspannung des picoUPS-100 beträgt 15 bis 18 V, die des picoUPS-120 13 bis 16 V. Pufferspeicher werden in Form von VRLA-Akkumulatoren (12 V) angeschlossen, für die eine Ladeschaltung vorhanden ist. Die Ausgangsspannung beträgt 12 V, bei einem Strom von maximal 10 A. Die Verlustleistung kann aufgrund der verbauten Metall-Oxid-Halbleiter-Feldeffekttransistoren gering gehalten werden. Besitzen die angeschlossenen Verbraucher eine andere Eingangsspannung (z. B. 5 V), ist noch ein Spannungswandler vorzuschalten. Die Platinen sind für umgerechnet etwa 25 € erhältlich.

6.3.3. Photovoltaik

Bei einer kostenintensiven oder nicht vorhandenen Stromversorgung über das öffentliche Netz kann auf Photovoltaikanlagen im Inselbetrieb zurückgegriffen werden. Der elektrische Strom wird dann ausschließlich von Photovoltaikmodulen erzeugt und in speziellen Solarbatterien gespeichert. Bei kleinen Anlagen werden die Verbraucher über einen Laderegler

Leistungsklasse	Strom	Max. Einspeiseleistung	Max. Entnahmleistung
0	0 bis 15 mA	15,4 W	0,44 bis 12,9 W
1	8 bis 13 mA	4,0 W	0,44 bis 3,84 W
2	16 bis 21 mA	7,0 W	3,84 bis 6,49 W
3	25 bis 31 mA	15,4 W	6,49 bis 12,9 W
4	35 bis 45 mA	15,4 W	reserviert

Tabelle 6.5.: Leistungsklassen nach IEEE 802.3af-2003 für Power over Ethernet

mit 12 V Gleichspannung versorgt. Ein Inselsystem besteht aus den Komponenten Solarmodul, Laderegler, Akkumulator und Verbraucher. Ein Wechselrichter erlaubt darüber hinaus auch den Betrieb von Verbrauchern mit 230 V Wechselspannung.

Der Energiegewinn ist von mehreren Faktoren abhängig. Neben dem Wirkungsgrad und der Leistung des Solarmoduls sind auch der Standort und das Strahlungsangebot entscheidend. Die Jahressumme der Globalstrahlung auf eine horizontale Empfängerfläche beträgt nach [BOLLIN, 2013] beispielsweise

- in Sizilien 2.000 kWh/m²,
- in Süddeutschland 1.100 kWh/m²,
- in Norddeutschland 1.000 kWh/m²,
- in Helsinki 700 kWh/m².

Die Tagessumme für den Standort Süddeutschland beträgt

- im Juli 5 kWh/m²,
- im März 3,5 kWh/m²,
- im Januar 1 kWh/m².

Für die Energiespeicherung kommen in der Praxis überwiegend spezielle Blei-Akkumulatoren mit flüssigen Elektrolyten und Blei-Gel-Akkumulatoren zum Einsatz. Sie speichern tagsüber gewonnenen Solarstrom zwischen und versorgen die Verbraucher in der Nacht und bei Schlechtwetterperioden. Der Laderegler schützt den Akkumulator vor Über- und Tiefentladung. Er muss neben den Eigenschaften der verschiedenen Akkumulatortypen auch deren Alterungszustand und Temperatur berücksichtigen.

Die genaue Planung und Zusammenstellung einer Photovoltaik-Inselanlage kann umfangreich sein und hängt von den angeschlossenen Verbrauchern und ihrer Leistungsaufnahme ab. Sollen neben dem Messclient auch Sensoren und weitere Systeme versorgt werden, sind diese in die Kalkulation mit einzubeziehen. Sie können größere Solarmodule und mehrere parallel geschaltete Akkumulatoren notwendig machen. Für die Versorgung des Messclients ohne Peripherie ist hingegen eine kleine Photovoltaik-Anlage ausreichend, die ggf. bereits Teil der unterbrechungsfreien Stromversorgung ist.

6.3.4. Power over Ethernet

Power over Ethernet (PoE) ist eine Technik, mit dem netzwerkfähige Geräte über ein Twisted-Pair-Kabel mit Strom versorgt werden können. Es existieren mehrere Industriestandards, von denen IEEE 802.3af-2003 am weitesten verbreitet ist. Verbraucher müssen für die Nutzung an das Ethernet angeschlossen sein. Der Energieversorger, Power Sourcing Equipment genannt, ermittelt, ob die Endgeräte PoE-kompatibel sind. Diese werden als Powered Device bezeichnet. Um eine Beschädigung von Hardware, die nicht PoE-fähig ist, auszuschließen, wird ein Verfahren zur Messung des Abschlusswiderstandes eingesetzt (Resistive Power Discovery). Der Verbraucher muss dabei dem Energieversorger die eigene Leistungsklasse mitteilen. Das Verfahren kann mit Ethernet, Fast-Ethernet und Gigabit-Ethernet eingesetzt werden. [ZHANG, 2010]

Die Energieversorger sind in der Regel PoE-fähige Hubs oder Switches. Mit sogenannten Injektoren können bestehende Infrastrukturen, die nicht für PoE ausgelegt sind, nachgerüstet werden. Bei Endgeräten ohne PoE kommen Splitter zum Einsatz.

Für IEEE 802.3af-2003 sind fünf Leistungsklassen definiert, mit Entnahmleistungen zwischen 0,44 W und 12,9 W (Tabelle 6.5). Die Versorgungsspannung beträgt 48 V, die maximale Stromaufnahme der Endgeräte 350 mA. Power over Ethernet eignet sich daher nur für Netzwerkgeräte mit einer niedrigen Leistungsaufnahme, wie IP-Telefone, WLAN-Access-Points oder Kameras. Die verfügbare Leistung ist ausreichend, um den Messclient und ggf. angeschlossene Sensoren mit Strom zu versorgen. [METTER U. BUCHER, 2007] Werden Endgeräte im Außenbereich mit PoE betrieben, sollten sie mit einem Schutz gegen Überspannungen, z. B. durch Blitzeinschläge, ausgerüstet sein.

6.4. Gehäuse

Das Gehäuse dient der Aufnahme der einzelnen Komponenten des Messclients und soll vor Witterung und elektromagnetischer Strahlung schützen. Es kommen verschiedenen im Handel erhältliche Gehäuse in Frage, die sich in Bauform und Material unterscheiden. Diese sind häufig gegen das Eindringen von Staub und Wasser geschützt. Der Hersteller klassifiziert in diesem Fall eine Schutzart nach DIN EN 60529 (Tabelle 6.6). Auch aus diesem Grund soll auf eine Eigenentwicklung verzichtet und stattdessen auf bereits verfügbare Lösungen zurückgegriffen werden. Die folgenden Materialien sind verbreitet:

- Kunststoff
 - Acrylnitril-Butadien-Styrol
 - Polyamid
 - Polycarbonat
 - Polyester
- Metall
 - Aluminium
 - Edelstahl
 - Stahlblech

Kunststoffgehäuse bieten keinen Schutz gegen elektromagnetische Strahlung. Um dennoch die elektronischen Bauteile im Inneren abzuschirmen, kann auf Klebefolien aus Kupfer zurückgegriffen werden. Eine weitere Möglichkeit ist die Nutzung eines zweiten Gehäuses aus

1. Ziffer	Schutz gegen Fremdkörper und Berührung	2. Ziffer	Schutz gegen Wasser
0	ungeschützt	0	ungeschützt
1	geschützt gegen feste Fremdkörper > 50 mm	1	geschützt gegen Tropfwasser
2	geschützt gegen feste Fremdkörper > 12,5 mm	2	geschützt gegen Tropfwasser bis 15° Neigung
3	geschützt gegen feste Fremdkörper > 2,5 mm	3	geschützt gegen Sprühwasser
4	geschützt gegen feste Fremdkörper > 1 mm	4	geschützt gegen Spritzwasser
5	geschützt gegen Staub	5	geschützt gegen Strahlwasser
6	dicht gegen Staub	6	geschützt gegen starkes Strahlwasser
		7	geschützt gegen zeitweiliges Untertauchen
		8	geschützt gegen dauerhaftes Untertauchen

Tabelle 6.6.: IP-Schutzarten nach DIN EN 60529

Metall. Je nach Bauart der beiden Gehäuse lassen sich diese durch vorgefertigte Systeme (Steckverbindungen, Hutschienen) zusammenfügen. Die einzelnen Gehäusetypen können wie folgt kategorisiert werden:

19"-Gehäuse dienen dem Einbau von Komponenten in einem Rack. Dabei handelt es sich um genormte Haltevorrichtungen, die 19" (48,26 cm) breite Einschübe für Einbauten besitzen. Entsprechende Gehäuse sind überwiegend aus Stahlblech und Kunststoff gefertigt und auch als reine Baugruppenträger verfügbar. In der Regel werden sie vom Hersteller keiner genormten Schutzart zugewiesen. Sie sind nicht witterungsbeständig und benötigen für den Außeneinsatz ein weiteres Gehäuse.

Handgehäuse sind aufgrund ihrer Größe und Form für den tragbaren Gebrauch ausgelegt und weisen oft ein ergonomisches Design auf, weshalb nur kleine Komponenten integriert werden können. Sie werden aus verschiedenen Kunststoffen hergestellt. Viele Gehäuse weisen einen einfachen Schutz gegen das Eindringen von festen Fremdkörpern, Staub und Spritzwasser auf.

Hutschienengehäuse können auf U-förmigen Tragschienen montiert werden. Diese Profilschienen sind häufig in Schaltschränken und Verteilerkästen zu finden und erlauben eine schnelle und einfache Montage von elektrischen Betriebsmitteln. Hutschienengehäuse sind zwar auch außerhalb eines Installationsgehäuses einsetzbar, sie sind dort aber nur bedingt gegen Wasser und Staub geschützt.

Installationsgehäuse sind meist aus pulverbeschichtetem Stahlblech gefertigt und dienen der Aufnahme von elektrischen Baugruppen. Bei vielen Installationsgehäusen wird der Zugriff auf das Innere durch eine Tür an der Front ermöglicht. Wenn zusätzlich

Tragschienen angebracht werden, lassen sich mehrere Hutschienengehäuse einfach einsetzen. Durch Schraubverbindungen ist auch eine Montage von Universal- und Wandgehäusen möglich. Die Schutzarten sind in der Regel IP55 bis IP69K; eine Verwendung im Außenbereich ist somit möglich. In größeren Installationsgehäusen können auch Mehrfachsteckdosen verbaut und die Netzkupplung der Netzanschlusses ins Innere verlagert werden.

Pultgehäuse besitzen eine geneigt Frontplatte, meist aus Aluminium, die sich als Bedienfeld für Steuer- und Kommandogeräte aller Art gestalten lässt. Sie sind auch in den Schutzarten IP40 und IP54 verfügbar.

Tischgehäuse sind für den Betrieb auf Schreibtischen und Arbeitsflächen ausgelegt und teilweise mit Standfüßen ausgestattet. Die Größe kann an die nach DIN 41494 oder IEC 60297 standardisierten Europakarten angepasst sein. Sie sind auch in den Schutzarten IP40 und IP55 erhältlich.

Universalgehäuse sind nach Typ unterteilte Kunststoff- und Metallgehäuse, zum Teil mit Flansch. Durch die standardisierte Größe sind sie günstig und leicht erhältlich. Die Schutzart liegt, sofern vorhanden, in der Regel zwischen IP40 und IP67.

Wandgehäuse aus Metall und Kunststoff weisen zahlreiche Befestigungspunkte am Gehäuseboden auf, die die Verschraubung an Wänden und Decken erleichtern. Gehäuse aus Polycarbonat können temperaturbeständig, UV-stabilisiert und selbstverlöschend sein. Sie eignen sich auch zur Befestigung in Installationsgehäusen.

7. Software

Der Messclient besteht neben der Hardware auch aus mehreren Software-Komponenten. Dieses Kapitel soll auf diese näher eingehen. Dazu werden verschiedene Betriebssysteme näher betrachtet, die für die Ausführung der Messclient-Software in Frage kommen. Die Messclient-Software erfordert umfangreiche Anpassungen. Daher soll untersucht werden, inwiefern neue technologische Entwicklungen in die Software einfließen können.

7.1. Betriebssystem

Mit dem Aufkommen von Einplatinen-Computern, wie dem BeagleBoard oder dem Raspberry Pi, begann die Portierung unixartiger Betriebssysteme auf diese. Insbesondere die verbreiteten Linux-Distributionen unterstützten die Hardware zeitnah. Diese Entwicklung ist noch nicht abgeschlossen. Einige Betriebssysteme sind nur rudimentär auf diesen Computern lauffähig und können noch nicht alle Hardwarefunktionen nutzen. In diesem Abschnitt werden am Beispiel des Raspberry Pi einige kompatible Betriebssysteme näher beleuchtet und auf ihre Eignung hin untersucht. Die folgenden Betriebssysteme wurden bereits portiert:

- Haiku
- RISC OS
- Unix
 - FreeBSD
 - NetBSD
 - OpenBSD
 - Plan 9 from Bell Labs
 - QNX
- Linux
 - Android
 - Arch Linux ARM
 - Debian
 - Fedora Remix
 - OpenWrt
 - Raspbian
 - Slackware ARM
 - u. a.

Nicht alle von ihnen eignen sich als Grundlage für einen Messclient. Exotische, überholte und nur kommerziell vertriebene Betriebssysteme können bei der Untersuchung unberücksichtigt bleiben.

7.1.1. Linux

Linux ist ein unixoides Open-Source-Betriebssystem, das seit 1991 aktiv entwickelt wird. Die Grundlage des Systems bilden der Linux-Kernel und die Software des GNU-Projektes. Es war ursprünglich nur für *x86*-basierte Personal Computer ausgelegt. Heute unterstützt

Linux mehr Hardware-Plattformen, als jedes andere Betriebssystem. Die Portabilität des Systems führte zu einer starken Verbreitung. Neben Servern und Supercomputern ist Linux auch auf Desktop- und Embedded-PCs sowie mobilen Endgeräten zu finden.

Variationen aus Kernel und weiteren Programmen werden in Form verschiedener Distributionen vertrieben. Die Distributionen sind auf bestimmte Anwendungsgebiete spezialisiert. Einige richten sich an Endanwender und beinhalten daher bereits eine grafische Benutzeroberfläche. Auf Servern und eingebetteten Systemen kann darauf verzichtet werden, weshalb für diese andere Distributionen in Frage kommen. Für die Installation von Dritthersteller-Software kann bei fast allen auf eine Paketverwaltung zurückgegriffen werden. Mit dieser lassen sich vorkompilierte Pakete herunterladen und installieren oder Programme aus ihrem Quelltext kompilieren.

Auf dem Raspberry Pi können nur Linux-Distributionen betrieben werden, die den verbauten ARM-Prozessor unterstützen. Dabei ist zu beachten, dass je nach Portierung nicht alle Befehle des Prozessors verwendet werden. Im Linux-Umfeld wird zwischen zwei ARM-Portierungen unterschieden: *armel* und *armhf*. Während *armel* für langsame Prozessoren ausgelegt ist, die nur die ARMv4-Befehle verstehen, unterstützt *armhf* die schnelle ARMv7-Architektur. Beide unterscheiden sich auch bei Rechenoperationen: *armel* hat keinen Zugriff auf die Gleitkommaeinheit (FPU) des Prozessors, weshalb Berechnungen mit Gleitkommazahlen softwareseitig durchgeführt werden müssen („Soft Float“). Dagegen sind hardwareseitige Berechnungen durch die FPU mit *armhf* zum Teil erheblich schneller („Hard Float“). Einige Programme, wie die Java-Laufzeitumgebung, unterstützen die Binärschnittstelle des Raspberry Pi nicht und sind auf ein „Soft Float“-Betriebssystem angewiesen.

Nachfolgend sind einige Linux-Distributionen aufgelistet, die für den Betrieb auf dem Raspberry Pi in Frage kommen. Sie unterstützen entweder nur „Soft Float“ oder nur „Hard Float“.

Arch Linux richtet sich an fortgeschrittene Linux-Anwender und bietet nur wenige Hilfen bei Installation und Konfiguration. Systemupdates werden als *Rolling Releases* eingespielt, d. h. es gibt keine Betriebssystem-Versionen sondern nur laufende Aktualisierungen. Für die ARMv6-Plattform wird „Hard Float“ unterstützt.

Debian ist eine der populärsten Linux-Distributionen. Mehr als 20.000 Software-Pakete sind für das Betriebssystem verfügbar. Der *armel*-Port setzt auf das langsamere „Soft Float“; die *armhf*-Version funktioniert mit dem ARMv6-Prozessor des Raspberry Pi nicht ohne tiefgreifende Anpassungen.

Fedora wird nur in Form eines inoffiziellen „Remixes“ namens Pidora angeboten. Dieser umfasst die Software des Fedora-Projektes, den Fedora-ARM-Port und optimierte Pakete für den Raspberry Pi. Dadurch ist auch „Hard Float“ gewährleistet.

OpenWrt ist eine kleine Linux-Distribution für Embedded-Geräte. Sie wird häufig auf WLAN-Routern eingesetzt. Der Raspberry Pi wird aber nur rudimentär unterstützt („Soft Float“).

Raspbian ist eine speziell an den Raspberry Pi angepasst Distribution, die auf Debian basiert und „Hard Float“ unterstützt. Es wird von der Raspberry Pi Foundation für den

Betrieb empfohlen, beinhaltet jedoch unnötige Software (grafische Benutzeroberfläche, Webbrowser, Multimedia-Programme u. a.). Eine inoffizielle Server-Edition ohne Drittsoftware ist erhältlich.

7.1.2. FreeBSD

FreeBSD ist ein Unix-artiges Open-Source-Betriebssystem, das aufgrund seiner hohen Betriebs- und Datensicherheit besonders für den Einsatz auf Servern und Appliances geeignet ist. Verwendung findet das System vor allem bei Internet Service Providern, Hochleistungsroutern und Internet-Backbones; es ist aber auch auf Desktop-Computern verbreitet. Mit über 24.000 Applikationen existiert eine große Anzahl von quelloffenen Anwendungen, die entweder als Binärpaket oder über das Ports-System¹ installiert werden können. FreeBSD ist neben NetBSD und OpenBSD das am weitesten verbreitete BSD-Betriebssystem. Die verschiedenen Derivate vereint die BSD-Lizenz, eine Familie von freien Softwarelizenzen, die den Nutzern nur minimale Restriktionen bei der Verteilung von entsprechend lizenzierten Software auferlegen. [ZISLER, 2006]

Im Juli 2012 gab es seitens der Community erste erfolgreiche Versuche, FreeBSD auf dem Raspberry Pi zu betreiben. Diese Versionen waren aber noch nicht vollumfänglich lauffähig. Kurz darauf wurden weitere Komponenten des Raspberry Pi, darunter Interrupt-Controller, Watchdog, System-Zeitgeber und Video-Framebuffer, unterstützt. Die aktuellen Entwicklungen fließen in den *Current* genannten Entwicklungszweig von FreeBSD ein, der nur für Entwickler und Tester geeignet ist und kein stabiles System für den Produktiveinsatz darstellt.

Problematisch sind nicht nur die inoffiziellen und möglicherweise instabilen Betriebssystem-Images, sondern auch das Fehlen geeigneter Binärpakete für die ARMv6-Plattform. Software von Drittanbietern muss daher aus den Ports installiert werden. Die geringe Rechengeschwindigkeit des Raspberry Pi kann zu sehr langen Kompilationszeiten führen.²

Von einem Betrieb auf dem Messclient sollte aus den oben genannten Gründen abgesehen werden, bis ein offiziell unterstützter *Stable*- oder *Release*-Zweig des Betriebssystems veröffentlicht wurde und ein Repository für Binärpakete verfügbar ist. Die unkomplizierte Portierung der Client-Software nach FreeBSD schließt eine spätere Nutzung des Betriebssystems aber nicht aus.

7.1.3. NetBSD

NetBSD ist ein unixoides Betriebssystem der BSD-Familie. Die erste Version wurde 1993 veröffentlicht. Das System unterstützt eine Vielzahl von Hardware-Plattformen und ist sehr

¹ FreeBSD Ports ist ein Paketverwaltungssystem, das einen einfachen und konsistenten Weg zur Installation von Softwarepaketen darstellt. Vor der Installation müssen die Programme häufig erst aus dem Quelltext kompiliert werden.

² Es existieren Werkzeuge, die die Kompilationszeit verkürzen. Die Programme *distcc* und *ccache* ermöglichen die Verteilung der Rechenlast beim Kompilieren auf mehrere Rechner in einem Netzwerk. Mit dem Werkzeug *poudriere* kann ein vollständiger Port-Baum kompiliert werden, der sich anschließend als *pkgng*-Repository exportieren lässt. Ein weiterer Ansatz ist die User-Mode-Emulation der ARM-Architektur auf einem Host-System, wie sie die virtuelle Maschine QEMU erlaubt.

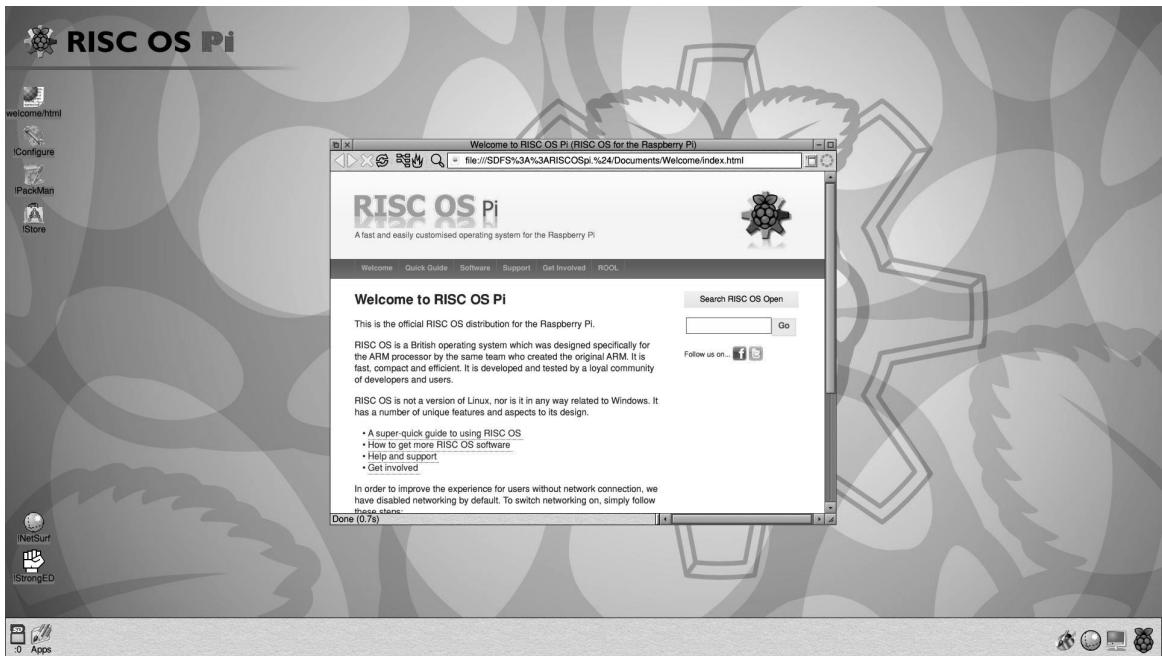


Abbildung 7.1.: Bildschirmfoto der grafischen Benutzeroberfläche von RISC OS Pi

portabel.³ Dieser Umstand beschleunigte die Entwicklung einer Version für den Raspberry Pi. Als Paketverwaltung für die Distribution von Drittsoftware kommt die NetBSD Packages Collection (*pkgsrc*) zum Einsatz. Anwendungen werden vorrangig aus dem Quelltext kompiliert. Binärpakete stehen ebenfalls zur Verfügung.

Mit der Veröffentlichung von NetBSD 6.0 wird auch der Raspberry Pi rudimentär unterstützt. USB und Ethernet funktionieren seit Release 6.1. Es handelt sich aber immer noch um experimentelle Entwickler-Version, die noch nicht für den produktiven Einsatz geeignet sind.

7.1.4. RISC OS

Das britische Unternehmen Acorn Computers Ltd. entwickelte 1987 das RISC OS für die eigenen ARM-basierten Computer. Es ist ein kleines und schnelles Multitasking-Betriebssystem mit grafischer Benutzeroberfläche, das nur auf Computern mit ARM-Chipsatz und speziellen Emulatoren lauffähig ist. Die RISC OS Open Ltd. verwaltet den Quelltext und ermöglicht das kostenfreie Herunterladen.⁴ Obwohl der Quelltext offen geteilt und vertrieben wird, handelt es sich nach Angaben des Distributors nicht um Open Source. Mit RISC OS Pi steht auch eine offizielle Version für den Raspberry Pi zur Verfügung (Abbildung 7.1).

Es besteht aufgrund des Alters des Betriebssystems nur eine geringe Hardwareunterstützung für externe Geräte. Beispielsweise ist der Betrieb von WLAN-Adaptoren wegen fehlender Treiber nicht möglich. In diesem Fall muss auf einen universellen Ethernet-WLAN-Adapter

³NetBSD unterstützt 57 verschiedenen Architekturen, darunter i386, amd64, ARM und MIPS. Eine vollständige Liste ist online einsehbar: <http://netbsd.org/ports/>.

⁴<http://www.riscosopen.org/>

zurückgegriffen werden. Die praktische Bedeutungslosigkeit des Betriebssystems und die veraltete Technologie machen es als Grundlage für einen Low-Cost-Messclient uninteressant.

7.2. Programmiersprache

Die Client- und die Server-Komponente der DABAMOS-Software sind in der Programmiersprache Java geschrieben. Der Client ist eine reine *Java-SE*-Anwendung, während der Server als Enterprise-Applikation konzipiert ist und das Framework *JavaServer Faces* setzt. Die Komponenten weisen konzeptionelle Schwächen auf, die in einem umfangreichen Architektur- und Code-Review-Prozess überarbeitet werden müssen. Dazu zählen vor allem die Reimplementierung der internen und externen Kommunikation sowie Dokumentations- und Refactoring-Aufgaben.

Des Weiteren besteht das Problem, dass sich viele Module des Clients im Server wiederverwendeten lassen und umgekehrt, weil sich die Software-Architekturen unterscheiden. Dadurch erhöht sich der Entwicklungsaufwand. Außerdem reduziert sich die Geschwindigkeit von Rechenoperationen mit Gleitkommazahlen auf dem Einplatinen-Computer, weil die Java-Laufzeitumgebung für die ARM-Architektur nur die „Soft Float“-Binärschnittstelle unterstützt.

Die Alternative zur Überarbeitung der Software ist eine Neuimplementierung in einer anderen Programmiersprache, die diese Nachteile nicht aufweist. Interessant ist diesem Zusammenhang insbesondere die Programmiersprache Go. Die Entwicklung der Sprache begann im Jahr 2007, als drei Mitarbeiter von Google in Teilzeit an einem Entwurf arbeiteten. Im November 2009 erschien die erste Version unter der BSD-Lizenz. Die Programmiersprache ist damit Open-Source. Nachdem viele Beiträge weiterer Entwickler einflossen, konnte im März 2012 die erste stabile Version veröffentlicht werden. Ein Jahr später erschien Go 1.1 mit zahlreichen Verbesserungen, insbesondere hinsichtlich der Performance.

Die Programmiersprache ist unter Windows, Linux, Mac OS X und FreeBSD nutzbar und eignet sich grundsätzlich für die Sensoransteuerung. Auch die Server-Komponente kann in Go geschrieben werden, wodurch sich einzelne Module wiederverwenden lassen und die Entwicklungsgeschwindigkeit erhöht wird. Darüber hinaus lassen sich Go-Anwendungen auch innerhalb von Cloud-Plattformen ausführen. Die Verbreitung der Sprache ist noch beschränkt. Auf dem „TIOBE Programming Community Index“ belegte sie im Juli 2013 den Platz 42. [TIOBE SOFTWARE BV, 2013] Ungeachtet dessen wird Go bereits in einer Vielzahl von Projekten eingesetzt, u. a. bei der Datenbankskalierung von YouTube und den HTTP-Download-Servern von Google.

7.3. Datenbank

Die Messclient-Software muss die von den Sensoren gesammelten Daten vor der Übertragung auf den Server zeitweise oder bei einem rein lokalen Betrieb dauerhaft speichern. Die Verwaltung der Daten kann über Textdateien oder Datenbanken realisiert werden. Textdateien sind die einfachste Möglichkeit der Speicherung. Auf sie kann direkt zugegriffen werden

und hinzukommende Daten lassen sich an das Ende der Datei anhängen. Die Verarbeitung von in Textdateien gespeicherten Daten ist hingegen aufwendig. Die fehlende Indizierung der Datensätze führt zu einem langsamen Zugriff mit linearer Zeitkomplexität.

Die Speicherung der Messdaten in einer Datenbank ist nicht nur aus Gründen der Indizierung und Anfrageoptimierung vorzuziehen. Auch in Hinblick auf Datensicherheit, Datenintegrität und einen möglichen Mehrbenutzerbetrieb sind Datenbanken im Vorteil. Beim Einsatz in eingebetteten Umgebungen ist aber der im Vergleich zu Textdateien erhöhte Ressourcenverbrauch zu beachten. Es existieren verschiedene Formen von Datenbanksystemen, die sich in dem zugrunde liegenden Datenbankmodell unterscheiden. Verbreitet sind relationale, objektorientierte und dokumentenorientierte Datenbankmodelle.

Die erste Version der DABAMOS-Software nutzt die objektorientierte NoSQL-Datenbank *database for objects (db4o)* des Unternehmens Versant für die Speicherung der Messdaten auf dem Client und dem Server. Diese Datenbank eignet sich für eingebettete Anwendungen, weil der Speicherbedarf nur gering und eine Abbildung objektorientierter Daten in relationale nicht erforderlich ist. Die Nutzung ist aber auf die Java- und die .NET-Plattform beschränkt. Für die Programmiersprache Go existiert keine Schnittstelle, um *db4o*-Datenbanken anzusprechen.

Zur Familie der NoSQL-Datenbanken gehören auch Key/Value-, Dokument-, Wide-Column- und Graphendatenbanken. Für die Speicherung von Messdaten sind insbesondere Dokumentendatenbanken interessant. Sie eignen sich für unstrukturierte und semi-strukturierte Daten, die ohne festes Schema unter einem Schlüssel gespeichert werden. Dabei kann, anders als bei relationalen Datenbanken, jeder Eintrag seine eigene Struktur haben. Messwerte verschiedener Sensoren lassen sich so zusammen in einer Kollektion speichern. In relationalen Datenbanken sind diese hingegen auf mehrere Tabellen zu verteilen, weil die Struktur der Daten nicht identisch ist. Die Messwerte von Neigungssensoren habe eine andere Struktur als Messwerte von Tachymeter oder GNSS-Empfängern. Als Datenformat wird je nach Datenbank u. a. XML, JSON oder YAML verwendet.

Nach [DB-ENGINES, 2013] ist MongoDB die am weitesten verbreitete NoSQL-Datenbank. Die Daten werden schemilos in UTF-8-kodierten Dokumenten in einem JSON-ähnlichen Format gespeichert. Eine Datenbank besteht aus Kollektionen, die wiederum mehrere Dokumente zusammenfassen. Kollektionen sind mit den Tabellen relationaler Datenbanken vergleichbar, mit dem Unterschied, dass sie schemafrei sind und beliebige Daten enthalten können. Datenbanken müssen vor Verwendung nicht angelegt werden. Bei der ersten Verwendung werden Datenbank und Kollektionen automatisch erzeugt. [HOLLOSI, 2012] Für die Programmiersprache Go ist ein inoffizieller Treiber verfügbar.

8. Praktische Umsetzung

Für Entwicklung des Messclients lässt sich in zwei Arbeitsbereiche aufteilen. Zum einen die Zusammenstellung und Montage der Komponenten, die zusammen die Hardware des Systems bilden und zum anderen die Neuimplementierung der Messclient-Software, die vorerst nur Sensoren ansteuern und Messwerte in einer Datenbank speichern soll. Die dabei anfallenden Arbeitsschritte sollen in diesem Kapitel ausgeführt werden.

8.1. Hardware

Die Hardware-Entwicklung befasst sich mit der Integration des Einplatinen-Computers, der Kommunikation mit Sensoren und Netzwerken, der Stromversorgung und dem Einbau der Komponenten in ein geeignetes Gehäuse. Konzeptionelle Überlegungen und auftretende Fragen sollen nachfolgend dokumentiert werden.

8.1.1. Einplatinen-Computer

Die Wahl eines geeigneten Einplatinen-Computers entscheidet maßgeblich über den Funktionsumfang des Messclients und beeinflusst die Zusammensetzung und die Anforderungen der anderen Komponenten. Nicht alle untersuchten Systeme erfüllen die notwendigen und hinreichenden Bedingungen, um als Grundlage für einen Low-Cost-Messclient in Frage zu kommen.

Bei der Umsetzung eines ersten Messclient-Prototypen soll der Raspberry Pi (Modell B) zum Einsatz kommen und genauer untersucht werden. Die Punkte, die ihn für den Einsatz auszeichnen sind die Rechengeschwindigkeit und die Speichergröße, die vielen kompatiblen Betriebssysteme, die Möglichkeit, ihn mit Erweiterungsplatinen auszurüsten sowie die Kamera- und die Display-Schnittstelle. Der niedrige Preis und die leichte Verfügbarkeit im Handel sind ebenfalls entscheidend. Als Nachteil ist lediglich die fehlende Echtzeituhr zu nennen, die nachgerüstet oder durch NTP ersetzt werden muss. Den fehlenden Netzschalter, der ein ACPI-Event senden kann, um das System herunterzufahren, hat der Raspberry Pi mit anderen Einplatinen-Computern gemein.

8.1.2. Kommunikation

Das Modell B des Raspberry Pi ist mit einem integrierten USB-Hub, der zwei USB-Ports bereitstellt, und einer Ethernet-Schnittstelle ausgestattet. Über die USB-Ports können Adapter für verschiedene Kommunikationsstandards nachgerüstet werden. Sensoren lassen sich



Abbildung 8.1.: Einfacher WLAN-Adapter mit fester Antenne für den USB-Port

direkt oder über ein Konverterkabel mit dem Einplatinen-Computer verbinden. Für die Ansteuerung eines Leica Nivel210 kommt ein solches Konverterkabel mit „Prolific PL-2303 HXD“-Chipsatz zum Einsatz. Die Tachymeter Leica TM30 und TS30 mit 8-poligen LEMO-Rundsteckverbindern haben dagegen bereits eine Datenleitung für USB und erfordern keinen Konverter mehr.

Die Ethernet-Schnittstelle wird vorrangig für Netzwerkverbindungen zu einem Router genutzt, über den auch ein Internet-Zugang besteht. Ein WLAN-Stick (802.11b/g/n) mit Antenne und RTL8192CU-Chipsatz steht für Versuche ebenfalls zur Verfügung (Abbildung 8.1). Ein passender Linux-Treiber ist bei Debian „Wheezy“ bereits integriert, weshalb der Adapter nach dem Einstecken sofort nutzbar ist. Eine WLAN-Verbindung zu einem Router lässt sich erfolgreich aufbauen. Für weitere Versuche mit Netzwerkanbindung wird aber vorerst nur die Ethernet-Schnittstelle verwendet.

8.1.3. Stromversorgung

Für praktische Versuche wird ein USB-Power-Set von Voltcraft eingesetzt, das, anders als Steckernetzteile, mit einem Netzkabel ausgestattet ist. Das Netzteil kann somit auch in einem wetterfesten Gehäuse, z. B. einem Schaltschrank, untergebracht werden.

Als unterbrechungsfreie Stromversorgung kommt die in Abschnitt 6.3.2 beschriebene Platine „LiPo Rider Pro“ zum Einsatz. Sie zeichnet sich durch niedrige Kosten, geringe Abmessungen, eine Ausgangsspannung von 5 V sowie Anschlussmöglichkeiten für einen Lithium-Akkumulator und ein Solar-Panel aus.

Als Batteriespeicher ist ein Lithium-Akkumulator mit 3,7 V und 6.000 mAh an die USV-Platine angeschlossen. Der Raspberry Pi (Modell B) benötigt zum Betrieb eine Spannung von 5 Volt bei einem Strom von 700 mA. Die Leistungsaufnahme beträgt demnach 3,5 W. Der Akkumulator kann den Einplatinen-Computer theoretisch für etwa 8:30 h mit Strom versorgen. Durch den Verlust bei der Spannungsumwandlung und die abnehmende Spannung des Akkumulators mit fortschreitender Entladung ist die tatsächliche Laufzeit kürzer. Sollen mit dem Akkumulator zusätzliche Geräte mit Strom versorgt werden, z. B. A/D-Umsetzer, WLAN- oder Bluetooth-Adapter, reduziert sich die Laufzeit weiter.

Um genaue Angaben zur Laufzeit machen zu können, ist eine experimentelle Bestimmung notwendig. Für diese wird ein Shell-Skript genutzt, das im Minutentakt Datum und Uhrzeit ausgibt und die Werte an eine Textdatei anhängt (Listing 8.1). Nach fortlaufender Entleerung des Akkumulators wird die Ausgangsspannung der USV-Platine soweit absinken, dass sich der Raspberry Pi zu einem bestimmten Zeitpunkt ausschaltet. Die maximale Laufzeit bei Verwendung des getesteten Akkumulators berechnet sich aus der Differenz zwischen erster und letzter Zeitangabe in der Textdatei. Bei dem durchgeföhrten Experiment konnte eine Laufzeit von 14:18 h erreicht werden. Eine Erklärung für die Differenz zwischen

theoretischer und tatsächlicher Laufzeit ist eine geringere Leistungsaufnahme, die unter den angenommenen 3,5 W liegt.

Die Solarzelle hat eine Abmessung von 138×160 mm und liefert eine Spitzenleistung von 3 W bei 5,5 V und 540 mA. Der Wirkungsgrad beträgt lediglich 16 %. Das vollständige Laden eines Lithium-Akkumulators mit 6.000 mAh dauert mit dieser Zelle min. 11:06 h und ist abhängig von Sonneneinstrahlung, Temperatur und tatsächlicher Leistung der Solarzelle. Die vom Hersteller angegebene Leistung reicht nicht zum autonomen Betrieb des Raspberry Pi aus. Daher kann sie nur zum Laden des Akkumulators eingesetzt werden. Weitergehende Untersuchungen hinsichtlich der praktischen Einsatzmöglichkeiten werden im Rahmen dieser Arbeit nicht durchgeführt.

Listing 8.1: Ein einfaches Shell-Skript zur Bestimmung der Laufzeit des Raspberry Pi bei Nutzung eines Akkumulators

```
1 #!/bin/sh
2 while true; do
3     echo `date` && echo `date` >> date
4     sleep 60
5 done
```

8.1.4. Gehäuse

Die einzelnen Komponenten des Messclients, darunter Einplatinen-Computer, USV-Platine und Akkumulator, sollen in einem einzelnen Kunststoffgehäuse befestigt werden (Abbildung 8.2). Das Gehäuse weist nur einen einfachen Schutz gegen Wasser und Staub auf und ist für den Einsatz im Innenbereich konzipiert. Im Außenbereich bietet ein zusätzliches Schaltschrankgehäuse aus Metall in IP68 Schutz vor Witterung und elektromagnetischer Strahlung. Diese Aufteilung ermöglicht eine kostengünstige Herstellung einer Grundversion und eine spätere Anpassung an individuelle Anforderungen.

Der erste Prototyp basiert auf einem Wandgehäuse aus ABS mit den Abmessungen $201 \times 163 \times 98$ mm. Die angegebene Schutzart ist IP65. Für die Befestigung von Bauteilen sind vorgebohrte Löcher im Inneren vorhanden. Eine rechteckige Leiterplatte dient als Träger für die Komponenten. Diese passt nicht ohne Weiteres in das Gehäuse und muss gefräst werden. Nach dem Ausmessen des Gehäuses erfolgt die Konstruktion in *AutoCAD* und der automatisierte Zuschnitt mit der CNC-Fräse. Die Fräse markiert bereits alle späteren Bohrlöcher, die anschließend mit der Säulenbohrmaschine gesetzt werden. Zylinderstifte fixieren den Einplatinen-Computer und die USV-Platine auf der Trägerplatte, selbstklebende Klettverschlüsse dienen der Befestigung des Akkumulators.

Ethernet- und USB-Anschluss des Raspberry Pi sollen auch von außen nutzbar sein. Im Handel erhältliche RJ45- und USB-Einbaubuchsen in IP68 bzw. IP67 werden dazu in vorgebohrte Löcher des Gehäuses gesteckt und verschraubt. Ein USB-Verlängerungskabel verbindet den USB-Anschluss des Raspberry Pi mit der Einbaubuchse. Die RJ45-Buchse hat dagegen nur Ausgänge für die einzelnen Adern eines Twisted-Pair-Kabels. Das Kabel muss daher entweder gelötet oder mit Steckverbindern angeschlossen werden.



Abbildung 8.2.: Erster Prototyp des Messclients im Kunststoffgehäuse

8.2. Software

Der grundsätzliche Aufbau und die Funktionsweise der Messclient-Software orientiert sich an den vorhergehenden Arbeiten [SCHWEIMLER, 2010], [WOLFF, 2010] und [HAMANN, 2010] zum Thema Datenbank-orientiertes Monitoring- und Analyse-System. Die dort ausgeführten Konzepte wurden größtenteils übernommen, weshalb an dieser Stelle auf eine Darstellung der Software-Architektur verzichtet wird.

Der Einsatz der Programmiersprache Go soll bei der Entwicklung der Messclient-Software evaluiert werden. Neben der Sensoransteuerung und dem Datenbankzugriff ist mittelfristig auch eine webbasierte grafische Benutzeroberfläche und eine Programmierschnittstelle für den externen Datenzugriff zu realisieren. Von Interesse ist auch die Frage der Eignung von MongoDB als Datenbank-Backend.

8.2.1. Betriebssystem

Bevor das Betriebssystem und die zur Ansteuerung der Sensoren entwickelte Messclient-Software auf dem Raspberry Pi installiert werden kann, sind die folgenden Komponenten zu beschaffen.

- Ein kabelgebundenes Netzwerk, an das der Raspberry Pi mit einem Ethernet-Kabel angeschlossen wird. Das Netzwerk kann durch einen einfachen Router aufgebaut werden, der die Zuweisung von Netzwerkkonfigurationen mittels DHCP unterstützt.
- Ein Host-Computer, der an das Netzwerk angeschlossen ist. Mit ihm wird das Betriebssystem-Abbild für den Raspberry Pi auf die SD-Karte kopiert und die anschließende

Konfiguration über eine SSH-Verbindung durchgeführt. Der Host muss dabei folgende Hard- und Software besitzen:

- Ein Programm, um Dateien von einem Webserver herunterzuladen, z. B. ein Webbrowser oder das Werkzeug *wget*.
- Ein Programm zum Schreiben von Abbildern auf SD-Karten. Unter unixartigen Betriebssystemen kann *dd* benutzt werden. Für Windows existiert das Programm *Win32DiskImager*, für Mac OS X das Programm *PiWriter*. Alternativ erlaubt das Werkzeug *NOOBS* das schnelle Kopieren und Wiederherstellen von Betriebssystem-Abbildern für den Raspberry Pi.
- Ein beliebiger Terminal-Emulator und ein SSH-Client, um sich über das Netzwerk mit dem Raspberry Pi zu verbinden.
- Ein SD-Karten-Lesegerät, mit dem das Abbild auf die leere SD-Karte kopiert werden kann.
- Ein Monitor, der direkt oder mittels Adapter mit dem HDMI-Anschluss des Raspberry Pi verbunden ist.
- Eine USB-Tastatur, die an den USB-Port des Raspberry Pi angeschlossen wird.
- Eine SD-Karte, auf die das Betriebssystem kopiert wird. Eine Kapazität von 4 GB ist für erste Versuche ausreichend. Fallen große Datenmengen an, ist eine entsprechend dimensionierte Karte zu wählen. Zu beachten ist, dass nicht alle SD- und SDHC-Karten mit dem Raspberry Pi kompatibel sind. Hochgeschwindigkeitskarten können für den Bus zu schnell sein. Eine Liste kompatibler Karten ist online verfügbar.¹ Spezielle *Industrial Grade SD Cards* sind für die im Rahmen dieser Arbeit durchgeführten Messungen nicht zwingend erforderlich. Diese sehr robusten aber auch teureren Karten sind für den Einsatz unter extremen Bedingungen ausgelegt und weisen eine definierte MTBF² sowie garantierte Schreib- und Löschzyklen auf. In einer Produktivumgebung sind sie herkömmlichen SD- und SDHC-Karten vorzuziehen.
- Ein 5-Volt-Steckernetzteil für USB-Geräte, das einen Strom von mindestens 700 mA liefern kann.

Der Host-Computer ist ein PC mit FreeBSD 9. Standardmäßig werden alle Befehle auf dem Raspberry Pi – sofern nicht anders vermerkt – mit *Root*-Rechten ausgeführt.

Kopieren des Abbildes

Ein vorkonfiguriertes Abbild von Debian „Wheezy“ wird auf der offiziellen Raspberry-Pi-Webseite angeboten.³ Um Fehler auszuschließen, wird die Integrität des Abbildes vor dem Kopieren auf die SD-Karte geprüft:

```
$ sha1 2013-05-29-wheezy-armel.img
```

¹http://elinux.org/RPi_SD_cards

²Abkürzung für *Mean Time Between Failures*: Erwartungswert der Betriebsdauer zwischen zwei aufeinanderfolgenden Ausfällen.

³<http://www.raspberrypi.org/downloads>

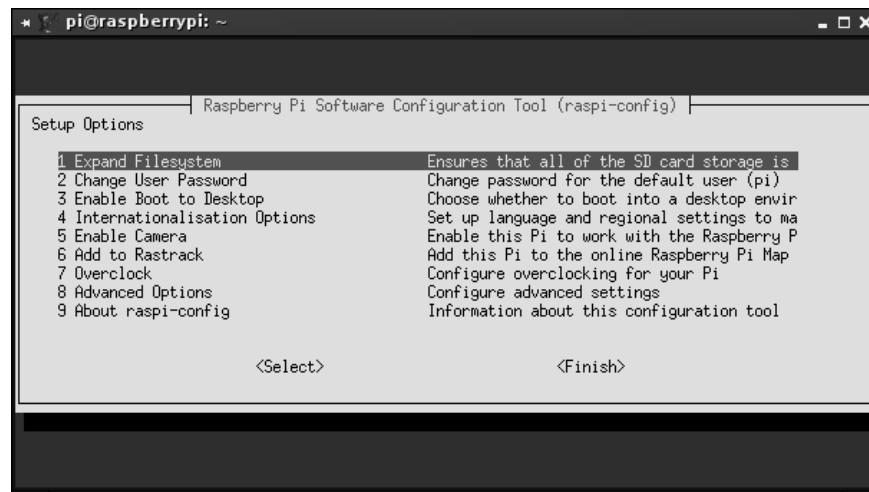


Abbildung 8.3.: Das Werkzeug *raspi-config* liegt in einigen Linux-Distributionen bei und vereinfacht die Konfiguration des Betriebssystems

Der generierte Hash muss mit jenem auf der Webseite übereinstimmen; andernfalls ist das Abbild fehlerhaft. Unter unixartigen Host-Betriebssystemen lässt sich die Image-Datei mit dem Programm *dd* blockweise auf die SD-Karte schreiben:

```
$ dd if=2013-05-29-wheezy-armel.img of=/dev/da0 bs=1M
```

Nach Abschluss des Kopievorgangs kann der Raspberry Pi mit der SD-Karte in Betrieb genommen werden.

Konfiguration

Debian „Wheezy“ für den Raspberry Pi ist bereits vorkonfiguriert. Änderungen an den Einstellungen lassen sich mit dem Programm *raspi-config*, das dem Betriebssystem beiligt, vornehmen (Abbildung 8.3).

Dateisystem vergrößern: Das Abbild ist für SD-Karten mit einer Größe von 2 GB ausgelegt. Werden Karten mit größerer Kapazität verwendet, bleibt der restliche Speicher ungenutzt. Die Funktion „Expand Filesystem“ dehnt das Dateisystem auf die Größe der verwendeten Karte aus.

Lokalisierung: Debian ist auf die englische Lokalisierung voreingestellt. Eine Anpassung an den deutschen Sprachraum ist manuell mit der Funktion „Internationalisation Options“ durchzuführen und umfasst Sprache, Tastatur-Layout und Zeitzone.

Passwort: Das voreingestellte Passwort „raspberry“ sollte aus Sicherheitsgründen mit der Funktion „Change User Password“ geändert werden.

Fernzugriff mit SSH: Mit SSH kann sich ein Benutzer, ähnlich wie mit Telnet, auf einem entfernen System über ein Netzwerk anmelden. Dazu wird eine verschlüsselte Verbindung zwischen zwei Rechnern aufgebaut. Der Raspberry Pi lässt sich dadurch nach erfolgreicher Authentifizierung vom Host aus fernsteuern. Die Authentifizierung kann durch Passwort und/oder Zertifikat erfolgen.

Der SSH-Dienst des Betriebssystems ist aus Sicherheitsgründen standardmäßig deaktiviert. Bevor der Dienst gestartet wird, sollten die SSH-Schlüssel, die dem Abbild bereits beiliegen, gelöscht und durch neu generierte ersetzt werden, da die alten Schlüsse ein Sicherheitsrisiko darstellen (*Man-in-the-middle-Angriff*).

```
# rm /etc/ssh/ssh_host_*key
# dpkg-reconfigure openssh-server
```

Der SSH-Dienst wird mit *raspi-config* aktiviert. Im Anschluss kann mit einem SSH-Client vom Host aus auf den Raspberry Pi zugegriffen werden:

```
$ ssh pi@192.168.0.101
```

Die IP-Adresse des Raspberry Pi wurde per DHCP zugewiesen und variiert unter Umständen. Nach Eingabe des Passwortes besteht ein Fernzugriff. Der verschlüsselte Datenaustausch zwischen beiden Rechnern ist mit SCP⁴ und SFTP⁵ möglich, die beide auf SSH basieren.

Aktualisierung des Systems und der Firmware

Das installierte Betriebssystem-Abbild von Debian „Wheezy“ ist nach der Installation auf dem Stand vom 29. Mai 2013 und sollte aktualisiert werden. Unter Debian lassen sich Paket- und Kernel-Updates mit Hilfe der integrierten Paketverwaltung einspielen:

```
# apt-get update
# apt-get upgrade
# apt-get dist-upgrade
```

Im Anschluss wird die Firmware des Raspberry Pi ebenfalls aktualisiert. Firmware-Updates beseitigen Fehler und stellen neue Funktionen bereit. Verwendet wird das Werkzeug *rpi-update*, das bereits vorinstalliert ist:

```
# rpi-update
```

Nach einem Neustart ist die neue Firmware aktiv.

8.2.2. Messclient-Software

Die in Go geschriebene Ansteuerungs-Software für den Messclient muss vor der Ausführung kompiliert werden. Der dazu notwendige Compiler ist sowohl auf der Entwicklungsplattform (*amd64*) als auch auf der Zielplattform (*armel*) zu installieren, weil eine plattform- und betriebssystemübergreifende Ausführung nicht möglich ist. Während der Compiler für die Entwicklungsplattform aus der Paketverwaltung des Betriebssystems installiert werden kann, ist er auf der Zielplattform hingegen direkt aus dem Quelltext zu kompilieren. Eine Virtualisierung der Zielplattform auf der Entwicklungsplattform kann die Entwicklungszeit verkürzen. Dieser Vorgehensweise erfordert aber noch weitergehende Untersuchungen.

⁴Secure Copy

⁵SSH File Transfer Protocol

Installation von Go

Die Client-Software ist in der Programmiersprache Go geschrieben und ist vor der Ausführung für die ARM-Architektur zu kompilieren. Über die offiziellen Paket-Quellen von Debian „Wheezy“ für *armel* wird kein Go-Compiler bereitgestellt. Daher muss auf inoffizielle Paket-Quellen zurückgegriffen oder der Compiler aus dem Quelltext übersetzt werden. Die Installation aus dem Quelltext gestaltet sich einfach. Lediglich die Versionverwaltungssysteme *Git* und *Mercurial* sind zuvor zu installieren. Danach kann das Go-Repository lokal geklont werden:

```
$ hg clone -u top https://code.google.com/p/go
```

Die Kompilation erfolgt automatisiert durch ein Bash-Skript:

```
# cd go/src && ./all.bash
```

Abschließend sind noch die Umgebungsvariablen zu setzen:

```
export GOARCH=arm
export GOARM=5
export GOOS=linux
export GOPATH=$HOME/go
export GOROOT=/usr/local/go
export PATH=$PATH:$GOROOT/bin
```

Danach steht der Go-Compiler für die Nutzung bereit:

```
$ go version
go version devel +84917868ef43 Wed Jul 03 10:43:24 2013 -0700
linux/arm
```

Installation von MongoDB

Die Client-Software speichert die gesammelten Sensordaten in einer MongoDB-Datenbank. Die NoSQL-Datenbank ist nicht in den offiziellen Paketquellen enthalten und muss für ARM ebenfalls kompiliert werden. Zwei Versionen sind für den Raspberry Pi verfügbar: *mongo-nonx86* und *mongopi*. Die erste ist eine Portierung auf x86-fremde Plattformen, wie SPARC, PowerPC oder ARM, während die zweite auf den Raspberry Pi zugeschnitten ist.

Beide Versionen werden durch ein Klonen des Git-Repositorys heruntergeladen und anschließend übersetzt. Die Installation von *mongo-nonx86* erfordert folgende Schritte:

```
$ git clone https://github.com/skrabban/mongo-nonx86
$ cd mongo-nonx86
$ scons
$ sudo scons --prefix=/opt/mongo install
```

Nach Abschluss muss der Anwendungspfad zur Pfadvariable des Systems hinzugefügt werden. Der Dienst startet nach einem Aufruf von:

```
$ mongod --dbpath $HOME/db
```

Der übergebene Parameter gibt den Ort der Datenbank an.



Abbildung 8.4.: Neigungssensor Leica Nivel210

8.3. Sensoransteuerung

Die Messclient-Software soll für erste Versuche zwei geodätische Sensoren ansteuern. Der Neigungssensor Nivel210 und das Tachymeter TM30 von Leica Geosystems eignen sich aufgrund ihrer offenen GeoCOM-Schnittstelle, die eine Interaktion mit den Systemen vereinfacht. Die Ansteuerung weiterer Sensoren, u. a. von anderen Herstellern und aus dem Low-Cost-Bereich, ist eine Aufgabe für weitere Arbeiten auf diesem Gebiet. Um die Umsetzung dieser Aufgabe zu vereinfachen, soll ein virtueller Sensor implementiert werden, der Sensoren jeder Art nachbilden und emulieren kann.

8.3.1. Neigungssensor

Das Nivel210 von Leica Geosystems ist ein hochpräziser Zweiachs-Neigungssensor, der gleichzeitig Neigung, Neigungsrichtung und Temperatur messen kann (Abbildung 8.4). Er basiert auf dem optoelektronischen Prinzip, bei dem die Richtungen mit einem Flüssigkeitshorizont gemessen werden. Die Auflösung der Neigungsmessung beträgt 0,001 mrad, die der Temperatur 0,1 °C. Ein Messzyklus dauert 300 ms. Das Nivel210 ist mit einer RS-232-Schnittstelle ausgestattet, die der Kommunikation mit einem angeschlossenen Computer dient. Mit dem Nivel220 wird ein Sensor mit den gleichen messtechnischen Spezifikationen angeboten, der aber im Unterschied mit einer RS-485-Schnittstelle ausgestattet ist. Mehrere Nivel220 lassen sich so an einen gemeinsamen Bus betreiben. Beide Typen werden im Rahmen von Überwachungsmessungen eingesetzt [RESNIK, 2013].

Die Ansteuerung des Nivel210 soll in die Messclient-Software integriert werden. Die Schnittstelle und die Steuerbefehle sind gut dokumentiert und für das Abfragen der Messwerte ist im einfachsten Fall nur ein Befehl nötig. Das Nivel210 besitzt zwei 5-polige „LEMO 0“-Buchsen für die Stromversorgung und den Datenaustausch. Das Datenkabel wird an die

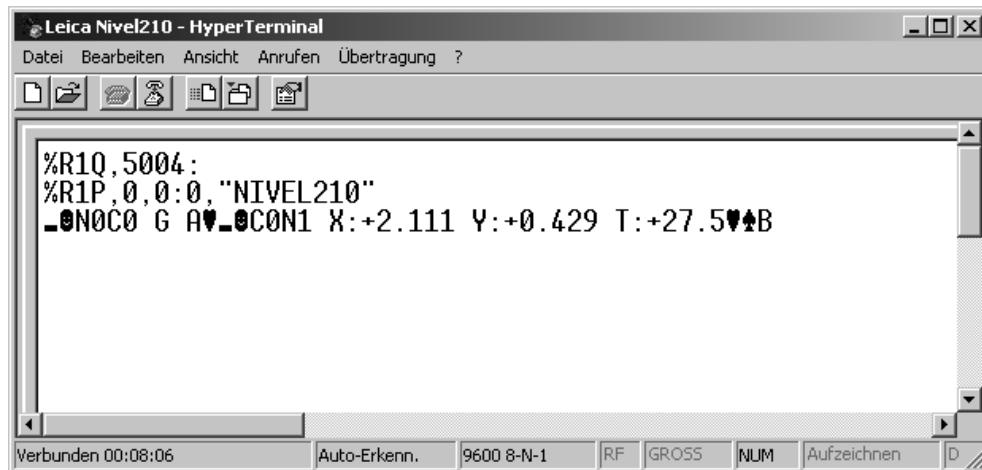


Abbildung 8.5.: Das Programm *HyperTerminal* kann für die einfache Kommunikation mit dem Leica Nivel210 verwendet werden. Die nicht druckbaren Steuerzeichen werden durch Symbole dargestellt.

RS-232-Schnittstelle des Computers angeschlossen, der darüber ASCII-Befehle an den Sensor schickt. Mit einem passenden Konverter ist auch die Nutzung der USB-Schnittstelle möglich.

Serielle Kommunikation mit HyperTerminal

Für eine experimentelle Ansteuerung wird das Nivel210 mit einem Konverter-Kabel an die USB-Schnittstelle des Host-Computers angeschlossen. Das Betriebssystem ist ein virtualisiertes Microsoft Windows XP unter *VirtualBox OSE*, mit FreeBSD 9 als Host-System. Für das Adapterkabel mit dem Chipsatz „Prolific PL-2303 HXD“ wird zunächst ein passender Treiber installiert. Anschließend lässt sich mit dem Systemprogramm *HyperTerminal* eine Verbindung zum Sensor aufbauen (Abbildung 8.5). Das Programm dient u. a. der Ansteuerung externer Geräte über die serielle Schnittstelle. Alternativ ist aber auch die Nutzung anderer Terminal-Programme möglich, z. B. *CoolTerm*, *Tera Term* oder *PuTTY*.

Für die Konfiguration sind nur wenige Einstellungen vorzunehmen: lediglich COM-Port, Baud-Rate, Parität sowie Daten- und Stoppbits (9600 Baud, 8N1). Nach einem erfolgreichen Verbindungsauftbau lassen sich Befehle an das Nivel210 senden. Bei aktiviertem lokalen Echo erscheinen diese auch in *HyperTerminal*. Der generische GeoCOM-Befehl **%R1Q,5004:** fordert den Sensor auf, sich mit seinem Namen zu identifizieren. Die Antwort lautet in diesem Fall:

%R1P,0,0:0"NIVEL210"

Alle Nivel-spezifischen Befehle sind nach dem gleichen Schema aufgebaut und zu Blöcken zusammengefasst:

Block-Start – Empfänger – Absender – Information – Block-Ende

Jeder Teil eines Blocks besteht aus mehreren ASCII-Zeichen. Der genaue Aufbau ist in der technischen Referenz dokumentiert [LEI, 2006]. Block-Start und -Ende bestehen aus

```
Willkommen zu minicom 2.6.1
Optionen: I18n
Übersetzt am Feb 11 2012, 20:50:27.
Port /dev/ttyUSB0

Drücken Sie CTRL-A Z für Hilfe zu speziellen Tasten
XR1Q,5004;
XR1P,0,0;0,"NIVEL210"

NOCO G ACOM1 X:+0.423 Y:+3.000 T:+24.57
```

CTRL-A Z = Hilfe | 9600 8N1 | NOR | Minicom 2.6.1 | VT102 | Offline

Abbildung 8.6.: Das Werkzeug *minicom* kann unter unixartigen Betriebssystemen für die einfache Kommunikation über die serielle Schnittstelle eingesetzt werden

ASCII-Steuerzeichen. Die Adressen von Empfänger und Sender werden im Format NxCx übermittelt, wobei das x in Nx die Nummer des Sensors darstellt. Der Befehl zum Abfragen von Neigung und Temperatur ist als G A definiert und wird im Informationsblock übermittelt. Der vollständige Befehl lautet:

```
<SYN><STX>NOCO G A<ETX><CR><LF>
```

Im Programm *HyperTerminal* müssen die Steuerzeichen mit Hilfe der Steuerungstaste eingegeben werden (hier in Caret-Notation):

```
^V^BNOCO G A^C^M^J
```

Der Sensor antwortet danach z. B. mit folgender Zeichenkette:

```
<SYN><STX>CON1 X:+1.835 Y:+0.371 T:+27.6<ETX>ss
```

Die Neigungen in X- und Y-Richtung sind in der Einheit Milliradian, die Temperatur in Grad Celsius. Die letzten beiden Byte sind eine 16 bit lange Prüfsumme der Zeichenkette.

Serielle Kommunikation mit dem Raspberry Pi

Der Raspberry Pi ist mit keiner RS-232-Schnittstelle ausgerüstet. Um trotzdem eine serielle Verbindung mit dem Nivel210 aufzubauen, kann entweder auf einen USB-RS-232-Konverter zurückgegriffen oder eine Buchse am GPIO-Port nachgerüstet werden. Als weitere Möglichkeit bietet sich eine Erweiterungsplatine an, die auf den Raspberry Pi gesteckt wird. Ein Beispiel dafür ist die „RasPiComm“-Platine des österreichischen Unternehmens Amescon, die neben RS-232 auch RS-485 und I²C zur Verfügung stellt.

Für den folgenden Versuch wird wieder auf einen Konverter zurückgegriffen. Für dessen Prolific-Chipsatz ist ein passendes Kernel-Modul verfügbar, das lediglich nachgeladen werden muss:

```
$ modprobe pl2303
```

Bei Debian „Wheezy“ für den Raspberry Pi ist die serielle Schnittstelle standardmäßig als Eingabe/Ausgabe-Konsole vorkonfiguriert. Diese soll aber stattdessen der Kommunikation mit externen Geräten dienen. Die Konsole kann in `/etc/inittab` durch Auskommentieren deaktiviert werden:

```
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Beim Hochfahren sendet der Raspberry Pi alle Systeminformationen an die serielle Schnittstelle. Ein angeschlossenes Gerät kann diese Meldungen empfangen und ggf. verarbeiten. Angeschlossene Sensoren erhalten diese Systeminformationen ebenfalls. Dieses Verhalten lässt sich durch Entfernen der entsprechenden Einträge in `/boot/cmdline.txt` unterbinden.

Einen einfachen Weg zum Testen der seriellen Schnittstelle unter Linux bietet das Programm *minicom* (Abbildung 8.6). Nachdem das Nivel210 mit dem Konverterkabel am Raspberry Pi angeschlossen wurde, lässt sich mit

```
$ minicom -s
```

eine Verbindung aufzubauen. Damit *minicom* die Zeichen ungefiltert an die serielle Schnittstelle weiterleiten kann, darf das Programm nicht innerhalb des Terminal-Multiplexers *tmux* laufen, da dieser die Steuerungstaste abfängt und somit das Schreiben von Steuerzeichen in Caret-Notation verhindert.

Implementierung in Go

Die Programmiersprache Go hat keine Funktion integriert, die eine Kommunikation über die serielle Schnittstelle ermöglicht. Mit *GoSerial* existiert aber eine externe Bibliothek, die diese Lücke rudimentär schließt.

Die Sensoransteuerung in Go wird analog zur Vorgehensweise mit einem Terminal-Programm durchgeführt und erfordert die Angabe von COM-Port und Baud-Rate. Anschließend lassen sich beliebige Daten byteweise lesen und schreiben. Damit die Sensoren die Messbefehle verstehen, müssen die Steuerzeichen des jeweiligen Befehls im Hexadezimal- oder Unicode-Format maskiert sein. Die zurückgesendeten Messwerte werden anschließend mit Hilfe regulärer Ausdrücke extrahiert und in einer MongoDB-Datenbank gespeichert. Mit *mgo* ist ein umfangreicher Datenbanktreiber für Go verfügbar, der MongoDB-Datenbanken ansprechen kann.

Für den Export der Messwerte werden zwei Funktionen umgesetzt. Die erste Funktion ist ein Webdienst, der die Daten in *JavaScript Object Notation* (JSON) ausgibt. JSON ist ein einfaches textbasiertes Datenaustauschformat, das ein ressourcenschonender Ersatz für XML ist. Es basiert auf der JavaScript-Syntax und ist bei Webanwendungen verbreitet [HOLLOSI, 2012]. Für viele Programmiersprachen sind Parser zum Kodieren und Dekodieren von in JSON strukturierten Daten verfügbar. Auch einige dokumentenorientierte NoSQL-Datenbanken speichern ihre Datensätze in JSON.

Für eine Einzelmessung mit dem Leica TM30 hat das JSON-Objekt der Messwerte beispielsweise folgenden Aufbau:

```
{
  "ID": "51f13587a1cef790c5e051b0",
  "Sensor": "Tachymeter",
  "Target": "P2",
  "Message": "%R1P,0,0:0,0.000013323334268,1.592032851688282,
              0.000002354383469,-0.000062458334355,0.000006242937702,
              0.000002356194491,5.8851,3317354\n\n",
  "Time": "2013-07-25T16:26:13.447+02:00"
}
```

Die Antwort des Tachymeters im ASCII-Format wird unter dem Schlüssel *Message* gespeichert. Die einzelnen Messwerte der Antwort sind durch Komma getrennt. Das Tachymeter gibt u. a. die Horizontalrichtung, den Vertikalwinkel, die Genauigkeit der Richtungsmessung, die Instrumentenneigungen und deren Genauigkeit sowie die gemessenen Distanz und deren Messdauer zurück.

Der Zugriff auf eine solche Ressource ist über ein implementiertes Interface auf Basis von *Representational State Transfer* (REST) möglich. Dieses Paradigma dient der eindeutigen Adressierung einer Ressource durch einen *Uniform Resource Locator* (URL). Über das REST-basierte Interface kann sich der Benutzer z. B. die letzten 100 Messwerte im JSON-Format durch den Aufruf einer eindeutigen URL mit der IP-Adresse des Messclients im Webbrower anzeigen lassen:

```
http://192.168.2.1:8080/rest/getlogs/100
```

Dieses Verfahren lässt sich auch anwenden, um alle Messdaten eines bestimmten Sensors oder Zielpunktes auszugeben. In diesem Fall muss eine eindeutige Bezeichnung des Geräts oder Punktes übergeben werden. Der Dienst liest die angeforderten Werte aus der Datenbank aus und gibt sie als JSON-Objekt zurück. Durch eine vorhergehende Authentifizierung, z. B. durch OAuth⁶, kann die Kommunikation auch abgesichert werden. REST bietet sich aufgrund dieser Eigenschaften als Basis für ein *Application Programming Interface* (API) an, um Messwerte in einem standardisierten webbasierten Verfahren abzurufen. Auch Fremdanbieter können darauf aufbauend mit dem Monitoring-System kommunizieren.

Die zweite umgesetzte Funktion ermöglicht den Export von Messwerten als *Comma-Separated Values* (CSV). Die Daten werden zeilenweise in einer Textdatei gespeichert, wobei die Bestandteile einer einzelnen Messung, darunter Zeitstempel, Sensortyp und Messwert, durch Kommas getrennt sind. Mit dem Go-Paket *encoding/csv* ist eine Bibliothek für den CSV-Export verfügbar.

8.3.2. Tachymeter

Neben einem Neigungssensor soll auch ein Tachymeter von dem Messprogramm angesteuert werden. Das verwendete Gerät ist ein Leica TM30. Es zeichnet sich durch eine Richtungsmessgenauigkeit von $\sigma_r = 0,5''$, eine Streckenmessgenauigkeit von $\sigma_s = 0,6 \text{ mm} + 1 \text{ ppm}$ (Reflektor) und einen verschleißarmen Betrieb aus. Das TM30 wurde vom Hersteller für den Einsatz bei dauerhaften Überwachungsmessungen konzipiert.

⁶Protokoll für die Abwicklung von Zugangsberechtigungen in Client-Server-Applikationen

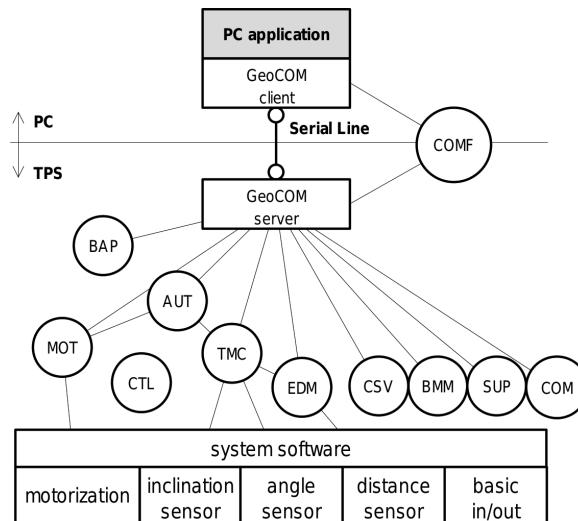


Abbildung 8.7.: Schematische Übersicht über die GeoCOM-Schnittstelle [LEI]

Wie eine Reihe anderer Sensoren von Leica Geosystems lässt sich auch das TM30 über GeoCOM steuern. GeoCOM ist eine Schnittstelle zur Kommunikation zwischen Computern und Sensoren und baut auf dem Client-Server-Modell auf. Grundlage ist ein ASCII-Protokoll, mit dem *Remote Procedure Calls* auf den Sensoren durchgeführt werden. Die Ansteuerung des TM30 unterscheidet sich nicht maßgeblich von der des Nivel210. Die Befehle sind lediglich komplexer und decken einen größere Funktionsumfang ab (Abbildung 8.7). [LEI] Das Auslösen einer Messung und Abfragen der Messwerte ist mit einem einzigen Befehl möglich:

%R1Q,2167:5000,1

Soll das Tachymeter auch ferngesteuert werden, z. B. um Punkte automatisch anzuzielen, ist eine „Erweiterte GeoCOM-Lizenz“ des Herstellers notwendig. Eine Übersicht über die freigeschalteten Lizenzen ist über die Steuerungssoftware abrufbar (<USER> → <STAT> → „System Information“).

8.3.3. Virtueller Sensor

Virtuelle Sensoren sind Software-Systeme, die das Verhalten und die Eigenschaften von echten Sensoren nachbilden und emulieren. Da sie keine realen Werte messen, müssen diese an Hand eines physikalischen Modells nachgebildet werden. Bei der Entwicklung einer Messclient-Software können virtuelle Sensoren sehr nützlich sein. Folgende Szenarien sind denkbar:

Ersatz: Ein virtueller Sensor kann den nachgebildeten ersetzen, weil dieser z. B. nicht oder nur für kurze Zeit für die Implementierung zur Verfügung steht. Auch die Integration von Geräten, die nur im Außenbereich oder nach aufwendiger Montage Daten liefern, wie GNSS-Empfänger oder Extensometer, lässt sich dadurch vereinfachen.

Modultest: Einzelne Komponenten der Messclient-Software können mit einem virtuellen Sensor sehr viel besser getestet und auf korrekte Funktionalität überprüft werden, weil sich Anwendungsfälle nachbilden lassen, deren Abdeckung unter realen Bedingungen

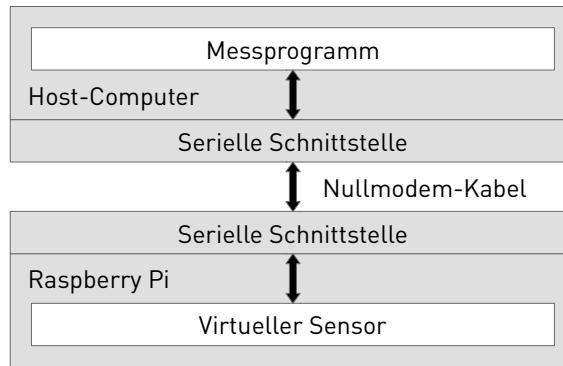


Abbildung 8.8.: Schematische Darstellung des Versuchsaufbaus bei der Entwicklung des virtuellen Sensors

nur schwer oder gar nicht möglich ist. Dazu zählt auch die Erzeugung von Messwerten in Grenzbereichen, z. B. Ausreißer oder Streckenmessungen über große Distanzen.

Ein virtueller Sensor muss zwei Funktionen aufweisen: Zum einen das Empfangen und Verstehen von Messbefehlen. Dies wird erreicht, indem er eine Schnittstelle auf einkommende Daten überwacht, einen etwaigen Messbefehl erkennt und ihn der richtigen Funktion zuordnet. Zum anderen müssen Messwerte erzeugt und zurückgesendet werden. Die Antwort soll wohlgeformt sein, d. h. festgelegten formalen Regeln entsprechen. Denkbar sind sowohl gespeicherte Messreihen des nachzubildenden Sensors als auch nach bestimmten Vorgaben erzeugte künstliche Werte. Die Berechnung der Daten umfasst entweder die vollständige softwareseitige Nachbildung des simulierten Sensors oder nur die Erzeugung pseudozufälliger Werte innerhalb definierter Schranken.

Ein Raspberry Pi dient bei der praktischen Umsetzung als Hardware-Grundlage für den virtuellen Sensor und simuliert dabei ein Leica Nivel210. Grundsätzlich ist jeder Computer mit RS-232-Schnittstelle dafür geeignet. Der Raspberry Pi ist durch ein Nullmodem-Kabel mit dem Host-Rechner verbunden, auf dem die Messclient-Software installiert ist. Der Host sendet die Nivel-spezifischen Befehle an den Raspberry Pi, der diese interpretiert, dem richtigen Sensor zuordnet und erzeugte Messwerte zurücksendet. Die Messclient-Software auf dem Host muss nicht angepasst werden, weil für diese kein Unterschied zwischen einem echten Nivel210 und dem virtuellen Sensor besteht.

Der virtuelle Sensor ist, wie die Messclient-Software, in Go geschrieben. Für die Kommunikation über die serielle Schnittstelle wird wieder auf die externe Bibliothek *GoSerial* zurückgegriffen. Nach dem Start des Sensors überwacht dieser den COM-Port, an dem das Nullmodem-Kabel angeschlossen ist. Sendet der Host den Messbefehl für Neigung und Temperatur, antwortet der virtuelle Sensor mit einem pseudo-zufällig erzeugten Messwert (Abbildung 8.8).

Grafische Benutzeroberfläche

Die erste Version der Client- und der Server-Software von DABAMOS hat bereits eine grafische Benutzeroberfläche integriert. Diese ist vollständig webbasiert und setzt auf XHTML



Abbildung 8.9.: Bildschirmfoto der neuen Weboberfläche von DABAMOS, die Messdaten in Echtzeit anzeigen kann

und AJAX⁷. Die Server-Komponente nutzt darüber hinaus das Framework *JavaServer Faces* zur Darstellung. Mit der JavaScript-Bibliothek *flot* können auch Diagramme aus den auf dem Server gespeicherten Messdaten im Webbrower dargestellt werden. Diese Version basiert noch nicht auf einem REST-Dienst, muss die JSON-Objekte für die Diagramme noch umwandeln und kann keine Messdaten dynamisch nachladen. Darüber hinaus passt sich die Oberfläche nicht den Anforderungen mobiler Endgeräte (Smartphones, Tablet-PCs) an und ist stattdessen auf die Nutzung mit Desktop-Computern ausgelegt.

Eine Neuimplementierung in Go soll die Nachteile der ersten Version ausgleichen und neue Funktionen hinzufügen. Im Rahmen dieser Arbeit kann keine vollständige grafische Benutzeroberfläche umgesetzt werden. Trotzdem soll ein erster Entwurf einen Einblick in die technischen Möglichkeiten geben. Als Grundlage der Oberfläche dient das CSS-Framework *Bootstrap*, das die meisten grafischen Elemente, wie Strukturen, Felder oder Schaltflächen, vordefiniert. Des Weiteren passt sich eine mit *Bootstrap* umgesetzte Webseite von selbst den kleinen Bildschirmen mobiler Endgeräte an. Die Diagramme werden wieder in *flot* gerendert. Die Messwerte können direkt im JSON-Format, d. h. ohne Umwandlung, vom REST-Dienst geladen werden. Eine dynamische Aktualisierung des Diagramms ist möglich. Diese Funktion konnte bereits mit dem virtuellen Sensor aus Abschnitt 8.3.3 getestet werden, arbeitet aber auch mit allen anderen Sensoren zusammen. Die Server-Komponente ist in Go umgesetzt und besteht aus einer einfachen HTTP-Schnittstelle, die HTML5-Seiten zurückgibt (Abbildung 8.9).

⁷Asynchronous JavaScript and XML

Fernwartung mit SSH

Die Konfiguration des Systems und das Auslesen der gesammelten Messwerte ist lokal auf zwei Wegen möglich. Einerseits können Tastatur und Monitor an den Einplatinen-Computer angeschlossen werden. Die Bedienung erfolgt dann direkt durch den Benutzer. Andererseits kann ein zweiter Rechner eine Netzwerkverbindung zum Messclient aufbauen. Über diese ist ein Fernzugriff via Telnet (unverschlüsselt) oder SSH (verschlüsselt) möglich. Der Benutzer kann nach diesem Prinzip nicht nur innerhalb eines lokalen Netzwerkes auf den Messclient zugreifen, sondern auch über das Internet.

Ein Problem stellen dabei Firewalls, Proxy-Server und NAT-Adressumsetzer dar, die den Verbindungsauflauf von außen unterbinden. Um diese zu umgehen, muss der Messclient die Verbindung zum Wartungsrechner aufbauen, und nicht umgekehrt. In der Praxis haben sich dafür zwei Verfahren etabliert: virtuelle private Netzwerke und SSH-Tunnelverbindungen zu einem Gateway. Die Funktionsweise beider Verfahren ist ähnlich: beide Rechnersysteme werden durch ein ideelles Verlängerungskabel – dem Internet – miteinander verbunden.

Für den Aufbau eines virtuellen privaten Netzwerkes ist eine VPN-Software und ein Gateway erforderlich. Als Gateway dient ein Rechner mit fester IP-Adresse, z. B. ein Internet-Server. Der Gateway arbeitet als Einwählknoten, über den Messclient und Wartungsrechner durch eine Tunnelverbindung miteinander kommunizieren.

Eine weitere Möglichkeit des Fernzugriffs ist ein SSH-Tunnel über einen Gateway. Der Vorteil dieser Lösung ist, dass keine VPN-Software installiert werden muss, wenn alle beteiligten Systeme über ein SSH-Programm verfügen. Vielen unixoiden Betriebssystemen liegt bereits die Software *OpenSSH* bei. Der SSH-Dienst des Gateways muss für diese Funktion entsprechend konfiguriert werden:

```
GatewayPorts yes
ClientAliveInterval 10
ClientAliveCountMax 30
```

Die erste Angabe aktiviert die Portweiterleitung des Dienstes. Die beiden folgenden Angaben führen dazu, dass die Netzwerkverbindung aufrecht erhalten wird, auch wenn *Keepalive*-Nachrichten vom Client verzögert eintreffen. Bei zu niedrigen Werten schließt der Gateway bei Verbindungsproblemen den SSH-Tunnel zu früh, wodurch die Verbindung unterbrochen wird.

Der Messclient muss sich anschließend mit dem Gateway verbinden:

```
$ ssh -R 9999:localhost:22 benutzer_server@gateway
```

Der Gateway öffnet daraufhin den Port 9999 für den Fernzugriff und leitet den eingehenden Datenverkehr zu Port 22 des Messclients weiter. Auf diesem Port wartet der SSH-Dienst des Messclients auf eingehende Verbindungen. Der Benutzer kann dann vom Wartungsrechner aus einen Tunnel aufbauen, indem er sich per SSH mit dem Port 9999 des Servers verbindet:

```
$ ssh -p 9999 benutzer_messclient@gateway
```

Der Gateway leitet nun alle vom Wartungsrechner gesendeten Daten an den Messclient weiter und umgekehrt. Über diese Verbindung ist nicht nur die Fernsteuerung des Messclients möglich; mit den SSH-basierten Programmen *SCP* und *SFTP* können auch Daten verschlüsselt ausgetauscht werden.

9. Überwachungsmessungen

Der Messclient ist hard- und softwareseitig im Rahmen von Kurzzeitüberwachungsmessungen auf Funktionsfähigkeit zu prüfen. Mit Hilfe des virtuellen Sensors konnten bereits Sensor-kommunikation und Messdatenspeicherung gestestet werden. Die Experimente sollen sich an der ingenieurgeodätischen Realität orientieren, die Ansteuerung echter Sensoren einschließen und reale Problemstellungen nachbilden. Dazu sind die folgenden Messungen angesetzt:

- Neigungsmessung an eines Messpfeilers im Außenbereich,
- tachymetrische Überwachung von fünf Messpfeilern im Labor und
- tachymetrische Bauwerksüberwachung im Umkreis des Campus der Hochschule Neu- brandenburg.

Gegenstand der Untersuchung ist in erster Linie die Verifizierung der Funktionsfähigkeit des Systems und nicht die Gewinnung von Messdaten. Die Dauer aller Experimente ist daher kurz gehalten. Die Ergebnisse der Messungen werden in Abschnitt 10.3 näher betrachtet.

9.1. Neigungssensor

Nach ersten Versuchen zur Ansteuerung des Neigungssensors Leica Nivel210 soll dieser für die Messung von Pfeilerbewegungen auf dem Dach des Fachbereichs Landschaftswissenschaften und Geomatik der Hochschule eingesetzt werden. Über einen Zeitraum von fünf Tagen erfasst der Sensor Neigungen eines Stahlbetonpfeilers. Die zu erwartenden Bewegungen des Pfeilers werden überwiegend durch den Tagesgang der Sonne verursacht.

Das Nivel210 wird für den Versuchsaufbau mit einer U-formigen Halterung aus Aluminium verschraubt und mit einer Schraubzwinge am Pfeiler befestigt. Ein serielles Datenkabel verbindet Sensor und Messclient miteinander. Um den Messclient vor Witterungseinflüssen zu schützen, befindet sich dieser in einem Schaltschrank der Schutzklasse IP66.

Über eine kabelgebundene Netzwerkverbindung ist der Messclient an einen Router ange- schllossen. Ein Internetzugang besteht wegen fehlender Freischaltung nicht, so dass auf eine Fernwartung verzichtet werden muss. Die Konfiguration und der Start der Messung erfolgt von einem zweiten Rechner aus über SSH.



Abbildung 9.1.: Das Präzisionstachymeter Leica TM30 kommt bei der praktischen Erprobung des Messclients im Labor und auf dem Dach des Fachbereichs Landschaftswissenschaften und Geomatik zum Einsatz

9.2. Tachymeter

Bei zwei weiteren Untersuchungen steuert der Messclient jeweils ein Tachymeter an, um Zielpunkte über einen festgelegten Zeitraum zu überwachen. Der erste Versuch im Labor dient in erster Linie dem Test der Sensor-Integration und der Fernwartung. Der zweite Versuch soll die Eignung des Messclients für Messungen im Außenbereich zeigen.

9.2.1. Überwachung von Messpfeilern im Labor

Die Ansteuerung eines Tachymeters durch das Monitoring-System soll im ersten Experiment unter Laborbedingungen stattfinden. Der Versuchsaufbau umfasst ein Leica TM30 (Abbildung 9.1) und fünf Prismen vom Typ Leica GPR112. Tachymeter und Prismen sind vom Hersteller für den Einsatz bei Überwachungsmessungen ausgelegt. Die Messung wird im Fachlabor „Ingenieurvermessung“ der Hochschule durchgeführt. Sowohl das Tachymeter als auch die Prismen sind auf Stahlbetonpfeilern mit Kopfplatten und 5/8"-Gewinden aufgebaut. Das Leica TM30 muss bei diesem Experiment ohne externe Stromversorgung auskommen. Stattdessen wird lediglich der interne Akkumulator verwendet, um im Anschluss Aussagen über die Laufzeit ohne Netzanschluss treffen zu können.

Der Messclient ist über ein USB-Kabel mit 8-poligem LEMO-Stecker mit dem Tachymeter verbunden. Nachdem die Horizontalrichtungen und Vertikalwinkel zu den fünf Prismen in die Konfigurationsdatei im JSON-Format eingetragen sind, wird die Messung in einer Fernrohrlage gestartet. Beginnend mit Pfeiler 2 misst das Tachymeter die Prismen im Uhrzeigersinn

an. Das Intervall zwischen zwei Messungen beträgt nur 50 s. Die Messung ist beendet, sobald sich das Tachymeter aufgrund des niedrigen Ladezustandes des Akkumulators selbst abschaltet.

Der Messclient ist über einen Router an das Internet angeschlossen. Vor Beginn der Messung wird eine SSH-Verbindung zu einem virtuellen FreeBSD-Server in Schweden aufgebaut, um die in Abschnitt 8.3.3 beschriebene Tunnelverbindung für die Fernwartung nutzen zu können.

Ob und in welchem Umfang mit Bewegungen der Pfeiler zu rechnen ist, lässt sich schwer vorhersagen und ist von der Laufzeit der Messung abhängig. Bewegungen im Millimeter- und Submillimeterbereich, die durch den Tagesgang der Sonne hervorgerufen werden, sind anzunehmen.

9.2.2. Überwachung von Bauwerken auf dem Campus der Hochschule

In einer zweiten Testmessung mit dem Leica TM30 sollen im Außenbereich der Hochschule mehrere Gebäude überwacht werden. Der Versuchsaufbau ähnelt der in Abschnitt 9.2.1 beschriebenen Messung. In das Kurzzeitmonitoring werden die folgenden Punkte einbezogen:

- P1: Hotel „Sankt Georg“ nördlich vom Campus der Hochschule (Südseite)
- P2: Dach von Haus 1 der Hochschule (über dem Treppenhaus an der Nordseite)
- P3: Dach von Haus 1 der Hochschule (über dem Treppenhaus an der Westseite)
- P4: Treppenaufgang an der Mensa (Nordseite),
- P5, P6: Dach von Wohnheim 3 des Studentenwerks Greifswald (beide Nordseite)
- P7: Dach von Haus 3 der Hochschule (Nordwestseite).

Die Punkte P1 und P4 sind als fest installierte Prismen bereits vorgegeben. Die Punkte P2 und P3 wurden neu gesetzt und mit zwei GPR112, die mit GHT112-Halterungen auf verankerte 5/8 "-Gewinde geschraubt sind, bestückt. Auf dem Dach von Studentenwohnheim 3 sind die Punkte P5 und P6 direkt im Stahlbeton der Außenwand des Dachaufbaus verankert. Zwei GPR112 wurden ohne Halterung aufgeschraubt. Der Punkt P7 war bereits in Form eines Aluminium-Zylinders mit Gewinde vorhanden. Als Prisma kommt auch bei ihm ein GPR112 mit Halterung zum Einsatz (Abbildung A.1 b bis f).

Das Tachymeter ist auf dem Dach von Haus 2 der Hochschule auf einem Stahlbetonpfeiler stationiert (Abbildung A.1 a). Die Verbindung zum Messclient erfolgt durch ein Y-Kabel, das zwei getrennte Anschlüsse für den seriellen Datenaustausch und die Stromversorgung besitzt, und ein USB-Konverterkabel. Eine Fernwartung über das Internet ist bei diesem Versuch nicht möglich, weil kurzfristig keine Freischaltung des Messclients im Netz der Hochschule erreicht werden konnte. Die Messung ist zeitlich auf sechs Tage beschränkt.

Durch den Temperatureinfluss der Sonne sind für die überwachten Punkte und den Instrumentenstandpunkt Bewegungen in allen drei Achsen im Millimeterbereich zu erwarten. Ein Trend ist aufgrund der kurzen Dauer des Monitorings aber sehr unwahrscheinlich.

10. Auswertung und Ausblick

Dieses Kapitel widmet sich einer abschließenden Betrachtung der Hard- und Software-Komponenten des ersten Prototyps und gibt einen Ausblick auf zukünftige Entwicklungen. Das System hat noch Raum für Optimierungen, die in weiteren Untersuchungen näher zu betrachten sind. Abschließend werden die Versuchsmessungen mit dem Neigungssensor und dem Tachymeter ausgewertet. Die Messergebnisse sind in Diagrammen dargestellt, die der Arbeit anhängen.

10.1. Hardware

Das Ziel dieser Arbeit, die Entwicklung eines Messclients für geodätische und geotechnische Überwachungsmessungen, konnte in Form eines ersten Prototyps erfolgreich realisiert werden. Das System hat im Rahmen praktischer Versuche seine Funktionsfähigkeit bewiesen. Bei der Planung, der Umsetzung und bei Experimenten mit Sensoren sind aber Probleme und Einschränkungen sichtbar geworden. Zu diesen zählen der fehlende Netzschalter des Einplatinen-Computers sowie die Anordnung und der Platzbedarf der Komponenten im Gehäuse. Die gewonnenen Erkenntnisse machen die Entwicklung eines zweiten Prototyps nötig. In weiteren Experimenten muss zudem die Stabilität und Zuverlässigkeit des Messclients, insbesondere unter verschiedenen klimatischen Bedingungen, weiter erprobt werden.

Die Hardwarekosten lagen mit 175 € weit über den veranschlagten Ausgaben von 100 €. Zwar besteht beim Einkauf größerer Mengen und bei einer gezielten Abstimmung der Komponenten zueinander noch Einsparungspotential, aber ohne Verzicht auf peripherie Teile des Systems können die Kosten bei Rückgriff auf im Handel verfügbarer Lösungen nicht reduziert werden. Die Integration weiterer Komponenten und die Ergänzung durch ein zweites Gehäuse wird darüber hinaus zu Kostensteigerungen führen. Unabhängig davon lässt sich konstatieren, dass es sich bei dem Messclient um eine Low-Cost-Lösung handelt, die kostengünstiger als Industrie-Computer und vollständige Neuentwicklungen ist.

10.1.1. Einplatinen-Computer

Die Untersuchungen haben gezeigt, dass der Raspberry Pi eine geeignete Grundlage für einen autonomen Low-Cost-Messclient darstellt. Die Rechenleistung hat sich als ausreichend erwiesen. Die Auslastung des Systems, insbesondere bei der Interaktion mit mehreren Sensoren gleichzeitig, muss noch getestet werden. Neben der Ansteuerung von Sensoren über die GPIO-Schnittstellen bedürfen Bluetooth-, WLAN- und ZigBee-Adapter weitergehender Betrachtung.

Von Interesse ist auch das Verhalten des Systems bei sehr hohen und sehr niedrigen Temperaturen. In praktischen Versuchen ist daher zu prüfen, inwieweit sich Low-Cost-Temperatursensoren direkt an der GPIO-Schnittstelle betreiben lassen. Sie helfen dabei, Aussagen über die Abwärme der Platinen zu treffen.

Die Unterstützung großer SD-Speicherkarten mit einer Kapazität von 16 GB und mehr ist ein Vorteil bei zeitlich ausgedehnten Überwachungsmessungen, bei denen die anfallenden Messdaten auch lokal zwischengespeichert werden. In diesem Zusammenhang wäre es möglich, die Messdaten auf einer zweiten Partition mit einem plattformübergreifend lesbaren Dateisystem zu speichern. Nach Entnahme der Speicherkarte können die Daten von einem zweiten PC aus direkt gelesen werden.

Als Nachteil hat sich der fehlende Netzschalter des Raspberry Pi erwiesen. Das System kann nur durch Entfernen des Netzkabels oder durch softwareseitiges Herunterfahren ausgeschaltet werden. Ein Hardwareschalter, der ein ACPI-Event auslöst, fehlt. Das Entfernen des Netzkabels kann zu Beschädigungen des Dateisystems und zu Datenverlusten führen. Ein Schalter zum sicheren Herunterfahren sollte daher nachgerüstet werden. Dazu ist eine Schaltung für die GPIO-Schnittstelle zu entwerfen, die ein Interrupt sendet. Ein Programm kann darauf reagieren und den Einplatinen-Computer softwareseitig ausschalten.

10.1.2. Unterbrechungsfreie Stromversorgung

Die USV-Platine LiPo Rider Pro hat fast alle gestellten Anforderungen erfüllt. Im Fall eines Netzausfalls bietet die Anlage aber keine Funktion zur Benachrichtigung der angeschlossenen Verbraucher. Der Nutzer kann somit nicht über den Ausfall informiert werden. Es ist daher in Betracht zu ziehen, die Platine zu einem späteren Zeitpunkt durch eine USV mit dieser Funktion zu ersetzen.

Die experimentelle Bestimmung der Laufzeit des Einplatinen-Computers im Batteriebetrieb hat gezeigt, dass auch längere Unterbrechungen des Netzanschlusses überbrückt werden können. In weiteren Untersuchungen können noch Akkumulatoren mit größerer elektrischer Ladung getestet werden.

10.1.3. Gehäuse

Das Wandgehäuse des ersten Prototyps hat bei den praktischen Versuchen seine Eignung gezeigt. Das Konzept, zwei Gehäuse einzusetzen – ein günstiges Kunststoffgehäuse für die Komponenten und ein stabiles Schaltschränkgehäuse für den Außeneinsatz –, sollte auch beim nächsten Prototyp wieder umgesetzt werden.

Die Anbringung des Einplatinen-Computers und der unterbrechungsfreien Stromversorgung nebeneinander auf einer Grundplatte vergrößert den Platzbedarf unnötig. Eine Montage übereinander ist aufgrund der unterschiedlichen Abmessungen der Platinen und der uneinheitlichen Bohrlöcher nicht möglich. Bei der Entwicklung eines zweiten Prototyps können stattdessen Kartenhalter oder Montageschienen zum Einsatz kommen (Abbildung 10.1). Diese Lösungen reduzieren den Platzbedarf im Inneren des Gehäuses drastisch.

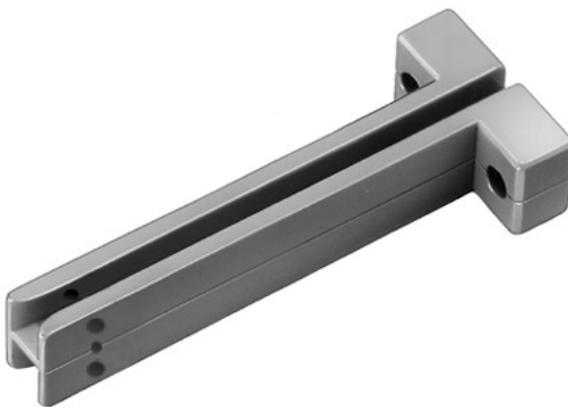


Abbildung 10.1.: Verriegelbare Montageschiene aus Polyamid zum Aufnehmen und Halten von Platinen [FISCHER ELEKTRONIK GMBH & Co. KG, 2013]

10.2. Software

Neben der reinen Hardware-Entwicklung eines autonomen Messclients für Überwachungsaufgaben wurde auch die Steuerungssoftware des Systems überarbeitet, um fortschrittlichere Technologien einsetzen zu können. Diese Arbeit ist, wie auch die Hardware-Entwicklung, noch nicht abgeschlossen und wird weitergeführt. Trotzdem haben sich bereits die Vorteile der Neuimplementierung gezeigt, sowohl in Hinsicht auf die Wahl der Programmiersprache als auch auf die Datenbank.

Die folgenden Ziele wurden bei der Umsetzung erreicht:

- Die Einstellungen des Systems sind nicht mehr als serialisierte Objekte gespeichert sondern in Konfigurationsdateien im JSON-Format ausgelagert.
- Eine REST-basierte Schnittstelle vereinfacht den Datenexport und ermöglicht sowohl die einfache Integration asynchron arbeitender Web-Applikationen als auch die Anbindung von Fremdsoftware über ein *Application Programming Interface*.
- Der Wechsel der Datenbank von einer objektorientierten hin zu einer dokumentenorientierten ermöglicht dynamische Persistenz und garantiert auch in Zukunft die einfache Anbindung von Sensoren jeglicher Art.
- Realisierung eines ersten Entwurfs für eine plattformunabhängige grafische Benutzeroberfläche, die auch mit mobilen Endgeräten nutzbar ist.

10.2.1. Programmiersprache

Im Zuge der Neuimplementierung der Messclient-Software wurde auch die Programmiersprache gewechselt. Während die alte Software noch in Java geschrieben ist und zum Ausführen eine *Java Runtime Environment* benötigt, ist die neue in der Sprache Go umgesetzt und ohne virtuelle Maschine lauffähig.

Im Vergleich zur erste Messclient-Software konnte die Ausführgeschwindigkeit erhöht und der Speicherverbrauch verringert werden. Die Sprachkonzepte von Go, zu denen Interfaces, Reflexionen und einfach integrierbare Nebenläufigkeit gehören, zeigen Vorteile bei der Umsetzung umfangreicher Anwendungen. Die Interoperabilität zwischen den verschiedenen webbasierten Technologien wurde verbessert und die Entwicklungsgeschwindigkeit erhöht. Die Weiterentwicklung der Software in Go wird daher in Zukunft fortgesetzt.

10.2.2. Datenbank

Die eingesetzte dokumentenorientierte NoSQL-Datenbank MongoDB hat sich bei der Speicherung von Messdaten als geeignet erwiesen. Daten verschiedener Sensoren lassen sich in dynamischen Schemata strukturiert im JSON-Format ablegen. Der Import und Export von Daten im JSON-Format erleichtert die Interaktion mit anderen Anwendungen, darunter REST-Dienste und JavaScript-basierte Diagramme zur Echtzeitdarstellung. Die weitere Entwicklungszeit kann dadurch verkürzt und der Datenaustausch einfacher gestaltet werden.

Von Nachteil ist, dass MongoDB auf 32-Bit-Systemen die Größe der Datenbanken auf insgesamt 2 GB limitiert. Diese Grenze wird bei umfangreichen Messungen mit mehreren Sensoren innerhalb weniger Monate erreicht. Einplatinen-Computer mit einer ARM-basierten 64-Bit-Architektur sind noch nicht erhältlich. Aus diesem Grund ist MongoDB für den Produktiveinsatz ungeeignet und sollte ersetzt werden. Ob andere Datenbanksysteme ohne diese Beschränkung auskommen und die gestellten Anforderungen erfüllen, müssen weitere Untersuchungen zeigen. Mögliche Alternativen sind die dokumentenorientierten Datenbanken CouchDB, Redis und RethinkDB. Aus konzeptionellen Gründen sollte die Abstraktion der Datenbankzugriffe angestrebt werden, die einen einfachen Wechsel der Datenbank auch zu einem späteren Entwicklungszeitpunkt möglich macht.

10.3. Überwachungsmessungen

Die Funktionsfähigkeit von Hardware und Software des Messclients sollte durch drei Experimente getestet werden. Neben einer Neigungsmessung wurden zwei tachymetrische Überwachungsmessungen durchgeführt. Der Versuchsaufbau jeder Messung orientierte sich an der ingenieurgeodätschen Realität. Nachfolgend werden die Ergebnisse der Messungen vorgestellt. Die Diagramme sind im Anhang unter A.2 zu finden.

10.3.1. Neigungsmessung

Die Neigungsmessung hatte zum Ziel, einen Messpfeiler auf dem Dach von Haus 2 der Hochschule mit einem Leica Nivel210 auf mögliche Bewegungen zu überwachen. Die Neigungsmessung dauerte fünf Tage und kam zu keinem Ergebnis. Für die Messung stand kein zweiter vollständiger Messclient zur Verfügung, sondern nur ein einzelner Raspberry Pi. Aus zeitlichen Gründen wurde auf die Installation eines Go-Compilers verzichtet und stattdessen eine vorkompilierte Version der Messclient-Software verwendet. Ein Konfigurationsfehler

führte dazu, dass statt einer Dauermessung nur eine Einzelmessung erfolgte. Die Messung muss daher wiederholt werden. Eine Auswertung dieser zweiten Messung ist im Rahmen dieser Arbeit nicht mehr möglich.

10.3.2. Überwachung von Messpfeilern im Labor

Die Laboruntersuchung mit dem Leica TM30 war bereits nach sieben Stunden beendet, nachdem sich das Tachymeter aufgrund der niedrigen Ausgangsspannung des Akkumulators selbst abschaltete. Trotz der kurzen Dauer wurden insgesamt 2241 Einzelmessungen der fünf Zielpunkte durchgeführt.

Ein MATLAB-Programm berechnete die dreidimensionalen Koordinaten der Zielpunkte aus den gemessenen Horizontalrichtungen, Vertikalwinkeln und Schrägstrecken durch polares Anhängen (Listing A.1). Die aus den Koordinatenunterschieden ablesbaren Bewegungen in allen drei Achsen liegen lediglich im Submillimeterbereich. Ein allgemeiner Trend ist nicht erkennbar. Über eine mögliche Temperaturkorrelation kann keine Aussage getroffen werden, weil die Raumtemperatur des Labors nicht gemessen wurde. In Abbildung A.2 sind die reduzierten Koordinaten der fünf Pfeiler grafisch dargestellt.

10.3.3. Überwachung von Bauwerken auf dem Campus der Hochschule

Die Überwachung von Gebäuden am und auf dem Campus der Hochschule Neubrandenburg dauerte insgesamt sechs Tage. In dieser Zeit wurden 848 Einzelmessungen zu den sieben Zielpunkten durchgeführt. Das Messintervall zwischen den Messungen betrug 10 min. Für jeden Punkt konnten etwa 120 Messwerte gesammelt werden. Die Auswertung des Monitorings erfolgte, wie schon bei der Laboruntersuchung, in MATLAB. Die Koordinaten der Zielpunkte wurden durch polares Anhängen berechnet und anschließend reduziert, damit die Bewegungen in allen drei Achsen vergleichbar sind (Abbildung A.3).

Die Punkte weisen Bewegungen von bis zu 4,5 mm in der Lage und 7 mm in der Höhe auf. Die Punkte 5 und 6, die beide an der selben Gebäudewand befestigt sind, zeigen über die gesamte Messungsdauer sehr ähnliche Bewegungen. Die zeitliche Differenz dieser Bewegungen ergibt sich aus der Tatsache, dass Punkt 6 immer 10 min nach Punkt 5 angemessen wurde. Auch Punkt 2 und 3, die beide auf dem Dach von Haus 1 der Hochschule befestigt sind, zeigen vergleichbare Bewegungen. Die Messergebnisse sind daher plausibel, auch wenn sich die genaue Ursache der zum Teil großen Bewegungen nicht genau bestimmen lässt.

Am wahrscheinlichsten sind zwei Einflüsse: zum einen der Temperatureinfluss auf die Streckenmessung, der 1 ppm pro °C beträgt, zum anderen die Eigenbewegung des Instrumentenstandpunktes infolge des Tagesganges der Sonne. Da keine Messung von Temperatur, Luftdruck und Luftfeuchte vorgenommen wurde, können die Strecken zu den Zielpunkten nicht korrigiert werden. Die Eigenbewegungen des Instrumentenstandpunktes könnte hingegen durch eine Netzausgleichung bestimmt und aus den Zielkoordinaten rechnerisch entfernt werden. Die vom Tachymeter aufgezeichneten Neigungen der Stehhachse sind sehr gering und treten nur als Ausreißer in einer Größenordnung von maximal 2 mgon auf.

Um die Aussagen hinsichtlich des Temperatureinflusses zu verifizieren, sollte noch eine weitere Messung durchgeführt werden, bei der der Instrumentenstandpunkt isoliert und ggf. überdacht ist. Die Erfassung von Temperatur und Luftdruck ist ebenfalls anzuraten, um eine Streckenkorrektur durchführen zu können.

A. Anhang

A.1. Abbildungen



(a) Leica TM30 auf Haus 2 der Hochschule



(b) Punkt 1 an der Südseite des Hotels „Sankt Georg“



(c) Punkt 2 auf Haus 1 der Hochschule



(d) Punkt 4 an der Mensa der Hochschule



(e) Punkt 5 auf Wohnheim 3 des Studentenwerks



(f) Punkt 7 auf Haus 3 der Hochschule

Abbildung A.1.: Angesteuertes Tachymeter Leica TM30 und Prismen auf dem bzw. am Campus der Hochschule Neubrandenburg

A.2. Diagramme

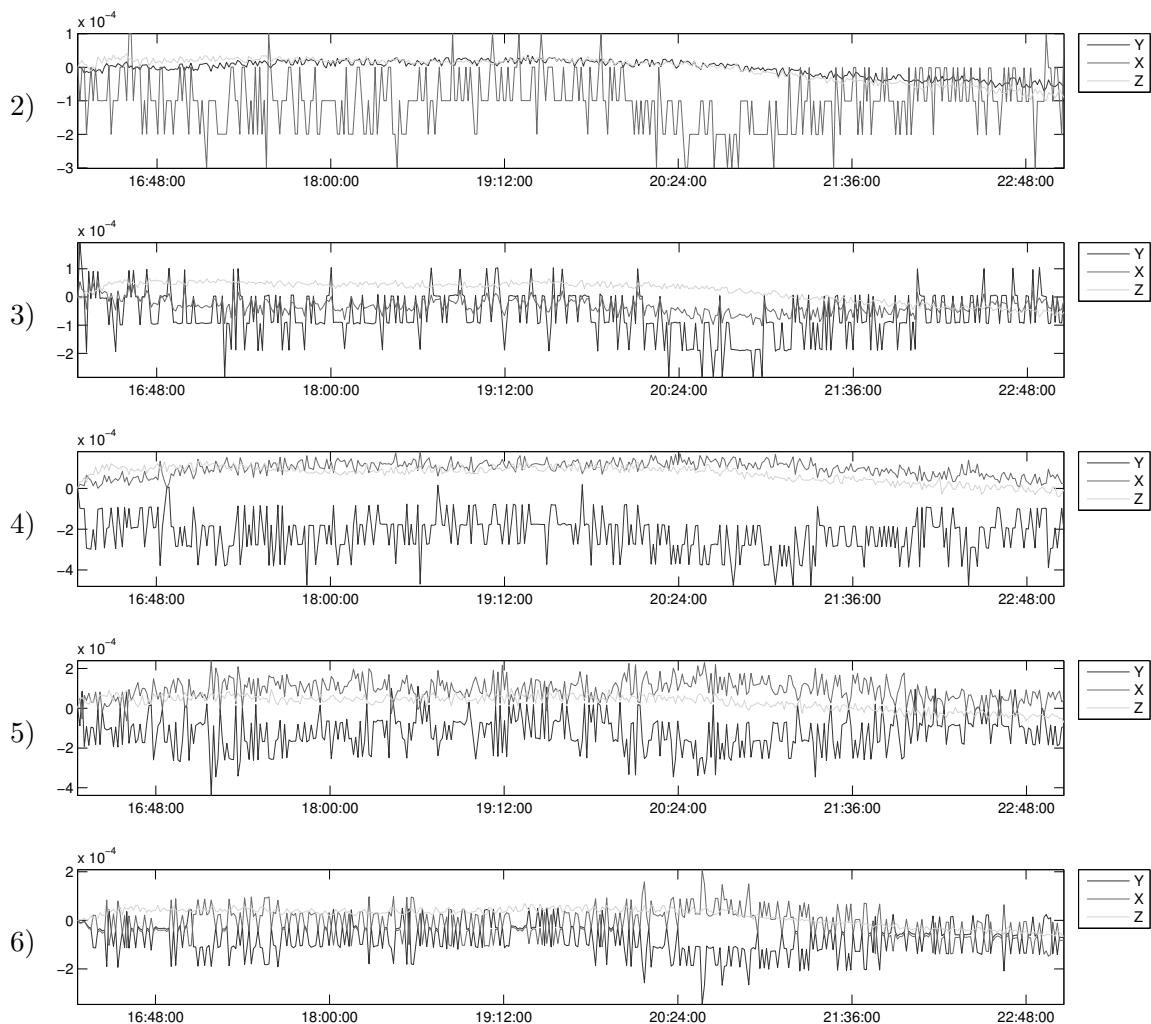


Abbildung A.2.: Die Bewegungen in Lage und Höhe der Pfeiler 2 bis 6 in Meter, gemessen im Labor 318 über einen Zeitraum von sieben Stunden (Intervall: 50 s)

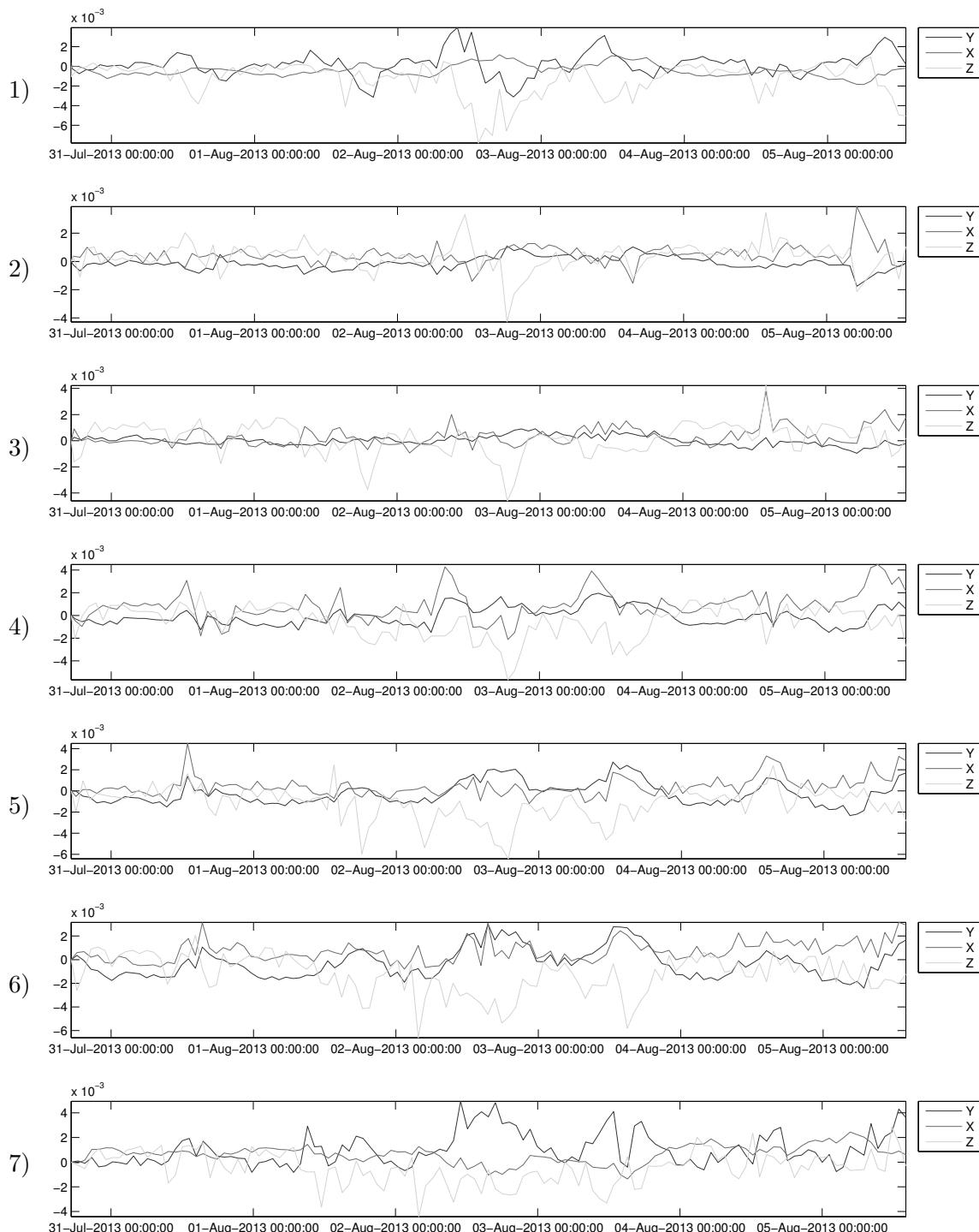


Abbildung A.3.: Die Bewegungen in Lage und Höhe der Punkte 1 bis 7 in Meter, gemessen vom Dach von Haus 2 der Hochschule über einen Zeitraum von sechs Tagen (Intervall: 10 min)

A.3. MATLAB-Quelltext

Listing A.1: MATLAB-Programm zum Berechnen, Reduzieren und Plotten von Koordinaten

```

1 fid = fopen('Leica_TM30.csv', 'r');
2 A = textscan(fid, '%s%f%f%*f%*f%*f%*f%*f%*f', ,
    delimiter', ',');
3 fclose(fid);
4
5 % Nummer des Punktes in der CSV-Datei
6 punkt = 'P2';
7
8 P = [A{1}];      % Punkt-Nr.
9 Hz = [A{2}];     % Horizontalwinkel
10 V = [A{3}];     % Vertikalwinkel
11 I = [A{4}];     % Neigung
12 S = [A{5}];     % Strecke
13 T = [A{6}];     % Zeitstempel
14
15 % Koordinaten des Instrumentenstandpunktes
16 p1_x = 1000;
17 p1_y = 1000;
18 p1_z = 1000;
19
20 % Arrays initialisieren
21 Y(1) = 0;
22 X(1) = 0;
23 Z(1) = 0;
24 DT(1) = 0;
25 Inc(1) = 0;
26
27 n = length(P);
28 j = 1;
29
30 % polares Anhaengen
31 for i = 1:n
32     if strcmp(P(i), punkt)
33         Y(j) = p1_y + S(i) * sin(Hz(i)) * sin(V(i));
34         X(j) = p1_x + S(i) * cos(Hz(i)) * sin(V(i));
35         Z(j) = p1_z + S(i) * cos(V(i));
36         DT(j) = datenum(T(i), 'yyyy-mm-ddTHH:MM:SS');
37         Inc(j) = I(i) * 200/pi;
38         j = j + 1;
39     end
40 end
41

```

```
42 % Reduzieren
43 m = length(Y);
44
45 for i = 2:m
46     Y(i) = Y(1) - Y(i);
47     X(i) = X(1) - X(i);
48     Z(i) = Z(1) - Z(i);
49 end
50
51 Y(1) = 0;
52 X(1) = 0;
53 Z(1) = 0;
54
55 % Plotten
56 hFig = figure(1);
57 set(hFig, 'Position', [400 400 1000 175], 'Name', punkt);
58 axes('position', [.025 .15 .97 .7]);
59
60 plot(DT, Y, 'b')
61 hold all
62 plot(DT, X, 'r')
63 plot(DT, Z, 'g')
64 hold off
65
66 % X-Achse
67 dateFormat = 13;
68 datetick('x', dateFormat, 'keepticks')
69 axis tight
70
71 % Legende
72 legend('Y', 'X', 'Z', 'Location', 'NorthEastOutside');
```

B. Eidesstattliche Erklärung

Ich versichere, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit lag in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vor.

Neubrandenburg, den 19. August 2013

.....
(Unterschrift des Kandidaten)

Abbildungsverzeichnis

3.1. Die GNSS-Antenne Leica AS10 unterstützt GPS, GLONASS, Galileo, Compass und SBAS und wird für einzelne Referenzstationen oder RTK- und Monitoringnetze verwendet	8
3.2. Der Low-Cost-Einfrequenz-Empfänger „Piksi“ soll durch Echtzeitkinematik eine Lagegenauigkeit von bis zu 1 cm erreichen	9
3.3. Linearer Dehnungsmesstreifen mit einem Messgitter aus Konstantan	10
3.4. Bluetooth-fähiger Low-Cost-Zweiachsneigungssensor DXL360S mit einer Genauigkeit von $\pm(0,08^\circ + 1\%)$, bei einer Auflösung von $0,01^\circ$	12
4.1. Bildschirmfoto der Weboberfläche von DABAMOS	14
4.2. Testmessungen mit DABAMOS an einem Brückenmodell	15
4.3. Schematischer Aufbau eines Monitorings mit DABAMOS bei Nutzung von zwei Tachymetern	16
4.4. Bildschirmfoto des Monitoring-Systems Leica GeoMoS	18
5.1. Der PIC18F2620-I/SP ist ein einfacher 8-Bit-Mikrocontroller mit 40 MHz Systemtakt und 64 KB Programmspeicher.	24
6.1. Ein <i>GSM Shield</i> mit eingesetzter SIM-Karte, das auf einem Arduino Uno steckt	28
6.2. Die kreditkartengroße Platine des BeagleBone Black	29
6.3. Einplatinen-Computer Carambola (gelbe Platine) auf dem Entwickler-Board, das die Anschlüsse bereitstellt	30
6.4. Einplatinen-Computer Raspberry Pi (Modell B) mit Ethernet-Schnittstelle und zwei USB-Ports	31
6.5. Das Neigungsmessgerät BlueCLIO von Wyler ist mit einer Bluetooth-Schnittstelle ausgerüstet	34
6.6. Das Triometer ALL3418 von ALLNET dient der netzwerkbasierten Erfassung von Temperatur und Luftfeuchte. Die Messwerte des Sensors werden als HTML-Seite über HTTP ausgegeben. Die Spannungsversorgung ist optional auch über Power over Ethernet möglich.	36
6.7. Beispiel für eine GPIO-Schnittstelle mit 26 Pins	37
6.8. Ein <i>Serial Device Server</i> von Digi International	39
6.9. Beispiel für einen USB-Konverter mit zwei Ports für RS-232, RS-422 und RS-485	40
6.10. Das ZigBee-Modul „XBee-PRO 868“ mit Drahtantenne von Digi International	42
6.11. Vergleich der Lade-Entlade-Spannung eines Lithium-Ionen-Akkumulators und eines Doppelschichtkondensators im Konstantstrombetrieb	44
6.12. USV-Platine „LiPo Rider Pro“ mit Lithium-Polymer-Akkumulator und Solarmodul	46

Abbildungsverzeichnis	92
-----------------------	----

7.1. Bildschirmfoto der grafischen Benutzeroberfläche von RISC OS Pi	55
8.1. Einfacher WLAN-Adapter mit fester Antenne für den USB-Port	59
8.2. Erster Prototyp des Messclients im Kunststoffgehäuse	61
8.3. Das Werkzeug <i>raspi-config</i> liegt in einigen Linux-Distributionen bei und vereinfacht die Konfiguration des Betriebssystems	63
8.4. Neigungssensor Leica Nivel210	66
8.5. Das Programm <i>HyperTerminal</i> kann für die einfache Kommunikation mit dem Leica Nivel210 verwendet werden. Die nicht druckbaren Steuerzeichen werden durch Symbole dargestellt.	67
8.6. Das Werkzeug <i>minicom</i> kann unter unixartigen Betriebssystemen für die einfache Kommunikation über die serielle Schnittstelle eingesetzt werden	68
8.7. Schematische Übersicht über die GeoCOM-Schnittstelle	71
8.8. Schematische Darstellung des Versuchsaufbaus bei der Entwicklung des virtuellen Sensors	72
8.9. Bildschirmfoto der neuen Weboberfläche von DABAMOS, die Messdaten in Echtzeit anzeigen kann	73
9.1. Das Präzisionstachymeter Leica TM30 kommt bei der praktischen Erprobung des Messclients im Labor und auf dem Dach des Fachbereichs Landschaftswissenschaften und Geomatik zum Einsatz	77
10.1. Verriegelbare Montageschiene aus Polyamid zum Aufnehmen und Halten von Platinen	81
A.1. Angesteuertes Tachymeter Leica TM30 und Prismen auf dem bzw. am Campus der Hochschule Neubrandenburg	85
A.2. Die Bewegungen in Lage und Höhe der Pfeiler 2 bis 6 in Meter, gemessen im Labor 318 über einen Zeitraum von sieben Stunden (Intervall: 50 s)	86
A.3. Die Bewegungen in Lage und Höhe der Punkte 1 bis 7 in Meter, gemessen vom Dach von Haus 2 der Hochschule über einen Zeitraum von sechs Tagen (Intervall: 10 min)	87

Tabellenverzeichnis

2.1. Kategorien der Messaufgaben	4
3.1. Beim Monitoring von Bauwerken eingesetzte geodätische Sensoren mit ihren Eigenschaften	7
6.1. Gegenüberstellung ausgesuchter Einplatinen-Computer mit ihren Einzelhandelspreisen	26
6.2. Spezifikationen der beiden Modellvarianten des Raspberry Pi	32
6.3. Leistungsklassen von Bluetooth	33
6.4. Vergleich zwischen WLAN, Bluetooth und ZigBee	41
6.5. Leistungsklassen nach IEEE 802.3af-2003 für Power over Ethernet	48
6.6. IP-Schutzarten nach DIN EN 60529	50

Literaturverzeichnis

- [8devices 2013] 8DEVICES: *Webseite*. <http://8devices.com/>. Version: 28. Mai 2013
- [Allelein 2013] *Kapitel Energiespeicherung*. In: ALLELEIN, H. J.: *Energietechnik. Systeme zur Energieumwandlung. Kompaktwissen für Studium und Beruf*. Wiesbaden : Springer Vieweg, 2013, S. 419–445
- [Arduino 2013] ARDUINO: *Webseite*. <http://arduino.cc/>. Version: 1. Juni 2013
- [Arduino Blog 2013] ARDUINO BLOG: *Welcome Arduino Yún – the first member of a series of wifi products combining Arduino with Linux*. <http://blog.arduino.cc/2013/05/18/welcome-arduino-yun-the-first-member-of-a-series-of-wifi-products-combining-arduino-with-linux/>. Version: 22. Mai 2013
- [Banzi 2011] BANZI, M.: *Getting Started with Arduino*. Sebastopol, CA : O'Reilly Media, 2011
- [Böllin 2013] *Kapitel Solartechnik*. In: BÖLLIN, E.: *Energietechnik. Systeme zur Energieumwandlung. Kompaktwissen für Studium und Beruf*. Wiesbaden : Springer Vieweg, 2013, S. 286–326
- [Bähring 2010] BÄHRING, H.: *Anwendungsorientierte Mikroprozessoren. Mikrocontroller und Digitale Signalprozessoren*. Berlin u. a. : Springer, 2010
- [Coley 2012a] COLEY, G.: *BeagleBoard System Reference Manual*. Rev D. Richardson, TX: BeagleBoard.org, 15. Oktober 2012. https://github.com/CircuitCo/BeagleBoard-RevD/blob/master/Beagle_SRM_D.pdf
- [Coley 2012b] COLEY, G.: *BeagleBoard-xM System Reference Manual*. Rev C2. Richardson, TX: BeagleBoard.org, 11. Oktober 2012. https://raw.github.com/CircuitCo/BeagleBoard-xM-RevC2/master/Beagle_SRM_XM_C2_0_0.pdf
- [Coley 2013] COLEY, G.: *BeagleBone Black System Reference Manual*. Rev A5.6. Richardson, TX: BeagleBoard.org, 15. Juni 2013. https://raw.github.com/CircuitCo/BeagleBone-Black/master/BBB_SRM.pdf
- [DB-Engines 2013] DB-ENGINES: *DB-Engines Ranking*. <http://db-engines.com/de/ranking>. Version: 27. Juni 2013
- [Deumlich u. Staiger 2002] DEUMLICH, F. ; STAIGER, R.: *Instrumentenkunde der Vermessungstechnik*. Heidelberg : Wichmann, 2002
- [Digi International Inc. 2013] DIGI INTERNATIONAL INC.: *Webseite*. <http://www.digi.com/>. Version: 20. Juni 2013

- [Eichhorn 2012] EICHHORN, A.: Monitoring: ein grundlegender Beitrag zur Überwachung von Bauwerken. In: *AVN – Allgemeine Vermessungs-Nachrichten* 4 (2012), S. 123 – 131
- [Emadi u. a. 2004] EMADI, A. ; NASIRI, A. ; BEKiarov, B. S. *Uninterruptible Power Supplies and Active Filters*. Boca Raton, FL : CRC Press, 2004
- [embedded projects GmbH 2013] EMBEDDED PROJECTS GMBH: Webseite. <http://gnublin.embedded-projects.net/>. Version: 20. Juni 2013
- [EXSYS Vertriebs GmbH 2013] EXSYS VERTRIEBS GMBH: Webseite. <http://www.exsys.ch/>. Version: 10. Juli 2013
- [Faludi 2011] FALUDI, R.: *Building Wireless Sensor Networks. With ZigBee, XBee, Arduino, and Processing*. Sebastopol, CA : O'Reilly Media, 2011
- [Farahani 2008] FARAHANI, S.: *ZigBee Wireless Networks and Transceivers*. Burlington, MA : Newnes, 2008
- [Fischer Elektronik GmbH & Co. KG 2013] FISCHER ELEKTRONIK GMBH & Co. KG: Webseite. <http://www.fischerelektronik.de/>. Version: 2. August 2013
- [Gessler u. Krause 2009] GESSLER, R. ; KRAUSE, T.: *Wireless-Netzwerke für den Nahbereich. Eingebettete Funksysteme: Vergleich von standardisierten und proprietären Verfahren. Grundlagen, Verfahren, Vergleich, Entwicklung*. Vieweg+Teubner, 2009
- [GOCA 2013] GOCA: Webseite. <http://www.goca.info/>. Version: 12. August 2013
- [Hamann 2010] HAMANN, M.: *Entwicklung eines virtuellen Sensors für Vermessungsaufgaben*. Neubrandenburg, Hochschule Neubrandenburg – University of Applied Sciences, Bachelorarbeit, 2010
- [Hollosi 2012] HOLLOSI, A.: *Von Geodaten bis NoSQL: Leistungsstarke PHP-Anwendungen. Aktuelle Techniken und Methoden für Fortgeschrittene*. München : Hanser, 2012
- [Horak 2007] HORAK, R.: *Telecommunications and Data Communications Handbook*. Hoboken, NJ : Wiley, 2007
- [Hottinger u. Keil 1994] KAPITEL Allgemeines zur Gerätetechnik. In: HOTTINGER, K. ; KEIL, S.: *Handbuch der industriellen Meßtechnik*. München : Oldenbourg, 1994, S. 121 – 359
- [Kartalopoulos 2011] KARTALOPOULOS, S. V.: *Free Space Optical Networks for Ultra-Broad Band Services*. Hoboken, NJ : Wiley, 2011
- [Kollar 2000] KOLLAR, L. ; GEVATTER, H.-J. (Hrsg.): *Automatisierungstechnik 1. Meß- und Sensortechnik*. Berlin u. a. : Springer, 2000
- [Lei] LEICA GEOSYSTEMS AG (Hrsg.): *Leica TPS1200+, Leica TS30 / TM30. GeoCom Reference Manual*. 1.50. Heerbrugg, CH: Leica Geosystems AG
- [Lei 2006] LEICA GEOSYSTEMS AG (Hrsg.): *Nivel200. Technical Reference Manual*. 1.0. Heerbrugg, CH: Leica Geosystems AG, 2006
- [Marte u. a. 2012] MARTE, R. ; SCHÄRINGER, F. ; PAULUS-GRILL, M. ; BOLEY, C. (Hrsg.): *Handbuch Geotechnik. Grundlagen – Anwendungen – Praxiserfahrungen*. Wiesbaden : Vieweg+Teubner, 2012

- [Metter u. Bucher 2007] METTER, M. ; BUCHER, R. ; SIEMENS AG (Hrsg.): *Industrial Ethernet in der Automatisierungstechnik. Planung und Einsatz von Ethernet-LAN-Techniken im Umfeld von SIMATIC-Produkten*. Erlangen : Publicis, 2007
- [Odendahl u. a. 2009] ODENDAHL, M. ; FINN, J. ; WENGER, A.: *Arduino. Physical Computing für Bastler, Designer und Geeks*. Köln : O'Reilly, 2009
- [Paul 1998] Kapitel Sensoren und Verfahren. In: PAUL, H.: *Sensortechnik. Handbuch für Praxis und Wissenschaft*. Berlin u. a. : Springer, 1998, S. 329–394
- [Pelzer u. a. 1988] PELZER, H. u. a.: Deformationsmessungen. In: PELZER, H. (Hrsg.): *Ingenieurvermessung. Deformationsmessungen. Massenberechnung. Ergebnisse des Arbeitskreises 6 des Deutschen Vereins für Vermessungswesen (DVW) e. V.* Stuttgart : Wittwer, 1988, S. 37–192
- [Raspberry Pi Foundation 2013] RASPBERRY PI FOUNDATION: Webseite. <http://www.raspberrypi.org/>. Version: 10. Mai 2013
- [Reith u. Wasmeier 2009] REITH, C. ; WASMEIER, P.: „Das geht schon vorüber?!” – Zeitreihenanalyse beim Bauwerksmonitoring. In: *Zeitabhängige Messgrößen – Verborgene Schätze in unseren Daten. Beiträge zum 85. DVW-Seminar am 7. und 8. September 2009 in Kassel*, Bd. 59. Augsburg : Wißner, 2009 (DVW-Schriftenreihe), S. 79–94
- [Resnik 2010] RESNIK, B.: Realisierung und Analyse von Schwingungsmessungen in Rahmen des Monitorings am Beispiel eines Brückenbauwerkes in Armenien. In: *AVN – Allgemeine Vermessungs-Nachrichten* 6 (2010), S. 227–232
- [Resnik 2013] RESNIK, B.: *Structural Health Monitoring (SHM) am Beispiel von Schwingungsmessungen an Brückenbauwerken*. Beitrag zur „tech13 – Aktuelle Trends und Herausforderungen in der Ingenieurgeodäsie“ an der TU Berlin, 18.–19. April 2013
- [Riggert 2012] RIGGERT, W.: *Rechnernetze. Grundlagen – Ethernet – Internet*. München : Fachbuchverlag Leipzig, 2012
- [RONJA 2013] RONJA: Webseite. <http://ronja.twibright.com/>. Version: 20. Juni 2013
- [Sauter 2011] SAUTER, M.: *Grundkurs Mobile Kommunikationssysteme. UMTS, HSDPA und LTE, GSM, GPRS und Wireless LAN*. Wiesbaden : Vieweg+Teubner, 2011
- [Schlemmer 1996] SCHLEMMER, H.: *Grundlagen der Sensorik. Eine Instrumentenkunde für Vermessungsingenieure*. Heidelberg : Wichmann, 1996
- [Schmidt 2012] SCHMIDT, M.: *Raspberry Pi. A Quick-Start Guide*. Dallas, TX : The Pragmatic Bookshelf, 2012
- [Schröder u. a. 2009] SCHRÖDER, J. ; GOCKEL, T. ; DILLMANN, R.: *Embedded Linux. Das Praxisbuch*. Berlin, Heidelberg : Springer, 2009
- [Schwarz 2013] SCHWARZ, W.: *Faseroptische Sensoren. Messsysteme mit Zukunft*. Beitrag zur „tech13 – Aktuelle Trends und Herausforderungen in der Ingenieurgeodäsie“ an der TU Berlin, 18.–19. April 2013

- [Schweimler 2010] SCHWEIMLER, B.: *Einsatzfähigkeit aktueller Datenbanksysteme für Überwachungsmessungen*. Neubrandenburg, Hochschule Neubrandenburg – University of Applied Sciences, Bachelorarbeit, 2010
- [Specovius 2010] SPECOVIUS, J.: *Grundkurs Leistungselektronik. Bauelemente, Schaltungen und Systeme*. Wiesbaden : Vieweg+Teubner, 2010
- [Spurgeon 2000] SPURGEON, C. E.: *Ethernet. The Definitive Guide*. Sebastopol, CA : O'Reilly Media, 2000
- [Staiger 2009] STAIGER, R.: Instrumente und Messsysteme zur Beobachtung zeitabhängiger Größen. In: *Zeitabhängige Messgrößen – Verborgene Schätzungen in unseren Daten. Beiträge zum 85. DVW-Seminar am 7. und 8. September 2009 in Kassel*, Bd. 59. Augsburg : Wißner, 2009 (DVW-Schriftenreihe), S. 11–21
- [Stempfhuber 2009] STEMPFHUBER, W.: Geodätische Monitoringsysteme – Stand der Technik und Abgrenzung der gegenwärtigen Systeme. In: *Zeitabhängige Messgrößen – Verborgene Schätzungen in unseren Daten. Beiträge zum 85. DVW-Seminar am 7. und 8. September 2009 in Kassel*, Bd. 59. Augsburg : Wißner, 2009 (DVW-Schriftenreihe), S. 181–191
- [Stempfhuber u. Alberding 2012] STEMPFHUBER, W. ; ALBERDING, J.: Geodätische Monitoringsysteme mit RTK Low-Cost-GNSS. In: *AVN – Allgemeine Vermessungs-Nachrichten* 4 (2012), S. 132–139
- [Swift Navigation Inc. 2013] SWIFT NAVIGATION INC.: *Piksi Datasheet. Flexible, high-performance GPS receiver platform running open-source software*. http://docs.swift-nav.com/pdfs/piksi_datasheet_v2.3.1.pdf. Version: 10. August 2013
- [TIOBE Software BV 2013] TIOBE SOFTWARE BV: *TIOBE Programming Community Index for July 2013*. http://www.tiobe.com/index.php/content/paperinfo/tpci_index.html. Version: 24. Juli 2013
- [Welsch u. a. 2000] WELSCH, W. ; HEUNECKE, O. ; KUHLMANN, H. ; MÖSER, M. (Hrsg.) u. a.: *Handbuch Ingenieurgeodäsie. Auswertung geodätischer Überwachungsmessungen*. Heidelberg : Wichmann, 2000
- [Wolff 2010] WOLFF, Ch.: *Entwicklung eines Webinterfaces zur Fernwartung von Überwachungsmessungen*. Neubrandenburg, Hochschule Neubrandenburg – University of Applied Sciences, Bachelorarbeit, 2010
- [Zeimetz u. a. 2009] ZEIMETZ, P. ; ELING, C. ; KUHLMANN, H.: Analyse von GPS-Referenzstationsbeobachtungen mit Methoden der Zeitreihenanalyse. In: *Zeitabhängige Messgrößen – Verborgene Schätzungen in unseren Daten. Beiträge zum 85. DVW-Seminar am 7. und 8. September 2009 in Kassel*, Bd. 59. Augsburg : Wißner, 2009 (DVW-Schriftenreihe), S. 193–210
- [Zhang 2010] ZHANG, P.: *Advanced Industrial Control Technology*. Burlington, MA : Elsevier, 2010
- [Zisler 2006] ZISLER, H.: *FreeBSD*. Poing : Franzis, 2006